

Werner Neundorf

**Numerik gewöhnlicher Differentialgleichungen
mit Computeralgebrasystemen**

Anfangswertprobleme

Wichtiger Hinweis

Die in diesem Buch wiedergegebenen Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind für Lehrzwecke bestimmt.

Alle Angaben und Programme in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert.

Trotzdem sind Fehler nicht ganz auszuschließen. Der Autor und der Verlag weisen deshalb darauf hin, dass weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann.

Für die Mitteilung eventueller Berichtigungen ist der Autor jederzeit dankbar.

Werner Neundorf

Numerik
gewöhnlicher Differentialgleichungen
mit Computeralgebrasystemen
Anfangswertprobleme



Universitätsverlag Ilmenau
2013

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Technische Universität Ilmenau/Universitätsbibliothek

Universitätsverlag Ilmenau

Postfach 10 05 65

98684 Ilmenau

www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung

Verlagshaus Monsenstein und Vannerdat OHG

Am Hawerkamp 31

48155 Münster

www.mv-verlag.de

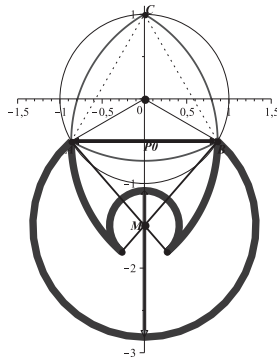
ISBN 978-3-86360-064-8 (Druckausgabe)

URN urn:nbn:de:gbv:ilm1-2013100028

**Für Babcia Bronia
in Dankbarkeit**

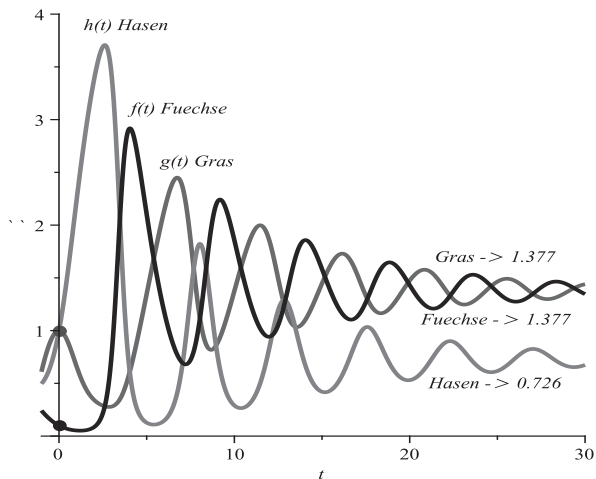
Eric Temple Bell, 1883-1960

*“Offensichtlich“
ist das gefährlichste Wort in der Mathematik.*



Archimedes

*Es gibt Dinge, die den meisten Menschen
unglaublich erscheinen, die sich nicht mit
Mathematik beschäftigen haben.*



Vorwort

Im vorliegenden Buch werden zahlreiche wichtige Aspekte der numerischen Behandlung von Anfangswertproblemen gewöhnlicher Differentialgleichungen dargestellt.

Sein Inhalt ist aus Vorlesungen und Seminaren hervorgegangen, die ich in vielen Jahren an der Technischen Universität Ilmenau für Studenten der Mathematik, der Natur- und Ingenieurwissenschaften gehalten habe. Dabei wird der Bogen gespannt von praktischen Modellen über die numerischen Verfahren hin bis zur Implementierung und Wertung von Software. Programmtechnische Umsetzungen erfolgen vorwiegend in den Computeralgebrasystemen Maple und MATLAB, wobei der teilweise eingebundene Programmcode auch dem Leser, der keine speziellen Kenntnisse zu CAS besitzt, verständlich sein wird. Damit finden die Anwender nützliche Hinweise zur Lösung ihrer Probleme, die mit Anfangswertproblemen verbunden sind.

Es gibt auf dem Gebiet der Numerik der Differentialgleichungen ein sehr großes Angebot an Literatur. In Lehrbüchern über Numerische Mathematik, in speziellen Büchern und Monographien dazu findet der Leser sowohl die traditionellen Methoden als auch die an den aktuellen Stand der Forschung heranreichenden Ergebnisse.

Anliegen dieses Buches ist es, Studenten und mathematisch interessierten Naturwissenschaftlern und Ingenieuren mit Vorkenntnissen der Analysis und linearen Algebra ein einbändiges Lehrbuch und Nachschlagewerk zur numerischen Behandlung von AWP in Verbindung mit den genannten CAS bereitzustellen.

Das Besondere ist die große Anzahl von Differentialgleichungen, die mittels Maple bezüglich ihrer Eigenschaften untersucht werden. Dabei können sich in den Beispielen durchaus gewisse Informationen wiederholen. Aber beim genauen Studium wird der Leser immer wieder Neues und Interessantes entdecken. Es wird auch auf viele Details beim Umgang mit Maple in Bezug auf Differentialgleichungen hingewiesen.

Das Buch stellt im Kapitel 1 zahlreiche Differentialgleichungen und Systeme, viele mit praktischem Bezug, mit ihren Eigenschaften vor. Hier werden auch schon einfache numerische Verfahren einbezogen. Das umfangreiche Kapitel 2 zeigt, wie Differentialgleichungen mit dem CAS Maple gelöst werden können (mit Bezug auf die nachfolgenden Kapitel) und was der Nutzer dabei alles berücksichtigen und untersuchen sollte.

Da Differentialgleichungen mit Ableitungen zu tun haben, wurde das Kapitel 3 zu Differenzenapproximationen und Taylor-Reihen eingefügt.

In den weiteren Kapiteln zur Numerik stehen dann die

- expliziten Einschrittverfahren,
- linearen expliziten und impliziten Mehrschrittverfahren,
- impliziten Einschrittverfahren,
- Fragen der Konsistenz, Konvergenz und Stabilität der Verfahren

mit einer breiten Vielfalt von Aspekten im Mittelpunkt.

Dabei wird großer Wert auf die Anschaulichkeit der Vorgehensweise und Ergebnisse gelegt. Dazu werden einige wichtige Beweise von Zusammenhängen und Sachverhalten geführt.

Der Leser bemerkt beim genauen Studium der Methoden zur analytischen und numerischen Lösung von Differentialgleichungen zahlreiche Details und interessante Aspekte. In der Literatur wird darauf kaum eingegangen, aber hier sollen diese nicht unerwähnt bleiben. Sie ergeben sich oft aus den in den CAS implementierten Verfahren. Letztere sind aber aufgrund der ständigen Weiterentwicklung von Hard- und Software stets aufs Neue zu testen.

Die beigefügten Übungsaufgaben dienen der Wiederholung und Vervollständigung des behandelten Stoffes sowie der Anregung zu weiterführenden Betrachtungen auf diesem interessanten Gebiet. Zu einigen sind die Lösungen sogar in den angeführten Beispielen enthalten.

Anregungen, nützliche Hinweise und Verbesserungsvorschläge zu diesem Buch sind willkommen und erreichen mich unter meiner E-Mail-Adresse

`werner.neundorf@tu-ilmenau.de`

Werner Neundorf

Ilmenau, 2013

Inhaltsverzeichnis

1	Differentialgleichungen in der Praxis	1
1.1	Elektrotechnik	1
1.2	Mechanik	4
1.3	Thermodynamik und Chemie	7
1.4	Physik	8
1.5	Mathematik	21
1.6	Biologie und Populationsdynamik	28
1.7	Medizin	33
1.8	Weitere Beispiele	35
1.8.1	Skalare gDGl 1. Ordnung	35
1.8.2	Systeme von zwei gDGl 1. Ordnung	46
1.8.3	Systeme von n gDGl 1. Ordnung	50
1.8.4	Skalare gDGl 2. und höherer Ordnung	52
2	Differentialgleichungen mit Maple	59
2.1	Analytische Lösung von Differentialgleichungen	60
2.2	Lösungsdarstellung von Differentialgleichungen	68
2.2.1	Richtungsfeld der Differentialgleichung	69
2.2.2	Numerische Lösung	75
2.2.3	Richtungsfeld und numerische Lösung	97
2.3	Definitionsbereiche von Lösungen	146
2.4	Eindeutige Lösung von Anfangswertproblemen	203
2.5	Mehrdeutige Lösung von Anfangswertproblemen	210
2.6	Systeme gewöhnlicher Differentialgleichungen	253
2.7	Spezielle gewöhnliche Differentialgleichungen	297

3	Differenzen, Differenzenquotienten, Taylor-Reihen	333
3.1	Differenzen	333
3.2	Dividierte Differenzen	336
3.3	Approximation von Ableitungen mit Differenzenquotienten	340
3.3.1	Einfache Differenzenquotienten	340
3.3.2	Symmetrische Quotienten für die 1. und 2. Ableitung	341
3.3.3	Symmetrische Differenzenquotienten bis zur 6. Ableitung	341
3.3.4	Differenzenquotienten für die 1. Ableitung	342
3.3.5	Differenzenquotienten für die 2. und höhere Ableitungen	343
3.3.6	Berechnung von Ableitungen	343
3.4	Taylor-Reihen und Ableitungen	353
4	Einführung in die Numerik	355
4.1	Grundbegriffe	355
4.2	Das Eulersche Polygonzugverfahren	378
4.3	Erste Fehlerbetrachtungen	392
5	Explizite Einschrittverfahren	395
5.1	Das Konzept der Einschrittverfahren	395
5.2	Runge-Kutta-Verfahren	400
5.2.1	Parameterschemata	421
5.3	Konvergenz von Einschrittverfahren	431
5.4	Schrittweitensteuerung	440
5.4.1	Schrittweitenkontrolle mittels lokalen Diskretisierungsfehler	440
5.4.2	Schrittweitenkontrolle mittels Einbettung	454
5.4.3	Schrittweitenkontrolle mittels Schrittkenzahl	461
5.4.4	Schrittweitensteuerung mit Grob- und Feinrechnung	464
5.4.5	Variable Schrittweitensteuerung	484
6	Lineare Mehrschrittverfahren	497
6.1	Das Konzept der linearen Mehrschrittverfahren	497
6.1.1	Adams-Bashforth-Verfahren	501
6.1.2	Adams-Moulton-Verfahren	507
6.1.3	Prädiktor-Korrektor-Verfahren	511

6.2	Die Ordnung linearer Mehrschrittverfahren	517
6.3	Homogene lineare Differenzgleichungen	523
6.4	Konsistenz und Konvergenz von Mehrschrittverfahren	528
7	Stabilität von Schrittverfahren	549
7.1	Absolute Stabilität	549
7.1.1	Stabilität und Stabilitätsgebiet von Einschrittverfahren	551
7.2	Stabilität von Mehrschrittverfahren	568
7.2.1	Stabilitätsgebiet von Mehrschrittverfahren	569
7.3	Integration steifer Systeme	606
7.3.1	Systeme mit ausgeprägter Steifheit	643
7.3.2	Steife Systeme und Linearisierung	667
8	Implizite Einschrittverfahren	673
8.1	Motivation für implizite Schrittverfahren	673
8.2	Das implizite Euler-Verfahren	675
8.3	Parameterschemata	682
8.4	Implementation ausgewählter impliziter RKV	691
8.5	Rosenbrock-Wanner-Verfahren	697
8.5.1	Einfache ROW-Verfahren	700
8.5.2	Eingebettete ROW-Verfahren	705
9	Übungsserien	719
	Index	763
	Literaturverzeichnis	767

Kapitel 1

Differentialgleichungen in der Praxis

Viele praktische Beispiele führen auf die Anwendung von Differentialgleichungen. Aus der Literatur zitieren und erläutern wir zahlreiche Modelle. Ihre Analyse fällt in das Gebiet der Theorie der gewöhnlichen Differentialgleichungen (gDGL).

Hier soll nur ein kleiner Überblick zur Darstellung und Simulation verschiedenartiger Sachverhalte durch Differentialgleichungen auf den Gebieten Technik, Physik, Chemie, Mathematik, Biologie, Ökonomie, Ökologie und Medizin gegeben werden.

Einige Beispiele werden ausführlicher beschrieben. Dabei soll der Leser erfahren, wie wirkungsvoll das Kalkül der Differentialgleichungen in die Modellierung einbezogen wird und wie nützlich, notwendig und zugleich elegant es ist.

1.1 Elektrotechnik

1. Elektrische Schaltung [94]

Aus der Beziehung für den Spannungsabfall berechnet man die Stromstärke.

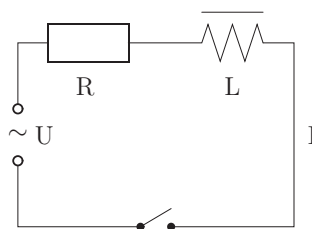


Abb. 1.1

Datei *bss1.pic*,
Schaltkreis mit
Widerstand und Spule

$$U(t) = U_0 \sin(\omega_0 t), \quad t_0 = 0 \leq t \leq T, \quad I(t_0) = 0.$$

Die Stromstärke ist Lösung der gDGL

$$U(t) = \underbrace{RI(t)}_{\text{Ohmscher}} + L \underbrace{\frac{dI(t)}{dt}}_{\text{Induktiver}}. \quad (1.1)$$

Spannungsabfall

Die allgemeine Lösung für die Stromstärke lautet

$$I(t) = e^{-\int \frac{R}{L} dt} \left(c_0 + \int \frac{R}{L} e^{\int \frac{R}{L} d\tau} dt \right), \quad (1.2)$$

wobei c_0 aus der Anfangsbedingung (AB) zu ermitteln ist.

Die Lösung im speziellen Fall

$$R = R_0 = \text{const}, \quad L = L_0 = \text{const}, \quad U(t) = U_0 \sin(\omega_0 t), \quad U_0 = \text{const}, \quad t_0 = 0$$

ist

$$I(t) = \frac{U_0 \omega_0 L_0}{R_0^2 + \omega_0^2 L_0^2} e^{-\frac{R_0}{L_0} t} + \frac{U_0}{\sqrt{R_0^2 + \omega_0^2 L_0^2}} \sin(\omega_0 t - \alpha) \quad \text{mit} \quad \alpha = \arctan\left(\frac{\omega_0 L_0}{R_0}\right).$$

2. Einschalten eines Stromkreises [81]

Die Anwendung der Kirchhoffschen Gesetze führt auf die gDGL

$$\frac{dU}{dt} + \frac{1}{RC} U = \frac{1}{RC} U_0, \quad I(t) = C \frac{dU(t)}{dt}. \quad (1.3)$$

Die Transformation der Variablen liefert eine gDGL 1. Ordnung in der Form

$$\dot{x} + f x = g(\tau), \quad x = x(\tau), \quad \tau \text{ transformierte Zeit.}$$

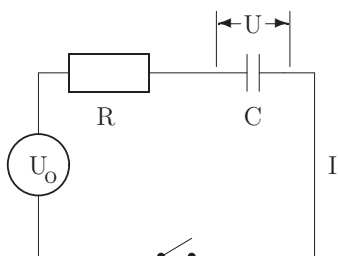


Abb. 1.2

Datei *bss2.pic*,
Schaltkreis mit
Widerstand und Kondensator

Hängen Kapazität C und/oder Widerstand R auch vom Strom I ab (nichtlineare Kapazitäten oder Widerstände), so gilt

$$\dot{x} + f(\dot{x}) x = g(\tau, \dot{x}).$$

Diese gDGL ist i. Allg. nicht geschlossen integrierbar.

3. Nichtlinearer ungedämpfter Schwingkreis [81]

Man betrachtet den ungedämpften Schwingkreis mit nichtlinearer Induktivität.

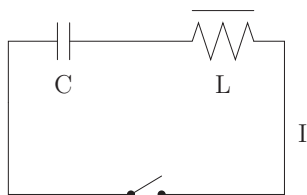


Abb. 1.3

Datei *bss3.pic*,
Schaltkreis mit
Kondensator und Spule

Wenn der Zusammenhang zwischen Strom dem $I(t)$ und dem Fluss $\psi(t)$ die Kennlinie $I = a\psi + b\psi^3$ besitzt, so erhält man die gDGL

$$\frac{d^2\psi}{dt^2} + \frac{a}{C}\psi + \frac{b}{C}\psi^3 = 0. \quad (1.4)$$

Nach Normierung mit $\Psi = \max \psi(t) = \psi(0)$ durch

$$x(t) = \frac{\psi(t)}{\Psi}, \quad \tau = \sqrt{\frac{a}{C}}t, \quad \lambda = \Psi^2 \frac{a}{b},$$

bekommt man die gDGL 2. Ordnung

$$\ddot{x} + x + \lambda x^3 = 0,$$

die nicht geschlossen lösbar ist.

Das Anlegen einer sinusförmigen Erregerschwingung an diesen Schwingkreis liefert eine erzwungene Schwingung mit der normierten gDGL (Duffingsche DGL)

$$\ddot{x} + x + \lambda x^3 = k \cos(\alpha\tau).$$

4. Relaxationsschwingungen [81]

Zahlreiche Probleme der nichtlinearen Elektrotechnik (ungedämpfte Schwingungen, Vibrator, Multivibrator) führen zu der normierten gDGL 2. Ordnung

$$\ddot{y} - \varepsilon(1 - y^2)\dot{y} + y = 0, \quad t \geq 0, \quad \varepsilon \geq 0, \quad (1.5)$$

der Van der Pol DGL.

Für $\varepsilon = 0$ liefert sie bei AB y_0, \dot{y}_0 an der Stelle $t = 0$ die harmonische Schwingung $y(t) = \dot{y}_0 \sin(t) + y_0 \cos(t)$ mit der Periode $T = 2\pi$.

Wächst $\varepsilon > 0$ an, so verzerrt sich die Schwingung immer mehr und die Periode vergrößert sich.

Bei $\varepsilon \gg 1$ (z.B. $\varepsilon = 100, 1000$) beschreibt die gDGL Relaxationsschwingungen. Dabei wechseln sich Zeiten sehr langsamer Bewegung mit solchen sehr schneller ab. Der Parameter ε legt auch die Einschwingdauer fest.

Bei erregten Schwingkreisen steht ein zusätzlicher Term auf der rechten Seite der gDGL.

5. Mehrmaschige elektrische Netzwerke [81, 111]

Das zweimaschige gekoppelte Netzwerk für die normierten Funktionen $x_1(t), x_2(t)$ hat die Form

$$\begin{aligned} \ddot{x}_1 + \alpha_1 \dot{x}_1 + \varepsilon_1(1 - x_2^2)\dot{x}_2 + x_1 &= 0, \\ \ddot{x}_2 + \alpha_2 \dot{x}_2 + \varepsilon_2(1 - x_1^2)\dot{x}_1 + x_2 &= 0. \end{aligned} \quad (1.6)$$

Eine geschlossene Lösung ist nicht möglich. Insbesondere sind Näherungslösungen für große Parameterwerte von $|\varepsilon_1|, |\varepsilon_2|$ gesucht.

6. Dynamische Kennlinie des Asynchronmotors [35]

Bei Asynchronmotoren ist die Drehzahl stark von der Belastung abhängig. Für schwankende Motorleistung ist es zur Berechnung von Torsionsschwingungen notwendig, den Zusammenhang zwischen Motormoment und Winkelgeschwindigkeit zu bestimmen.

Mit den normierten Größen $u(t)$, $v(t)$, t dimensionslose Zeit, erhält man das nicht geschlossen lösbare DGI-System 1. Ordnung (SysgDGI)

$$\begin{aligned}\beta\dot{u} &= x - u + xv, & u(0) &= u_0, & \beta &\in \mathbb{R}^+, & x &= x(t), \\ \beta\dot{v} &= -xu - v, & v(0) &= v_0.\end{aligned}\tag{1.7}$$

7. Weitere Beispiele

(a) Spannungsstabilisatorschaltung, Ferroresonanzstabilisator [81]

$$\ddot{y} - \delta\dot{y} + ay + by^9 + \alpha y = -\Gamma \cos(t + \phi), \quad y = y(t).\tag{1.8}$$

(b) Frequenzteilerschaltung

$$\ddot{y} - \varepsilon(1 - y^2 - \dot{y}^2)\dot{y} + y + 2By\dot{y} = B \sin(2t).\tag{1.9}$$

(c) System der Drei-Wellen-Wechselwirkung

$$\begin{aligned}\dot{a}_0 &= a_0 + 2a_0a_1^2 \sin(\vartheta), \\ \dot{a}_1 &= -a_1 - a_0^2a_1 \sin(\vartheta), \\ \dot{a}_2 &= 2(a_1^2 - a_0^2) - 2\delta + 2(2a_1^2 - a_0^2) \cos(\vartheta).\end{aligned}\tag{1.10}$$

(d) Rössler-System

$$\begin{aligned}\dot{X} &= -Y - Z, \\ \dot{Y} &= X + aY, \\ \dot{Z} &= bX - cZ + XZ.\end{aligned}\tag{1.11}$$

1.2 Mechanik

1. Vibrationen von Membranen und Stäben [49, 96]

Diese führen auf gDGI vom Duffingschen Typ

$$\ddot{y} - \varepsilon \dot{y} - \beta y + \alpha y^2 = f \cos(\omega t).\tag{1.12}$$

Die gDGI stellt eine erste Näherung für die Schwingung dar und wird für verschiedene Parameterkonstellationen $\varepsilon, \beta, \alpha, f, \omega$ analysiert.

2. Schwingungen von Eisenbahnwagen [63]

$$\begin{aligned}I\ddot{x} + 2lf_T \left(\frac{l\dot{x}}{v} + \frac{r_1 - r_2}{2r_0} \right) + K_x x &= 0, \\ M\ddot{y} + 2f_L \left(\frac{\dot{y}}{v} - x \right) K_y y + F_g(y - \delta) &= 0.\end{aligned}\tag{1.13}$$

3. Bewegung eines Himmelskörpers [97]

Die Bewegung einer Rakete oder eines Satelliten im Schwerfeld zweier großer Himmelskörper führt auf ein restringiertes Dreikörperproblem.

Bsp.: Erde – Mond – Rakete

Sonne – Erde – Rakete

Sonne – Saturn – Saturnmond

$$\begin{aligned} y_1'' &= 2y_2' + y_1 - \frac{\mu^*(y_1 + \mu)}{r_1^3} - \frac{\mu(y_1 - \mu^*)}{r_2^3}, \\ y_2'' &= -2y_1' + y_2 - \frac{\mu^*y_2}{r_1^3} - \frac{\mu y_2}{r_2^3} \end{aligned} \quad (1.14)$$

mit $\mu^* = 1 - \mu$, $0 < \mu < 1$ und $r_1 = \sqrt{(y_1 + \mu)^2 + y_2^2}$, $r_2 = \sqrt{(y_1 - \mu)^2 + y_2^2}$.

4. Orientierung von Telekommunikationssatelliten [109]

Dabei geht es um die positive Orientierung der Antenne des Satelliten auf seiner elliptischen Umlaufbahn durch die Vorgabe einer geeigneten Anfangsorientierung.

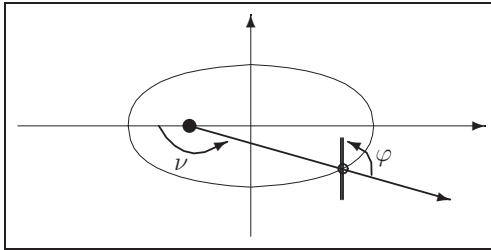


Abb. 1.4

Datei *bss7.pic*,
Satellitenumlauf
um die Erde mit
Antennenorientierung

$$(1 + e \cos(\nu)) \frac{d^2\varphi}{d\nu^2} - 2e \sin(\nu) \frac{d\varphi}{d\nu} + \alpha \sin(\varphi) \cos(\varphi) = 2e \sin(\nu), \quad (1.15)$$

wobei e Exzentrizität, α Konstante zur Beschreibung der Massenverteilung, ν Umlaufwinkel, $\varphi = \varphi(\nu)$ gesuchte Abweichung vom Leitstrahl.

5. Pendel mit Reibung [7]

$$\ddot{x} + k\dot{x} + x = 0, \quad x(0), \quad \dot{x}(0) \text{ geg.}, \quad k \geq 0 \text{ Reibungskoeffizient.} \quad (1.16)$$

Die Überführung in ein SysgDGL bedeutet

$$\begin{aligned} \dot{x} &= y, & x(0) \text{ geg.}, \\ \dot{y} &= \ddot{x} = -k\dot{x} - x = -x - ky, & y(0) = \dot{x}(0) \text{ geg.}, \end{aligned} \quad (1.17)$$

die Notation in Matrixform ist

$$\dot{X} = AX = \begin{pmatrix} 0 & 1 \\ -1 & -k \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \quad \text{trace}(A) = -k, \quad \det(A) = 1.$$

Das Eigenwertproblem zur Matrix A lautet

$$\lambda(A): 0 = \det(A - \lambda I) = \lambda^2 + k\lambda + 1.$$

Falls $|k| > 2$, dann sind die Eigenwerte λ_i reell, negativ und verschieden.

Sei

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

Die Eigenvektoren sind

$$x_1 = \begin{pmatrix} -3 \\ 1 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 1 \\ -3 \end{pmatrix}, \quad U = (x_1, x_2) \text{ regulär, } AU = U\Lambda.$$

Die Transformation auf die Diagonalform ist möglich.

$$\begin{aligned} X &= UZ, \\ \dot{X} &= U\dot{Z} = AX = AUZ, \\ U\dot{Z} &= U\Lambda Z, \\ \dot{Z} &= \Lambda Z, \\ \dot{Z}_1 &= \lambda_1 Z_1, \quad \text{entkoppeltes SysgDGL,} \\ \dot{Z}_2 &= \lambda_2 Z_2, \\ Z_1(t) &= e^{\lambda_1 t} Z_1(0), \quad \text{Lösungen,} \\ Z_2(t) &= e^{\lambda_2 t} Z_2(0). \end{aligned}$$

Beispiel 1.1

$$X_0 = (x(0), y(0))^T = (2, 2)^T, \quad k = \frac{10}{3}, \quad \lambda_1 = -\frac{1}{3}, \quad \lambda_2 = -3$$

1. Darstellung der Lösungen als Phasenkurven im Raum der Eigenvektoren

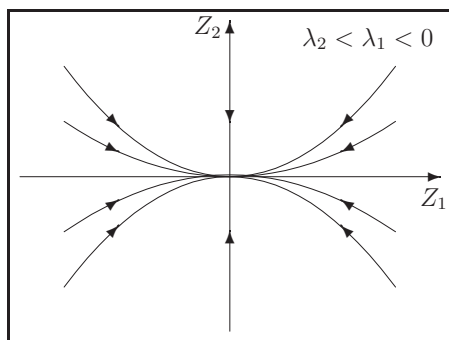


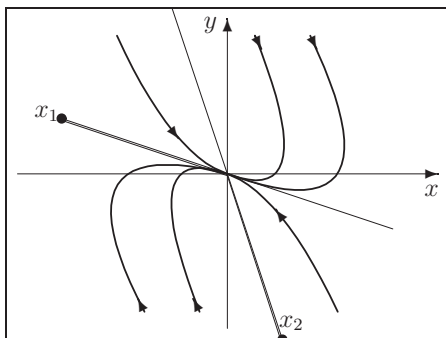
Abb. 1.5

Datei *bss10.pic*,
Phasenkurven
 $(Z_1(t), Z_2(t))$, $t \geq 0$,
zu verschiedenen
AB $(Z_1(0), Z_2(0))$

2. Darstellung der Lösungen als Phasenkurven im Raum (x_1, x_2)

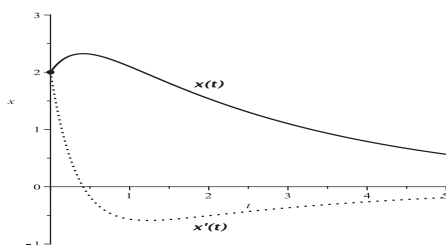
$$U^{-1} = \frac{1}{8} \begin{pmatrix} -3 & -1 \\ -1 & -3 \end{pmatrix}, \quad Z_0 = U^{-1}X_0 = \begin{pmatrix} -1 \\ -1 \end{pmatrix},$$

$$X = UZ = \begin{pmatrix} -3 & 1 \\ 1 & -3 \end{pmatrix} \begin{pmatrix} e^{\lambda_1 t} Z_1(0) \\ e^{\lambda_2 t} Z_2(0) \end{pmatrix} = \begin{pmatrix} 3e^{-t/3} - e^{-3t} \\ -e^{-t/3} + 3e^{-3t} \end{pmatrix}.$$

**Abb. 1.6**

Datei *bss10.pic*,
Phasenkurven
 $(X_1(t), X_2(t))$, $t \geq 0$,
zu verschiedenen
AB $(X_1(0), X_2(0))$

Bei gegebener Anfangsauslenkung kommt es durch den Anstoß zu einem kurzzeitigen Überschwingen des Pendels. Danach wird die Pendelbewegung wegen der Reibung jedoch stark gedämpft. Die Lösung $X_1(t) = x(t) = 3e^{-t/3} - e^{-3t}$, $X_1(0) = 2$, strebt mit wachsender Zeit asymptotisch gegen Null.

**Abb. 1.7**

Datei *bss10a.eps*,
Kurven
 $x(t), \dot{x}(t) = y(t)$, $t \in [0, 5]$,
zur AB $(x(0), y(0)) = (2, 2)$

1.3 Thermodynamik und Chemie

1. Lorenz-Modell der Wärmeströmung in einer Flüssigkeitsschicht [63, 65, 99]

Eine Flüssigkeitsschicht wird von einer Seite erwärmt und von der gegenüberliegenden Seite gekühlt. Die daraus resultierende konvektive Bewegung ist von Interesse. Das SysgDGL entsteht aus den Navier-Stokes-Gleichungen:

- y_1 konvektive Durchmischung,
- y_2 horizontale Temperaturveränderung,
- y_3 vertikale Temperaturveränderung,

$$\begin{aligned} y_1' &= -\sigma y_1 + \sigma y_2, & y_1(0) &= y_{10}, \\ y_2' &= -y_1 y_3 + r y_1 - y_2, & y_2(0) &= y_{20}, \\ y_3' &= y_1 y_2 - b y_3, & y_3(0) &= y_{30}, \end{aligned} \tag{1.18}$$

wobei die Prandtl-Zahl σ , reduzierte Rayleigh-Zahl r und Wellenzahl b positive Parameter sind.

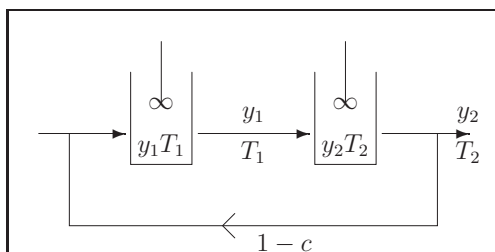
2. Schwingungen im Wärmeaustauscher [109]

Durch Wärmeunterschiede in einem Durchlaufsystem können Röhren unter Spannung, andere unter Druck stehen. Dadurch entstehen auch Schwingungen.

$$\ddot{y} + \varepsilon_1 \dot{y} + \varepsilon_2 y + y^3 + \dot{y}^3 = 0. \quad (1.19)$$

3. Chemische Reaktoren [63]

Die Beschreibung einer Kaskade von zwei chemischen Tankreaktoren mit Rührwerk und Rückkopplung liefert ein System von vier gDGl 1. Ordnung.

**Abb. 1.8**

Datei *bss8.pic*,
Kaskade von
2 Tankreaktoren

y_i Konzentration, T_i Temperatur, $0 < c \leq 1$ Parameter.

1.4 Physik

1. Freier Fall

Angenommen das Anfangswertproblem (AWP)

$$y' = a\sqrt{y}, \quad y = y(t), \quad y(0) = c, \quad a = \text{const} > 0, \quad c = \text{const} \geq 0, \quad (1.20)$$

beschreibt den freien Fall eines Körpers. Die Funktion $y(t)$ sei also der zur Zeit t zurückgelegte Weg.

- Welche Lösung erhält man im Fall $c = 0$?
Gibt es noch eine zweite Lösung der gDGl mit der AB $y(0) = 0$?
- Mit welcher AB kann ein Einschrittverfahren einen freien Fall berechnen?
- Gegeben sei $\tau \geq 0$ und die Funktion

$$z(t) = \begin{cases} 0, & \text{falls } 0 \leq t \leq \tau, \\ \frac{1}{4}c^2(t - \tau)^2, & \text{falls } t > \tau. \end{cases} \quad (1.21)$$

Erfüllt $z(t)$ die gDGl?

Welche Auswirkung hat die Antwort auf das Problem der Eindeutigkeit der Lösung? Worin liegt die Ursache für diese Situation?

Hinweis: Man untersucht die Glattheit der rechten Seite $f(t, y)$ der gDGl.

2. Bewegungsgleichung eines Körpers in laminarer Strömung

Die Bewegungsgleichung eines Körpers mit der Masse m in einer zähen Flüssigkeit (laminare Strömung) mit dem Stokes'schen Widerstandsgesetz $R(v) = K_1 v$ (K_1 Materialeigenschaften des Mediums/Geometrie des Körpers) ergibt sich aus dem Kraftgesetz

$$m\dot{v} = A + G + R(v), \quad G \text{ Erdanziehung, } A \text{ Auftrieb.} \quad (1.22)$$

Die abgeleitete Modellgleichung für die Geschwindigkeit $v(t)$ ist

$$\dot{v} = c + k_1 v, \quad c > 0, \quad v(0) \text{ gegeben.} \quad (1.23)$$

Man diskutiere eine Lösung zu verschiedenen Startwerten $v(0)$.

Es sei $v(0) = 0$, $k_1 < 0$, $c > 0$. Gibt es eine Gleichgewichtsgeschwindigkeit $v_\infty = v(t = \infty) = \text{const}$?

3. Bewegungsgleichung eines Körpers in turbulenter Strömung

Die Bewegungsgleichung eines Körpers mit der Masse m in einer turbulenten Strömung (Neigung zur Wirbelbildung) mit dem Newtonschen Widerstandsgesetz $R(v) = K_2 v^2$ (K_2 Materialeigenschaften/Geometrie) ergibt sich aus dem Kraftgesetz

$$m\dot{v} = A + G + R(v). \quad (1.24)$$

Die abgeleitete Modellgleichung für die Geschwindigkeit $v(t)$ ist

$$\dot{v} = c + k_2 v^2, \quad c > 0. \quad (1.25)$$

Es sei $v(0) = v_0$, $k_2 < 0$, $c > 0$.

Man diskutiere die Lösung bei verschiedenen Startwerten.

Gibt es eine Gleichgewichtsgeschwindigkeit $v_\infty = v(t = \infty) = \text{const}$?

4. Bewegungsgleichung für den freien Fall

Die Bewegungsgleichung für den freien Fall (ohne Reibung) mit der von der Höhe $z(t)$ abhängigen Erdbeschleunigung $g(z)$ lautet

$$\ddot{z} = -g(z). \quad (1.26)$$

Nach dem Newtonschen Gravitationsgesetz gilt $g(z) = g_0 z_0^2 / z^2$, wobei $g_0 = g(z_0)$ den Wert der Erdbeschleunigung auf der Höhe z_0 bezeichnet.

Für kleine Fallhöhen kann man drei Approximationen der rechten Seite der gDGL anwenden.

(1) $g_0 = g = 9.806 \text{ ms}^{-2}$ Normwert, $r_E = 6.375 \cdot 10^6 \text{ m}$ Erdradius, $z_0 = z$,

$$\ddot{z} = -g. \quad (1.27)$$

$$(2) \quad g(z) \approx g - 2\frac{g}{r_E}(z - r_E) = 3g - 2\frac{g}{r_E}z,$$

$$\ddot{z} = -3g + 2\frac{g}{r_E}z. \quad (1.28)$$

$$(3) \quad z_0 = r_E, \quad g_0 = g,$$

$$\ddot{z} = -\frac{gr_E^2}{z^2}. \quad (1.29)$$

Die AB sind jeweils $z(0) = r_E + H$, $\dot{z}(0) = 0$, $H > 0$ Fallhöhe.

- Welche konstante Funktion ist Lösung der Modellgleichung (1.28)?
Man finde ihre allgemeine Lösung.
- Man berechne die Fallzeit t_F und Endgeschwindigkeit $\dot{z}(t_F)$ bei einer Fallhöhe H und gegebenen AB für die Modellgleichungen (1.27), (1.28).
- Welchen Einfluss hat die Linearisierung des Modells?

5. Bewegungsgleichungen, Schwingungen, Pendel (parametrische gDGL)

(a) Schwingungsgleichung ohne Einwirkung äußerer Kräfte

$$y'' + 2hy' + k^2y = 0. \quad (1.30)$$

(b) GDGL der erzwungenen Schwingung $y(t)$ mit Einwirkung äußerer Kräfte
Man untersuche $y(t)$ für $t \rightarrow \infty$.

$$y'' + 2\rho y' + \omega^2 y = a \sin(\omega_1 t), \quad y(0) = 0, \quad y'(0) = \omega. \quad (1.31)$$

(c) Van der Polsche DGL [7]

$$\ddot{y} - \varepsilon(1 - y^2)\dot{y} + y = 0, \quad y(0) = y_0, \quad \dot{y}(0) = \dot{y}_0, \quad t \geq 0, \quad \varepsilon \geq 0. \quad (1.32)$$

Der Parameter ε legt die Einschwingdauer fest.

Die allgemeine Lösung für $\varepsilon = 0$ ist $y(t) = \dot{y}_0 \sin(t) + y_0 \cos(t)$.

Die Van der Polsche DGL mit periodischer Erregung lautet

$$\ddot{y} - \varepsilon(1 - y^2)\dot{y} + y = k \cos(\Omega t). \quad (1.33)$$

(d) Rotation eines Pendels mit der Winkelgeschwindigkeit ω

$$\ddot{y} + \sin y - \frac{1}{2}\omega^2 \sin(2y) = 0, \quad y(0) = \frac{\pi}{4}, \quad \dot{y}(0) = 0, \quad \omega \in [0, 10]. \quad (1.34)$$

(e) Pendel mit AB [112]

$$y'' + k^2 y = 0, \quad k > 0, \quad y(0) = y_0, \quad y'(0) = y'_0. \quad (1.35)$$

Lösung: $y = \frac{y'_0}{k} \sin(kx) + y_0 \cos(kx)$,

$$y = \frac{y'_0}{k} \sin(kx), \quad \text{falls } y_0 = 0,$$

$$y = \sin(x), \quad \text{falls } y_0 = 0, \quad y'_0 = 1, \quad k = 1,$$

$$y = c_1 \sin(kx) + y_0 \cos(kx), \quad \text{falls die AB } y'(0) \text{ fehlt.}$$

(f) Fadenpendel, allgemeine Darstellung [112]

$$y'' + ay' + by = 0, \quad D = a^2 - 4b \text{ Diskriminante.} \quad (1.36)$$

Lösung:

(1) $D > 0$

$$y = c_1 e^{k_1 x} + c_2 e^{k_2 x}, \quad k_1 = -\frac{1}{2}(a - \sqrt{D}), \quad k_2 = -\frac{1}{2}(a + \sqrt{D}), \quad (1.37)$$

z.B. falls $b < 0$, dann $D > 0$.

(2) $D = 0$

$$y = c_1 e^{k_1 x} + c_2 x e^{k_1 x}, \quad k_1 = -\frac{a}{2}, \quad (1.38)$$

z.B. falls $b = 0$, $a = 0$, dann $D = 0$.

(3) $D < 0$

$$y = e^{-ax/2} [c_1 \sin(\sqrt{|D|}x/2) + c_2 \cos(\sqrt{|D|}x/2)], \quad (1.39)$$

$$k_1 = -\frac{a}{2} + \frac{1}{2}\sqrt{-D} \, i, \quad k_2 = -\frac{a}{2} - \frac{1}{2}\sqrt{-D} \, i,$$

z.B. falls $b \gg 0$, dann $D < 0$; $a > 0 \rightarrow$ gedämpftes Pendel,
 $a < 0 \rightarrow$ erregtes Pendel.

(g) Bewegungsgleichung

$$\ddot{y} + f(y) = 0, \quad y(0) = 2, \quad \dot{y}(0) = 0, \quad f(z) = \begin{cases} -1, & \text{falls } z \leq -1, \\ z, & \text{falls } -1 < z < 1, \\ 1, & \text{falls } 1 \leq z. \end{cases} \quad (1.40)$$

(h) Duffingsche Bewegungsgleichungen, Vibration, $y = y(t)$

$$\ddot{y} + y + \lambda y^3 = 0, \quad y(0) = y_0, \quad \dot{y}(0) = \dot{y}_0, \quad \lambda \in \mathbb{R}^+,$$

$$\ddot{y} + y + \lambda y^3 = k \cos(\alpha t), \quad \lambda \in \mathbb{R}^+, \quad k, \alpha \in \mathbb{R},$$

$$\ddot{y} + \varepsilon \dot{y} - \beta y + \alpha y^3 = f \cos(\omega t),$$

$$\ddot{y} + 0.7\dot{y} + y^3 = 0.75 \cos(t), \quad y(0) = \dot{y}(0) = 0, \quad (1.41)$$

$$\ddot{y} + 0.2\dot{y} + y|y| = 1.5 \cos(2t) + 0.5, \quad y(0) = \dot{y}(0) = 0,$$

$$\ddot{y} + \dot{y} - \frac{1}{6}y^3 = 0.8 \sin(0.27\omega t), \quad y(0) = \dot{y}(0) = 0, \quad \omega = 0.92845.$$

(i) Bewegungsgleichung mit quadratischer Dämpfung

$$\ddot{y} + \frac{1}{2}\dot{y}^2 + y = 0, \quad y(0) = 1, \quad \dot{y}(0) = 0. \quad (1.42)$$

(j) Emdens Bewegungsgleichungen (Lane-Emden-Gleichung, Emdem-Fowler-Gleichung) [43]

$$\frac{d}{dt}(t^2 \dot{y}) + t^2 y^n = 0, \quad y(0) = 1, \quad \dot{y}(0) = 0, \quad n = 2, 3, 5, \dots,$$

$$\frac{1}{t^2} \frac{d}{dt}(t^2 \dot{y}) + f(y) = \ddot{y} + \frac{2}{t}\dot{y} + f(y) = 0, \quad y(0) = 1, \quad \dot{y}(0) = 0, \quad (1.43)$$

$$f(y) = \sin(y) \text{ bzw. } \cos(y).$$

- (k) Liouvillesche DGI [116]

$$\frac{d}{dt}(t^2 \dot{y}) + t^2 e^y = 0, \quad y(0) = \dot{y}(0) = 0. \quad (1.44)$$

- (l) Mathieus DGI für ein Pendel mit zeitabhängiger Länge [3, 121]

$$\ddot{y} + (a - 2q \cos(2t))y = 0, \quad y(0) = 1, \quad \dot{y}(0) = 0, \quad (1.45)$$

$a = 1.0, q = 0.1,$ aufschaukelnde Schwingung,
 $a = 1.2, q = 0.1,$ gedämpfte Schwingung.

Man kann einen (a, q) -Bereich finden, in dem die Lösung beschränkt bleibt (Pendel-Diagramm).

- (m) Gedämpfte Schwingung einer Membran, Besselsche DGI

$$t^2 y'' + t y' + (t^2 - k^2)y = 0, \quad k \in \mathbb{Z}. \quad (1.46)$$

Lösung: $y = J_0(t)$, falls $k = 0$, $y(0) = 1$, $y'(0) = 0$, Besselfunktion 1. Art,
 $y = J_1(t)$, falls $k = 1$, $y(0) = 0$, $y'(0) = 0.5$.

- (n) Resonanz zwischen Eigenschwingung und Störung

$$y'' + 6y' + 5y = 3e^{-t}. \quad (1.47)$$

- (o) Abrollen einer Kugel auf der Falllinie eines parabolischen Zylinders

$$\ddot{y} + \frac{y}{2025 + y^2}(31532 + \dot{y}^2) = 0, \quad y(0) = 30, \quad \dot{y}(0) = 0. \quad (1.48)$$

- (p) Rollpendel [64, 112]

$$\left(\frac{\Theta}{m} + (r^2 + s^2 - 2rs \cos(\phi)) \right) \ddot{\phi} + (r\dot{\phi}^2 + g) s \sin(\phi) = 0. \quad (1.49)$$

- (q) Hamiltonsche Bewegungsgleichung für das Hénon Heiles-Modell [99]

Das SysgDGI gleicht mathematisch einem nichtlinearen gekoppelten Oszillator und ist von Hénon und Heiles untersucht worden. Die Terme q_1, q_2 entsprechen linearen rücktreibenden Kräften (wie ein linearer Oszillator).

Die weiteren Terme sind nichtlineare Koppelglieder.

$$\begin{aligned} \dot{q}_1 &= p_1, & \dot{p}_1 &= -q_1 - 2q_1 q_2, \\ \dot{q}_2 &= p_2, & \dot{p}_2 &= -q_2 - q_1^2 + q_2^2, \end{aligned} \quad (1.50)$$

AB sind z.B. $q_1(0) = 0.5$, $q_2(0) = p_1(0) = p_2(0) = 0.1$.

- (r) Bewegungsgleichung für die biquadratische Hamilton-Funktion

$$\begin{aligned} \dot{q}_1 &= p_1, & \dot{p}_1 &= -\frac{1-r}{3}q_1^3 - q_1 q_2^2, & r &\in [0, 1], \\ \dot{q}_2 &= p_2, & \dot{p}_2 &= -\frac{1-r}{3}q_2^3 - q_1^2 q_2. \end{aligned} \quad (1.51)$$

Für wachsendes r wird das SysgDGI mehr und mehr chaotisch.

(s) Bewegungsgleichung für die Hamilton-Funktion *Toda lattice* [106]

$$\begin{aligned}\dot{a}_j &= a_j(b_j - b_{j+1}), \quad j = 1, 2 \quad (a_3 \equiv 0), \\ \dot{b}_1 &= -2a_1^2, \\ \dot{b}_2 &= 2(a_1^2 - a_2^2), \\ \dot{b}_3 &= 2a_2^2.\end{aligned}\tag{1.52}$$

(t) Schwingende Atwood-Maschine [99, 110]

$$\begin{aligned}(1 + \mu) \ddot{r} &= r(\dot{\Theta})^2 + \cos(\Theta) - \mu, \quad \mu \text{ Masse,} \\ r \ddot{\Theta} &= -2\dot{r}\dot{\Theta} - \sin(\Theta).\end{aligned}\tag{1.53}$$

(u) Chaotische Systeme mit Orbits (Homoklinen) [99]

Siehe auch Duffingsche Bewegungsgleichungen,

$$\begin{aligned}\ddot{u} - \frac{1}{2}u + u^2 + u^3 &= 0, \\ \ddot{u} + u^3 &= 0, \\ \ddot{u} + a\dot{u} + bu + cu^3 &= k_1 + k_2 \cos(\Omega t), \\ \text{Parametersatz z.B.} \\ a = 1, b = -10, c = 100, k_1 = 0, k_2 = 1.2, \Omega = 3.5, \\ \ddot{u} + a\dot{u} + b \sin(u) &= k \cos(\Omega t), \\ a = 0.2, b = 1, k = 1.1, \Omega = 0.8, \\ \ddot{u} + a\dot{u} + (1 + k \cos(\Omega t)) \sin(u) &= 0, \\ a = 0.15, k = 0.94, \Omega = 1.56.\end{aligned}\tag{1.54}$$

6. Fadenpendel [64, 112]

Wie verläuft die Bewegung eines Pendelkörpers unter den Vereinfachungen: keine Berücksichtigung des Luftwiderstandes, Masse des Fadens wird vernachlässigt, Pendelkörper ist Massepunkt (mathematisches Pendel).

Bekanntlich gilt nach Newton $F = m a$.

Für kleine Auslenkungen $\alpha(t)$ ergibt sich aus dem Newtonschen Grundgesetz folgende Gleichung für das mathematische Pendel (Kräfteparallelogramm)

$$m g \sin(\alpha(t)) = -m l \alpha''(t),\tag{1.55}$$

wobei l Fadenlänge,
 $l \alpha(t)$ Weg, den das Pendel zurücklegt = Länge des Kreisbogens,
 $a = l \alpha''(t)$ Beschleunigung des Pendelkörpers,
 g Erdbeschleunigung,
 $m g$ Gewichtskraft,
 F und α haben entgegengesetzte Richtungen.

Für kleine Winkel ist $\sin(\alpha) \approx \alpha$, sodass die einfache lineare gDGL 2. Ordnung

$$\alpha''(t) + \frac{g}{l} \alpha(t) = 0 \quad (1.56)$$

entsteht.

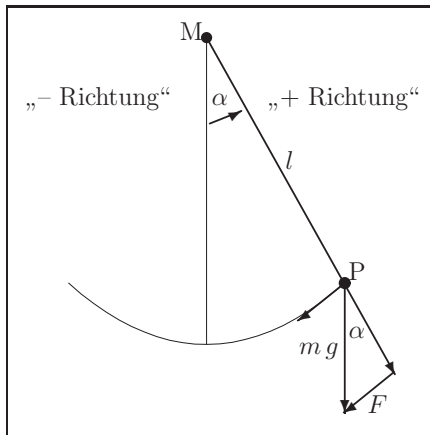


Abb. 1.9

Datei *bs17.pic*,
Fadenpendel
mit Parameter

- (a) Allgemeine Lösung der gDGL

Hinweis: Man findet die Basislösungen mittels des Ansatzes $e^{\lambda t}$.

Die beiden unabhängigen Lösungen sind $\sin(\sqrt{\frac{g}{l}}t)$, $\cos(\sqrt{\frac{g}{l}}t)$.

Die Lösungsmenge ist ein Vektorraum gebildet mittels der Linearkombination

$$\alpha(t) = c_1 \sin\left(\sqrt{\frac{g}{l}}t\right) + c_2 \cos\left(\sqrt{\frac{g}{l}}t\right), \quad c_{1,2} \in \mathbb{R}. \quad (1.57)$$

- (b) Berücksichtigung von AB

Es sollen zwei Varianten unterschieden werden.

- (c) Das Pendel wird nach Auslenkung um einen Winkel α_0 losgelassen.

Wenn zur Zeit $t = 0$ die AB $\alpha(0) = \alpha_0$ und $\alpha'(0) = 0$ bekannt sind, ergeben sich die Koeffizienten zu

$$c_1 = 0, \quad c_2 = \alpha_0.$$

Daraus folgt

$$\alpha(t) = \alpha_0 \cos\left(\sqrt{\frac{g}{l}}t\right).$$

- (d) Das Pendel wird aus der Lage $\alpha(0) = 0$ mit der Geschwindigkeit $v_0 = l\alpha'(0)$ angestoßen. Wenn zur Zeit $t = 0$ die AB $\alpha(0) = 0$ und $\alpha'(0)$ bekannt sind, ergeben sich die Koeffizienten zu

$$c_2 = 0, \quad c_1 = \sqrt{\frac{l}{g}} \alpha'(0).$$

Daraus folgt

$$\alpha(t) = \sqrt{\frac{l}{g}} \alpha'(0) \sin\left(\sqrt{\frac{g}{l}} t\right).$$

(e) Lösung der gDGI 2. Ordnung mittels Einschrittverfahren

Das AWP

$$\alpha''(t) = -\frac{g}{l} \alpha(t), \quad \alpha(0), \alpha'(0) \text{ gegeben, } t \in [0, T],$$

kann man mittels $\alpha'(t) = \beta(t)$ auf ein System von zwei gDGI 1. Ordnung transformieren.

$$\alpha'(t) = \beta(t), \quad \alpha(0) \text{ gegeben,}$$

$$\beta'(t) = \alpha''(t) = -\frac{g}{l} \alpha(t), \quad \beta(0) = \alpha'(0) \text{ gegeben.}$$

Nunmehr kann man auf das allgemein notierte SysgDGI

$$x' = f(t, x, y), \quad x(0) \text{ geg.,}$$

$$y' = g(t, x, y), \quad y(0) \text{ geg.,}$$

das numerische Einschrittverfahren von Euler (explizites Euler-Verfahren, Polygonzugverfahren (PZV)) anwenden.

$$\begin{aligned} x_{n+1} &= x_n + hf(t_n, x_n, y_n), \quad x_0 = x(0), \quad n = 0, 1, \dots, \quad t_n = nh, \quad h > 0, \\ y_{n+1} &= y_n + hg(t_n, x_n, y_n), \quad y_0 = y(0). \end{aligned} \quad (1.58)$$

Man vergleicht an den Integrationsstellen die Näherungslösung (x_n, y_n) mit der exakten Lösung $(x(t_n), y(t_n))$.

(f) Zwei Varianten der grafischen Darstellung

Beispiel und Parameter $\alpha(t) = \alpha_0 \cos\left(\sqrt{\frac{g}{l}} t\right)$, $\alpha'(t) = -\alpha_0 \sqrt{\frac{g}{l}} \sin\left(\sqrt{\frac{g}{l}} t\right)$, $\sqrt{\frac{g}{l}} = 2$, $\alpha_0 = 1$, $\alpha'_0 = 0$, $t \in [0, T] = [0, \pi]$, $N = 40$, $h = \frac{T}{N}$.

Vergleichende Grafik für die exakten Lösungen $\alpha(t)$, $\alpha'(t) = \beta(t)$ und Näherungen (Polygonzug) $\{\alpha_0, \alpha_1, \alpha_2, \dots\}$, $\{\alpha'_0, \alpha'_1, \alpha'_2, \dots\}$.

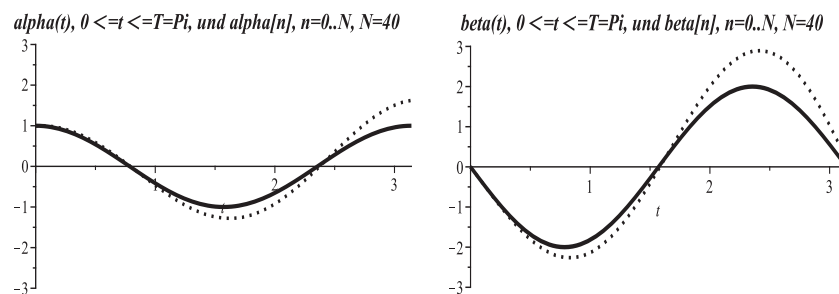


Abb. 1.10 Dateien *ependel_05sw, 06sw.eps*,
 Lösungen $\alpha(t)$, $\alpha'(t) = \beta(t)$ in $[0, \pi] \times [-3, 3]$,
 sowie entsprechende Näherungslösungen (gepunktet) mit PZV

Die vergleichende Grafik für die Phasenkurven betrifft die exakte Kurve $(\alpha(t), \alpha'(t))$ sowie ihre Näherung als Polygonzug $\{(\alpha_0, \alpha'_0), (\alpha_1, \alpha'_1), (\alpha_2, \alpha'_2), \dots\}$.

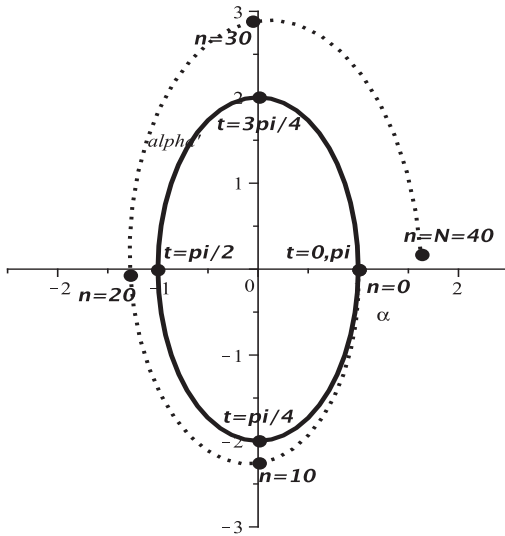


Abb. 1.11

Datei *ependel_07sw.eps*,
Fadenpendel mit
Phasenkurve (α, α')
und Näherung (gepunktet)
mittels PZV

7. Simulation eines Doppelpendels [45, 112]

Wir betrachten ein idealisiertes Doppelpendel.

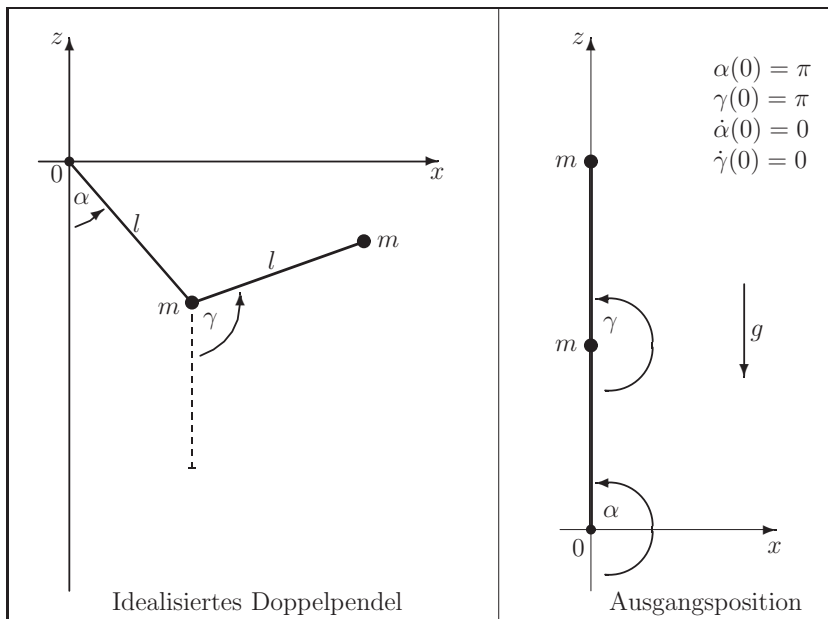


Abb. 1.12 Datei *bs22.pic*, Doppelpendel mit Parameter

Zwei Punktmassen gleicher Masse m , die an zwei masselosen Stangen der Länge l aufgehängt sind, bewegen sich unter dem Einfluss der Schwerkraft mg . Der Luftwiderstand, die Reibung der Lager und andere Störungen werden vernachlässigt. Der Winkel $\alpha(t)$ gibt die Lage des inneren Körpers 1 bzgl. der z -Achse an, $\gamma(t)$ die des äußeren Körpers 2 bzgl. der z -Achse an.

Ziel ist die Simulation und Beschreibung des Bewegungsablaufes mittels der Winkel $\alpha(t)$ und $\gamma(t)$ ohne bzw. mit Störgrößen.

Die Ausgangsposition sei gegeben durch $\alpha(0)$, $\gamma(0)$, $\dot{\alpha}(0)$, $\dot{\gamma}(0)$.

- (a) Charakterisierung der Geometrie, Rotation und Schwingung sowie gDGL
 (b) Geometrie, Lage des inneren und äußeren Körpers im \mathbb{R}^3

$$r_1 = l \begin{pmatrix} \sin(\alpha) \\ 0 \\ -\cos(\alpha) \end{pmatrix}, \quad r_2 = l \begin{pmatrix} \sin(\alpha) + \sin(\gamma) \\ 0 \\ -\cos(\alpha) - \cos(\gamma) \end{pmatrix}. \quad (1.59)$$

- (c) Energieerhaltungssatz

$$E = \dot{\alpha}^2 + \frac{1}{2}\dot{\gamma}^2 + \dot{\alpha}\dot{\gamma}\cos(\gamma - \alpha) - 2\cos(\alpha) - \cos(\gamma). \quad (1.60)$$

- (d) Drallsatz

$$\frac{dL_A}{dt} = \begin{pmatrix} 0 \\ 2\sin(\alpha) + \sin(\gamma) \\ 0 \end{pmatrix}. \quad (1.61)$$

- (e) SysgDGL 2. Ordnung

$$\begin{aligned} \ddot{\alpha} &= \frac{-2\sin(\alpha) + \dot{\gamma}^2 \sin(\gamma - \alpha) + \cos(\gamma - \alpha)[\sin(\gamma) + \dot{\alpha}^2 \sin(\gamma - \alpha)]}{2 - \cos^2(\gamma - \alpha)}, \\ \ddot{\gamma} &= \frac{-2\sin(\gamma) - 2\dot{\alpha}^2 \sin(\gamma - \alpha) + \cos(\gamma - \alpha)[2\sin(\alpha) - \dot{\gamma}^2 \sin(\gamma - \alpha)]}{2 - \cos^2(\gamma - \alpha)}. \end{aligned} \quad (1.62)$$

- (f) Man löse nun die gDGL 2. Ordnung mittels PZV, das für die gDGL $\ddot{y} = f(t, y, \dot{y})$ mit AB in der Gestalt

$$\begin{aligned} y_{n+1} &= y_n + \Delta t \dot{y}_n, & n = 0, 1, \dots, N, \quad t_n = n\Delta t, \quad y_0, \dot{y}_0 \text{ geg.}, \\ \dot{y}_{n+1} &= \dot{y}_n + \Delta t \ddot{y}_n = \dot{y}_n + \Delta t f(t, y_n, \dot{y}_n), \end{aligned} \quad (1.63)$$

Schrittweite $\Delta t > 0$ und Anzahl der Schritte N (Simulationsdauer).

Für obiges System arbeitet das PZV gemäß der Reihenfolge

– Berechnung von $\ddot{\alpha}$, $\ddot{\gamma}$ anhand der gDGL zum Zeitpunkt t_n

$$- \alpha_{n+1} = \alpha_n + \Delta t \dot{\alpha}_n,$$

$$\gamma_{n+1} = \gamma_n + \Delta t \dot{\gamma}_n,$$

$$- \dot{\alpha}_{n+1} = \dot{\alpha}_n + \Delta t \ddot{\alpha}_n,$$

$$\dot{\gamma}_{n+1} = \dot{\gamma}_n + \Delta t \ddot{\gamma}_n.$$

- (g) Tabellarische Ausgabe der Ergebnisse, evtl. auch als File

- (h) Darstellung der berechneten Bewegung in verschiedenen Plots
- Bewegungsablauf des Doppelpendels (der beiden Stangen) im (x, z) -Koordinatensystem (Grafik wie oben),
 - Darstellung der Winkelfunktionen $\alpha(t)$, $\gamma(t)$ entsprechend der Simulationsdauer. Dabei sollen Winkelwerte, die außerhalb von $[0, 2\pi]$ liegen, mittels Modulo-Rechnung auf dieses Intervall transformiert werden.

Die nachfolgende qualitative Abbildung entstand mit den AB $\alpha(0) = \pi$, $\dot{\alpha}(0) = 0$, $\gamma(0) = \pi$, $\dot{\gamma}(0) > 0$.

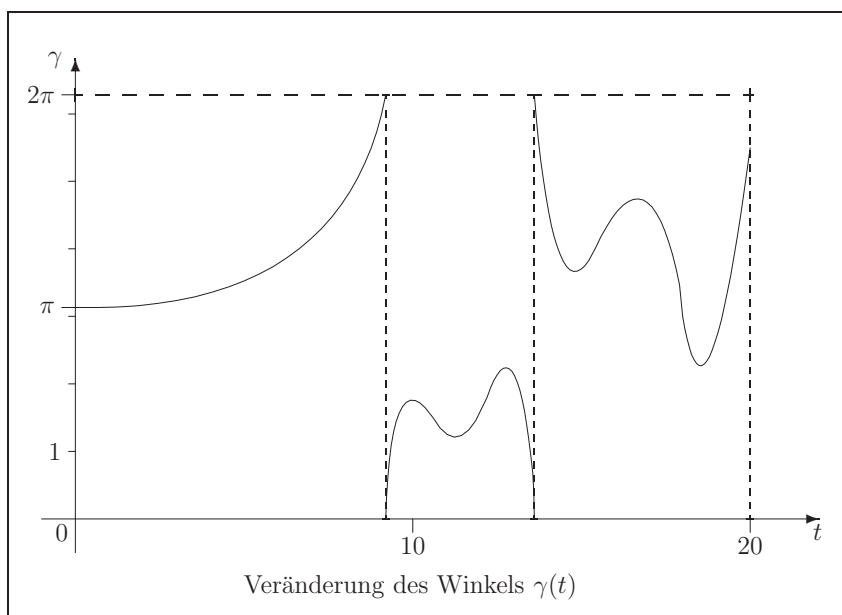


Abb. 1.13 Datei *bs23.pic*, Doppelpendel, Winkel $\gamma(t)$

- (i) Man kann in die Betrachtungen noch eine kleine Störung s einbeziehen (Masse auf der x -Achse, die eine geringe Beschleunigung verursacht). Größenordnungen für s sind $10^{-5} \dots 10^{-10}$. In den Beziehungen wirkt sich diese wie folgt aus.

Energieerhaltungssatz

$$E = \dot{\alpha}^2 + \frac{1}{2}\dot{\gamma}^2 + \dot{\alpha}\dot{\gamma}\cos(\gamma - \alpha) - 2\cos(\alpha) - \cos(\gamma) + s(2\sin(\alpha) - \sin(\gamma)).$$

Drallsatz

$$\frac{dL_A}{dt} = \begin{pmatrix} 0 \\ 2\sin(\alpha) + \sin(\gamma) \\ 0 \end{pmatrix} - s \begin{pmatrix} 0 \\ 2\cos(\alpha) + \cos(\gamma) \\ 0 \end{pmatrix}.$$

Gestörtes SysgDGI 2. Ordnung

$$\ddot{\alpha} = \frac{-2 \sin(\alpha) + \dot{\gamma}^2 \sin(\gamma - \alpha) + \cos(\gamma - \alpha)[\sin(\gamma) + \dot{\alpha}^2 \sin(\gamma - \alpha)]}{2 - \cos^2(\gamma - \alpha)}$$

$$+ s \frac{\cos(\gamma) \cos(\gamma - \alpha) - 2 \cos(\alpha)}{2 - \cos^2(\gamma - \alpha)},$$

$$\ddot{\gamma} = \frac{-2 \sin(\gamma) - 2\dot{\alpha}^2 \sin(\gamma - \alpha) + \cos(\gamma - \alpha)[2 \sin(\alpha) - \dot{\gamma}^2 \sin(\gamma - \alpha)]}{2 - \cos^2(\gamma - \alpha)}$$

$$+ s \frac{2 \cos(\gamma - \alpha)(\cos(\alpha) - 1)}{2 - \cos^2(\gamma - \alpha)}.$$

Analog kann man die gestörten Winkelfunktionen mit PZV berechnen, die Ergebnisse tabellarisch ausgeben sowie den Bewegungsablauf grafisch darstellen.

Im Programm gibt es mehrere Menüs mit Auswahl der Menüpunkte. Nach Eingabe der Problemparameter sowie Berechnung und Speicherung der Lösung kann z.B. mittels Menüpunkt F5 *Show* folgende grafische Lösungsdarstellung aufgerufen werden. Der gesamte Programmcode ist ca. 130kB groß.

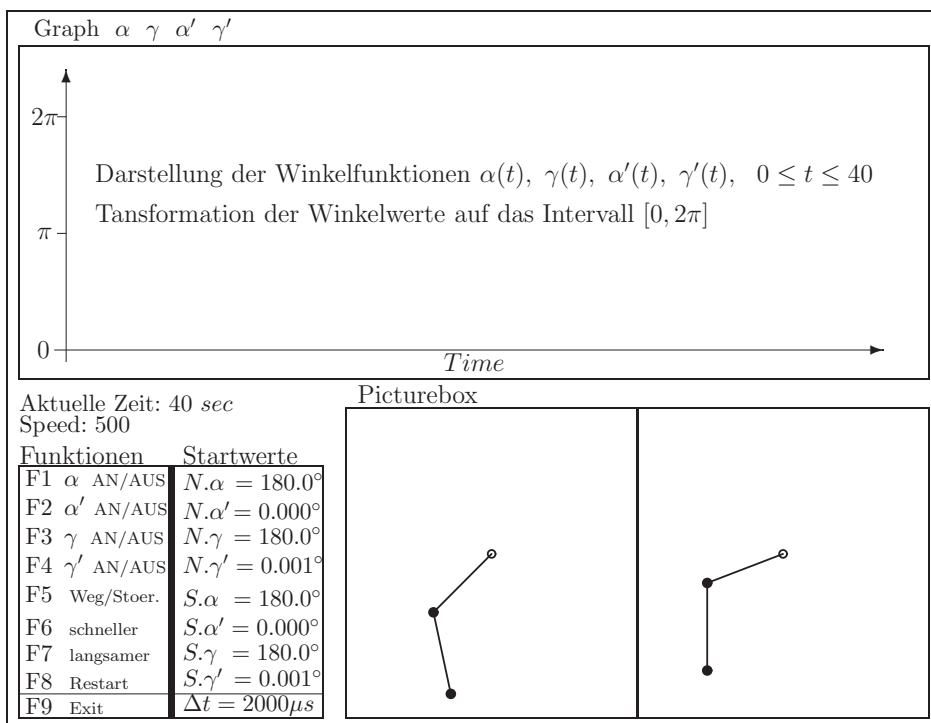


Abb. 1.14 Datei *bsi1.pic*, Menügestaltung zum Doppelpendel

Die Implementierung des Pendels ohne/mit Störung kann mittels einem modularen Konzept mit objektorientierten Komponenten in der Programmiersprache *Turbo Pascal* erfolgen. Die Programmoberfläche dazu ist als Anregung zu sehen. Die hardwaremäßige VESA-Unterstützung ist empfohlen, ansonsten ist das Programm UNIVBE.EXE zu laden.

Teile des Programmsystems sind:

- DOPPEL.PAS Hauptprogramm (Rahmen)
- VESA.PAS Routinen zur Programmierung des VESA-Standards

Include-Files

- TYPE.INC Globale Typdeklarationen
- CONST.INC Globale Konstanten
- VAR.INC Globale Variablen
- FORWARD.INC Forward-Deklarationen
- INIT.INC Grundinitialisierungen
- UTILS.INC Utilities
- GRAPH.INC Grafische Oberflächenroutinen
- AUSGABE.INC Anzeigeroutinen
- PROG.INC Tastenabfrage und Programmkern
- DIFF.INC Beschreibung der Differentialgleichungen
- CALC.INC gDGL-Löser (PZV)
- FILE.INC Filehandlingroutinen

Die numerische Lösung des AWP wird mittels PZV sowohl für den ungestörten als auch gestörten Fall $N.\alpha(t), \dots$ bzw $S.\alpha(t), \dots$ durchgeführt.

Die Darstellung der berechneten Bewegung erfolgt in verschiedenen Plots

- als Bewegungsablauf der beiden Stangen im (x, z) -Koordinatensystem,
- wahlweise Darstellung der Winkelfunktionen $\alpha(t), \alpha'(t), \gamma(t), \gamma'(t)$.

Die Geschwindigkeit des Ablaufs kann durch Steuerparameter geändert werden.

1.5 Mathematik

1. Bestimmung von Kurven mittels Abstandsbedingung I

Gesucht sind alle Kurven mit folgenden Eigenschaften.

Die Tangente an einen Kurvenpunkt P schneide die x -Achse im Punkt T .

Wie lautet die Gleichung der Kurve, wenn der Abstand zwischen P und T gleich dem Abstand zwischen T und dem Koordinatenursprung O für alle Punkte P der Kurve ist?

Aus der Abstandsbedingung $|OT|^2 = |TP|^2$ und der Tangentengleichung $y - y(x_0) = y'(x_0)(x - x_0)$ am Punkt $P(x_0, y_0)$ einer Kurve $y = y(x)$ folgt die gDGL

$$y' = \frac{2xy}{x^2 - y^2}. \quad (1.64)$$

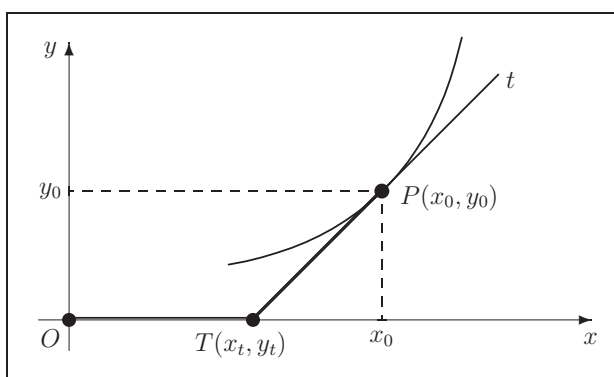


Abb. 1.15
Datei *bs19.pic*,
Kurvenbestimmung
mittels
Abstandsbedingung I

(a) Allgemeine Lösung der gDGL

Es ist eine homogene gDGL, die unter Nutzung der Substitution $u = \frac{y}{x}$ sowie mittels Partialbruchzerlegung gelöst werden kann. Mit $y' = u'x + u$ erhält man

$$y' = \frac{2\frac{y}{x}}{1 - (\frac{y}{x})^2},$$

$$u'x + u = \frac{2u}{1 - u^2},$$

$$u'x = \frac{u(1 + u^2)}{1 - u^2},$$

$$\frac{1}{x} = \frac{1 - u^2}{u(1 + u^2)} u' = \frac{u'}{u} - \frac{2uu'}{1 + u^2},$$

$$\ln(|u|) - \ln(|1 + u^2|) = \ln(|x|) + c,$$

$$\left| \frac{u}{1 + u^2} \right| = c_1|x|, \quad c_1 \in \mathbb{R}^+.$$

Mit $u = y/x$ erfolgt nach Umformung $y = c_1(y^2 + x^2)$ und

$$\left(y - \frac{c_2}{2}\right)^2 + x^2 = \left(\frac{c_2}{2}\right)^2, \quad c_2 = 1/c_1. \quad (1.65)$$

Das sind offenbar Kreisgleichungen. Die Mittelpunkte der Kreise liegen auf der y -Achse.

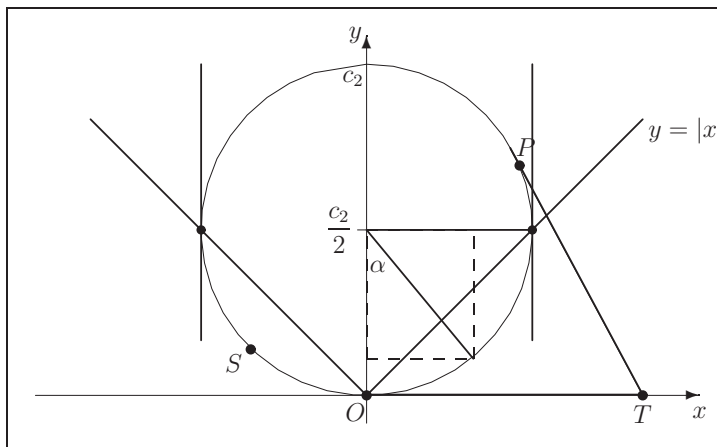


Abb. 1.16 Datei *bs20.pic*, Kurvenbestimmung mittels Abstandsbedingung und Kreisgleichung

- (b) Man löse nun das AWP $y' = \frac{2xy}{x^2 - y^2} = f(x, y)$, $y(x_0) = y_0$, mittels PZV
- $$\eta_{n+1} = \eta_n + hf(x_n, \eta_n), \quad n = 0, 1, \dots, \quad x_n = x_0 + nh, \quad \eta_0 = y_0, \quad h > 0. \quad (1.66)$$
- Jedoch sind zunächst einige Einschränkungen und Besonderheiten zu berücksichtigen.
- (c) Im kartesischen Koordinatensystem kann man mit der gDGL wegen der Eindeutigkeit der Lösungskurve maximal den unteren oder oberen Halbkreis beschreiben. Ansonsten ist die gDGL zu transformieren, indem die Koordinaten in Abhängigkeit vom Winkel α geschrieben werden, z.B. $x(\alpha) = \frac{c_2}{2} \sin(\alpha)$, $y(\alpha) = \frac{c_2}{2}(1 - \cos(\alpha))$, $\alpha \geq 0$.
- (d) Die AB kann nicht $(x_0, 0)$ sein. Dann ist $y'(x_0) = f(x_0, 0) = 0$ für $x_0 \neq 0$ oder man würde $y'(0) = f(0, 0)$ per Definition zu Null setzen. Das PZV $\eta_{n+1} = \eta_n(1 + h\frac{2x_n - \eta_n^2}{x_n^2 - \eta_n^2})$ liefert in beiden Fällen die bekannte Nulllösung der gDGL. Der Grenzfall ist der "unendliche" Kreis.
- (e) Als AB ist $y(x_0) = |x_0|$, $x_0 \neq 0$, nicht zugelassen, weil dann die Funktion $f(x, y)$ für den Startwert nicht definiert ist. Diese AB liegen dort, wo die Tangenten an den Kreis vertikal sind.
- (f) Man wählt also geeignete AB $S(x_0, y_0)$, berechnet mittels PZV die Näherungslösung und kontrolliert ihren Verlauf. Dazu macht man einen grafischen Vergleich der exakten Lösung (Kreisbogen) mit dem Lösungsverlauf (Polygonzug) des Näherungsverfahrens und testet verschiedene Schrittweiten h .

2. Bestimmung von Kurven mittels Abstandsbedingung II

Gesucht sind alle Kurven mit folgender Eigenschaft.

Der Schnittpunkt S der Kurvennormalen n in einem Kurvenpunkt P mit der x -Achse soll von der Abszisse von P stets den Abstand 2 haben.

Aus der Gleichung der Normalen $y - y(x_0) = -\frac{1}{y'(x_0)}(x - x_0)$ im Punkt $P(x_0, y_0)$ einer Kurve $y = y(x)$, ihrem Schnittpunkt $S(x_s, 0)$ mit der x -Achse sowie der Abstandsbedingung $x_s - x_0 = 2$ folgt die gDGl

$$y y' = 2. \quad (1.67)$$

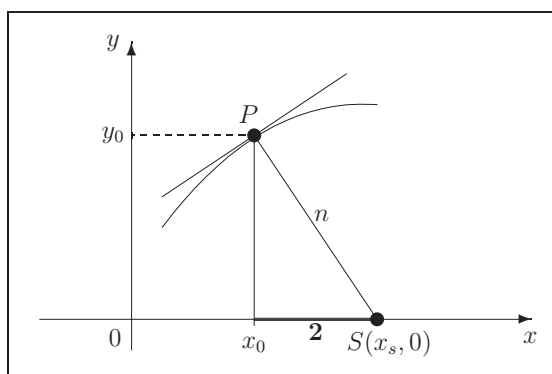


Abb. 1.17

Datei *bs21.pic*,

Kurvenbestimmung
mittels

Abstandsbedingung II

(a) Allgemeine Lösung der gDGl

Es gilt

$$\frac{1}{2}(y^2)' = 2,$$

$$y^2 = 4x + c,$$

$$y = 2\sqrt{x + c_1}, \quad x \geq -c_1. \quad (1.68)$$

(b) Man charakterisiere das Richtungsfeld der gDGl $y' = \frac{2}{y}$.

In jedem Punkt $P(x, y)$ des Gebietes $\Omega \in \mathbb{R}^2$ kann man die Steigung y' durch Eintragen eines kleinen "Strichs" für die Tangentenrichtung andeuten. Ein solcher Strich heißt Linienelement oder Richtungselement (x, y, y') der Lösungskurve. Die Gesamtheit aller Linienelemente heißt Richtungsfeld.

Was passiert, wenn $y = 0$ ist?

(c) Man bestimme die orthogonalen Trajektorien zur Lösungsschar.

Das sind die Kurven, die in jedem Punkt eine vorgegebene Kurvenschar orthogonal schneiden. Sie heißen orthogonale Trajektorien.

Gegeben sei die explizite gDGl 1. Ordnung $y' = f(x, y)$.

Für ihre Lösungsschar bestimmt man die orthogonale Kurvenschar, indem man lediglich y' durch $-\frac{1}{y'}$ ersetzt. Die Vektoren $(1, y')$ und $(1, -1/y')$ sind nämlich orthogonal.

Somit lautet zu (1.67) deren gDGl $\frac{y'}{y} = -\frac{1}{2}$. Ihre allgemeine Lösung ist $y = c_1 e^{-\frac{x}{2}}$.

Man zeichne ebenfalls das Richtungsfeld der orthogonalen Trajektorien.

- (d) Man löse nun die gDGL $y' = \frac{2}{y} = f(x, y)$ sowie die der orthogonalen Trajektorien $y' = -\frac{y}{2}$ mittels PZV (1.66). Man wählt geeignete AB $S(x_0, y_0)$, berechnet mittels PZV die Näherungslösung und kontrolliert ihren Verlauf. Dazu macht man einen grafischen Vergleich der exakten Lösung mit dem Lösungsverlauf des Näherungsverfahrens und testet verschiedene Schrittweiten h .

3. Phasenkurve

Für die Lösung $y = y(x)$ der gDGL $y' = f(x, y)$ ist manchmal auch eine andere Darstellung zu empfehlen.

Das trifft zum Beispiel auf Bewegungsdifferentialgleichungen bei Abhängigkeit von der Zeit t zu. Formal kann man $x = t$ setzen und erhält das SysgDGL

$$\dot{x} = 1, \quad \dot{y} = f(t, y). \quad (1.69)$$

Die Lösung $(x(t), y(t))$ wird Phasenkurve genannt und ihre Darstellung kann im Phasenraum (x, y) in Abhängigkeit von der Zeit t erfolgen.

Mit einer gegebenen AB $(y(t_0), x(t_0)) = (y_0, x_0)$ löst man das AWP.

Allgemein lautet das SysgDGL

$$\dot{x} = f_1(t, x, y), \quad \dot{y} = f_2(t, x, y). \quad (1.70)$$

Der eigentliche Typ von Phasenkurven entsteht bei der gDGL 2. Ordnung

$$y'' = f(t, y, y'), \quad (1.71)$$

wenn man diese durch die Definition $x = y'$ in das System von zwei Gleichungen

$$\begin{aligned} y' &= x = f_1(t, y, x), \\ x' &= y'' = f(t, y, x) = f_2(t, y, x) \end{aligned} \quad (1.72)$$

überführt und somit die Phasenkurve $(y, x) = (y, y')$ erhält.

Man konstruiere für folgende gDGL die Phasenkurven.

- $y' = 1 + y, y(0) = 1/2$.
- $y'' + y = 0, y(0) = 0, y'(0) = 1$, Pendel ohne Reibung.
- $y'' + ky' + y = 0, y(0) = y_0, y'(0) = y'_0$, Pendel mit Reibung $k > 0$,
z.B. $k = \frac{10}{3}$.
- $y'' - (2 + e^t)y' + y = 0, y(0) = 2, y'(0) = -1$.

Liefert der Punkt $(1, 0)$ als Grenzwert der Phasenkurve eine stationäre Lösung?

4. Van der Polsche DGL [7]

Die normierte Van der Polsche DGL 2. Ordnung lautet

$$\ddot{y} - \varepsilon(1 - y^2)\dot{y} + y = 0, \quad y(0) = y_0, \quad \dot{y}(0) = \dot{y}_0, \quad t \geq 0, \quad \varepsilon \geq 0. \quad (1.73)$$

5. Notation der Van der Polsche DGI als Phasenkurve

$$\ddot{y} = \varepsilon(1 - y^2)\dot{y} - y = f(t, y, \dot{y}), \quad x \equiv \dot{y}.$$

System von gDGI 1. Ordnung

$$\begin{aligned} \dot{y} &= x &= f_1(t, y, x), & y(t_0) = y_0, t_0 = 0, \\ \dot{x} &= f(t, y, x) = f_2(t, y, x), & x(t_0) = \dot{y}_0. \end{aligned} \quad (1.74)$$

Phasenkurve $(y, x) = (y, \dot{y})$.

6. Van der Polsche DGI und Lösung für $\varepsilon = 0$

$$\begin{aligned} \ddot{y} + y &= 0, \\ y(t) &= \dot{y}_0 \sin(t) + y_0 \cos(t) = A \sin(\omega t + \varphi), \quad \text{harmonische Schwingung,} \\ y(t) &= \sqrt{2} \sin\left(t + \frac{\pi}{4}\right), \quad \text{Lösung für die AB } y_0 = \dot{y}_0 = 1. \end{aligned}$$

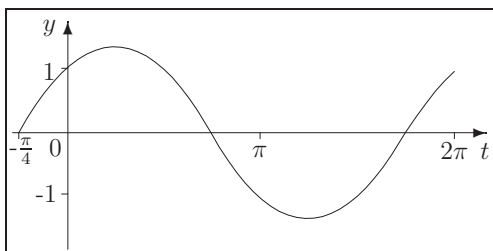


Abb. 1.18

Datei *bss4.pic*,

Lösungskurve mit
Amplitude $A = \sqrt{2}$,

Frequenz $\omega = 1$,

Periode $\frac{2\pi}{\omega} = 2\pi$

7. Lösung der gDGI $\ddot{y} + \omega^2 y = 0$, $\omega > 0$

$$\begin{aligned} y(t) &= A \sin(\omega t + \varphi), \quad \text{Ansatz zur Bestimmung von } A, \varphi, \\ \dot{y}(t) &= A\omega \cos(\omega t + \varphi), \\ \ddot{y}(t) &= -A\omega^2 \sin(\omega t + \varphi). \end{aligned}$$

Berücksichtigung der AB ohne Betrachtung von Sonderfällen

$$\begin{aligned} y_0 &= A \sin(\omega t_0 + \varphi), \\ \dot{y}_0 &= A\omega \cos(\omega t_0 + \varphi), \\ \tan(\omega t_0 + \varphi) &= \frac{\omega y_0}{\dot{y}_0}, \\ \varphi &= \arctan\left(\frac{\omega y_0}{\dot{y}_0}\right) - \omega t_0, \\ \omega^2 y_0^2 + \dot{y}_0^2 &= A^2 \omega^2 \sin^2(\omega t_0 + \varphi) + A^2 \omega^2 \cos^2(\omega t_0 + \varphi), \\ A^2 &= y_0^2 + \frac{\dot{y}_0^2}{\omega^2}, \quad A = \frac{1}{\omega} \sqrt{(\omega y_0)^2 + \dot{y}_0^2}. \end{aligned}$$

Die Lösung ist

$$y(t) = \frac{1}{\omega} \sqrt{(\omega y_0)^2 + \dot{y}_0^2} \sin\left(\omega(t - t_0) + \arctan\left(\frac{\omega y_0}{\dot{y}_0}\right)\right). \quad (1.75)$$

8. GDGl $\ddot{y} + \omega^2 y = 0$, $\omega = 1$, als SysgDGl

$$\begin{aligned}x &= \dot{y}, \\ \dot{x} &= \ddot{y} = -y.\end{aligned}$$

Somit gilt

$$\begin{aligned}\dot{y} &= x, & y(t_0) &= y_0, \\ \dot{x} &= -y, & x(t_0) &= \dot{y}_0,\end{aligned}$$

bzw.

$$\begin{aligned}\dot{x}_1 &= x_2, & x_1(t_0) &= x_{10} = y_0, \\ \dot{x}_2 &= -x_1, & x_2(t_0) &= x_{20} = \dot{y}_0.\end{aligned} \tag{1.76}$$

Phasenkurve $(y, x) = (y, \dot{y}) = (x_1, x_2)$.

Lösungsansatz mit Bestimmung des Koeffizienten A aus den AB ohne Betrachtung von Sonderfällen

$$\begin{aligned}x_1(t) &= A \cos(t + \varphi), \\ x_2(t) &= -A \sin(t + \varphi),\end{aligned}$$

$$x_1(t_0) = A \cos(t_0 + \varphi) = y_0,$$

$$x_2(t_0) = -A \sin(t_0 + \varphi) = \dot{y}_0,$$

$$\varphi = -\arctan\left(\frac{\dot{y}_0}{y_0}\right) - t_0 = -\operatorname{arccot}\left(\frac{y_0}{\dot{y}_0}\right) - t_0,$$

$$A = \sqrt{y_0^2 + \dot{y}_0^2},$$

$$\begin{aligned}y(t) = x_1(t) &= \sqrt{y_0^2 + \dot{y}_0^2} \cos\left(t - t_0 - \operatorname{arccot}\left(\frac{y_0}{\dot{y}_0}\right)\right) \\ &= \sqrt{y_0^2 + \dot{y}_0^2} \cos\left(t - t_0 - \left(\frac{\pi}{2} - \arctan\left(\frac{y_0}{\dot{y}_0}\right)\right)\right) \\ &= \sqrt{y_0^2 + \dot{y}_0^2} \sin\left(t - t_0 + \arctan\left(\frac{y_0}{\dot{y}_0}\right)\right).\end{aligned}$$

Spezialfälle

1. $y_0 = A$, $\dot{y}_0 = 0 \rightarrow \arctan(\infty) = \frac{\pi}{2}$.

$$y(t) = A \sin\left(t - t_0 + \frac{\pi}{2}\right) = A \cos(t - t_0).$$

2. $y_0 = 1$, $\dot{y}_0 = 2 \rightarrow A = \sqrt{5}$, $\varphi = -\arctan(2) - t_0 \approx -1.1 - t_0$.

$$y(t) = \sqrt{5} \sin(t - t_0 + \arctan(0.5)) \approx \sqrt{5} \sin(t - t_0 + 0.46),$$

$$y(t_0) = \sqrt{5} \sin(0.46) \approx 1,$$

$$\dot{y}(t) = \sqrt{5} \cos(t - t_0 + \arctan(0.5)) \approx \sqrt{5} \cos(t - t_0 + 0.46),$$

$$\dot{y}(t_0) = \sqrt{5} \cos(0.46) \approx 2.$$

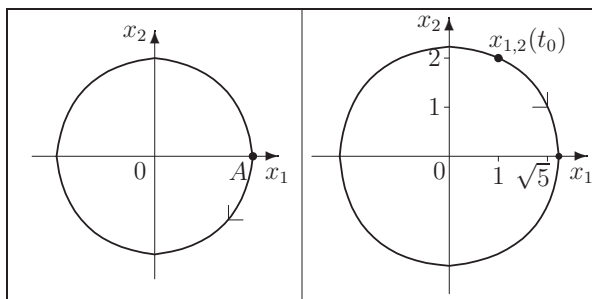


Abb. 1.19 Datei *bss5.pic*, Phasenkurven $(x_1(t), x_2(t))$ zu

$$\begin{aligned} x_1(t) &= A \cos(t - t_0), & x_1(t) &= \sqrt{5} \cos(t - t_0 - 1.1), \\ x_2(t) &= -A \sin(t - t_0), & x_2(t) &= -\sqrt{5} \sin(t - t_0 - 1.1) \end{aligned}$$

9. Van der Polsche DGL

$$\ddot{y} = \varepsilon(1 - y^2)\dot{y} - y = f(t, y, \dot{y}), \quad x_1 = \dot{y}, \quad \varepsilon \geq 0, \quad (1.77)$$

als SysgDGL 1. Ordnung

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -x_1 + \varepsilon(1 - x_1^2)x_2. \end{aligned} \quad (1.78)$$

Damit erhalten wir eine besondere Form der allgemeinen Störungsgleichung $\dot{x}_1 = x_2 + \varepsilon_1 f_1(x_1, x_2)$, $\dot{x}_2 = -x_1 + \varepsilon_2 f_2(x_1, x_2)$ mit entsprechenden Funktionen $f_{1,2}$.

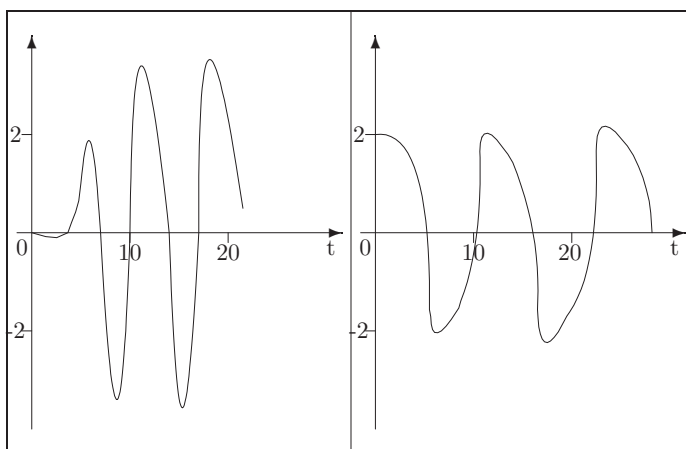


Abb. 1.20 Datei *bss6.pic*, Schwingungskurven $y(t)$ (ungefähre Verläufe) zu

$$\begin{aligned} \ddot{y} - (1 - y^2)\dot{y} + y &= 0, & \ddot{y} - 5(1 - y^2)\dot{y} + y &= 0 \\ y(0) &= 0, \quad \dot{y}(0) = -0.05, & y(0) &= 2, \quad \dot{y}(0) = 0 \end{aligned}$$

Wächst $\varepsilon > 0$ an, so verzerrt sich die Schwingung immer mehr und die Periode vergrößert sich. Bei $\varepsilon \gg 1$ (z.B. $\varepsilon = 100, 1000$) beschreibt die gDGL Relaxationschwingungen. Dabei wechseln sich Zeiten sehr langsamer Bewegung mit solchen sehr schneller Bewegung ab.

1.6 Biologie und Populationsdynamik

1. Zweispezimodell – ökologisches Gleichgewicht bei Populationen

Dies wird beschrieben durch das Volterrasche SysgDGl 1. Ordnung

$$\begin{aligned}\dot{x} &= ax(1-y), \quad x(0) = 1, \quad a = 2, \quad x = x(t), \\ \dot{y} &= -cy(1-x), \quad y(0) = 3, \quad c = 1, \quad y = y(t),\end{aligned}\tag{1.79}$$

bzw. entkoppelt durch die Form

$$\begin{aligned}x\ddot{x} &= \dot{x}^2 + acx^2 - cx\dot{x} + cx^2\dot{x} - acx^3, \\ y\ddot{y} &= \dot{y}^2 + acy^2 + ay\dot{y} - ay^2\dot{y} - acy^3.\end{aligned}\tag{1.80}$$

$$\text{AB } x(0) = 1, \quad y(0) = 0,$$

$$\begin{aligned}\text{Lösung: } x &= e^{2t}, \\ y &= 0.\end{aligned}$$

$$\text{AB } x(0) = 0, \quad y(0) = 1,$$

$$\begin{aligned}\text{Lösung: } x &= 0, \\ y &= e^{-t}.\end{aligned}$$

2. Lotka-Volterra-Modell [13]

Das originale Lotka-Volterra-Modell

$$\begin{aligned}\dot{x} &= x(1-y), \\ \dot{y} &= -y(1-x)\end{aligned}\tag{1.81}$$

ist ein Räuber-Beute-System mit der Beutepopulation/Futtermvorrat $x(t)$ und der Räuberpopulation/Verbraucher $y(t)$ und kann auf eine höhere Dimension verallgemeinert werden.

Die Verallgemeinerung mit drei Funktionen hat z.B. die Form

$$\begin{aligned}\dot{x}_1 &= x_1 - x_1x_2 + cx_1^2 - ax_3x_1^2 = x_1(1-x_2) + x_1^2(c-ax_3), \\ \dot{x}_2 &= -x_2 + x_1x_2 = -x_2(1-x_1), \\ \dot{x}_3 &= -bx_3 + ax_1x_3^2 = -x_3(b-ax_1x_3)\end{aligned}\tag{1.82}$$

mit $a, b, c \geq 0$ als sogenannte Bifurkationsparameter.

3. Modelle der Konkurrenz zweier Arten, die sich aus derselben beschränkten Resource ernähren [13]

$$\begin{aligned}\text{a) } \dot{u} &= u[-a_1 + c_1(1-b_1u-b_2v)], \quad u(0) = u_0, \quad c_1 > a_1, \\ \dot{v} &= v[-a_2 + c_2(1-b_1u-b_2v)], \quad v(0) = v_0, \quad c_2 > a_2.\end{aligned}\tag{1.83}$$

$$\begin{aligned}\text{b) } \dot{u} &= u[b_1 - \lambda_1(h_1u + h_2v)], \quad u(0) = u_0, \\ \dot{v} &= v[b_2 - \lambda_2(h_1u + h_2v)], \quad v(0) = v_0.\end{aligned}\tag{1.84}$$

4. Jäger-Beute-Modelle [13, 63]

$$\begin{aligned} \text{a)} \quad \dot{u} &= u \left[R \left(1 - \frac{u}{K} \right) - \frac{Rv}{u+D} \right], & u(0) &= u_0, \\ \dot{v} &= v \left[-S \left(1 - \frac{\gamma v}{u} \right) \right], & v(0) &= v_0. \end{aligned} \quad (1.85)$$

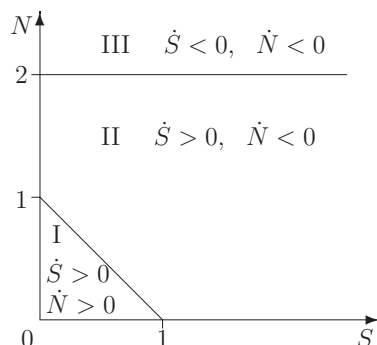
$$\begin{aligned} \text{b)} \quad \dot{x} &= ax - bxy - ex^2, & x(0) &= x_0, \\ \dot{y} &= -cy + dxy - fy^2, & y(0) &= y_0. \end{aligned} \quad (1.86)$$

$$\begin{aligned} \text{c)} \quad \dot{x} &= ax - b\sqrt{x}y, & x(0) &= x_0 > 0, \\ \dot{y} &= \begin{cases} d\sqrt{x}y, & \text{falls } x \neq 0, \\ -cy, & \text{falls } x = 0, \end{cases} & y(0) &= y_0 > 0. \end{aligned} \quad (1.87)$$

5. Ameisen-Kampf [13]

$$\begin{aligned} \dot{S} &= \frac{1}{10}S - \frac{1}{20}SN, & S(t_0) &= S_0 > 0, \\ \dot{N} &= \frac{1}{100}N - \frac{1}{100}N^2 - \frac{1}{100}SN, & N(t_0) &= N_0 > 0. \end{aligned} \quad (1.88)$$

- (a) Man zeige, dass die Geraden $N = 2$ und $N + S = 1$ den ersten Quadranten in drei Gebiete aufteilen, in denen $\frac{dS}{dt}$ und $\frac{dN}{dt}$ feste Vorzeichen haben.

**Abb. 1.21**Datei *bsi2.pic*,

Modell Ameisen-Kampf,

Gebietsaufteilung

bzgl. Vorzeichen von \dot{S} , \dot{N}

- (b) Man verifiziere, dass jede Lösung $S(t), N(t)$, die in den Gebieten I oder III startet, schließlich nach II geht.
- (c) Man beweise, dass jede im Gebiet II beginnende Lösung $S(t), N(t)$ für alle späteren Zeitpunkte dort bleibt.
- (d) Man schließe aus (c), dass für alle Lösungen $S(t), N(t)$ mit positiven $S(t_0), N(t_0)$ $\lim_{t \rightarrow \infty} S(t) = \infty$, gilt, sowie $\lim_{t \rightarrow \infty} N(t)$ endlich mit Grenzwert ≤ 2 ist.
- (e) Man beachte, dass ein t_0 existiert, sodass $\frac{dN}{dt} \leq -N$ für $t \geq t_0$ gilt. Man schließe aus dieser Ungleichung auf den Grenzwert $\lim_{t \rightarrow \infty} N(t) = 0$.

6. Veränderungen bei Populationen – Wachstumsmodell

Unter der Annahme, dass es eine maximale Population m gibt, die z.B. aus Gründen des Lebensraumes, nicht überschritten werden kann, sei die Bestandsänderung der Population proportional sowohl zum Bestand als auch zur jeweiligen Abweichung vom möglichen Höchststand. Dies ergibt die gDGl

$$y' = f(t, y) = p y (m - y), \quad y = y(t) \geq 0, \quad \text{AB } y(0) = y_0, \quad (1.89)$$

wobei $p > 0$ ein Proportionalitätsfaktor ist.

Sie wird auch als Wachstumsgleichung oder logistische Gleichung bezeichnet.

(a) Lösung des AWP

Hinweis: Trennung der Variablen, Idee der Partialbruchzerlegung.

$$\begin{aligned} p &= \frac{1}{y(m-y)} y' = \left[\frac{1}{my} + \frac{1}{m(m-y)} \right] y', \\ mp &= \frac{y'}{y} + \frac{y'}{m-y}, \\ \int m p dt &= \int \frac{y'}{y} dt + \int \frac{y'}{m-y} dt, \\ m p t + c &= \ln(y) - \ln(m-y) = \ln\left(\frac{y}{m-y}\right), \\ c_1 e^{m p t} &= \frac{y}{m-y}, \quad c_1 = e^c > 0. \end{aligned}$$

Somit haben wir die Bedingung für das Wachstum, die nach y aufzulösen ist.

$$y(t) = \frac{c_1 m e^{m p t}}{1 + c_1 e^{m p t}}, \quad c_1 \in \mathbb{R}^+. \quad (1.90)$$

Unter Berücksichtigung der AB ergibt sich die Beziehung

$$y(0) = \frac{c_1 m}{1 + c_1} \quad \text{und somit} \quad c_1 = \frac{y(0)}{m - y(0)}.$$

Die Wachstumsgleichung hat die endgültige Form

$$y(t) = \frac{y(0) m e^{m p t}}{m - y(0) + y(0) e^{m p t}}.$$

(b) Eigenschaften der gDGl und Lösung

$$y(t) = \frac{y(0) m e^{m p t}}{m + y(0) (e^{m p t} - 1)}, \quad t \in [0, T], \quad \text{AB } y(0) \in (0, m), \quad p > 0. \quad (1.91)$$

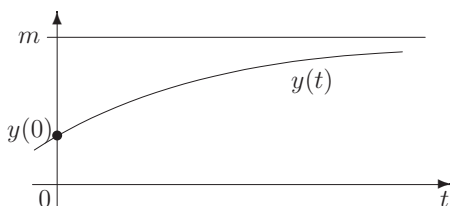


Abb. 1.22

Datei *bs16.pic*,

Wachstumsgleichung

(c) Man überprüfe folgende Aussagen zur Lösung.

$$\begin{aligned} y(0) &\leq y(t) < m, \quad m = \lim_{t \rightarrow \infty} y(t), \\ y(t_1) &< y(t_2) \quad \text{für } t_1 < t_2, \\ y'(t) &> 0, \quad y'(0) = p[m - y(0)]y(0). \end{aligned}$$

(d) Was passiert, wenn die AB $y(0) = 0$ bzw. $y(0) = m$ ist?

(e) Man charakterisiere das Richtungsfeld der gDGL.

In jedem Punkt $P(t, y)$ des Streifens $\Omega = [0, T] \times [0, m] \in \mathbb{R}^2$ kann man die Steigung y' durch Eintragen eines kleinen "Strichs" (Richtungselement (t, y, y')) für die Tangentenrichtung andeuten. Man nehme zwecks Vereinfachung für das Richtungsfeld die Parameter $p = m = 1$ und betrachte auch das Richtungsfeld außerhalb des Streifens.

(f) Man löse das AWP im Bereich mit zulässigem Startpunkt $y(0)$ mittels des PZV (1.66). Dazu soll ein grafischer Vergleich der exakten Lösung mit dem Lösungsverlauf des Näherungsverfahrens (Polygonzug) gemacht sowie verschiedene Schrittweiten $h > 0$ getestet werden.

Wie ist h zu wählen, damit z.B. analog zu den Eigenschaften der exakten Lösung die Bedingungen

$$\eta_n < \eta_{n+1} < \dots < m$$

erfüllt werden.

Man rechne das PZV auch mit AB, die außerhalb des Streifens liegen.

7. Modellierung eines Ökosystems

Das typische Räuber-Beute-Modell kann auf der Grundlage von gDGL beschrieben werden. Dazu nimmt man an, dass

- die räumliche Verteilung der Spezies keine Rolle spielt,
- die Modellierung mit reellen Zahlen aufgrund einer hohen Anzahl von Individuen sinnvoll ist,
- eine geeignete Normierung der Anzahlen vorgenommen wird.

Bezeichnet man die Menge Gras, die Zahl der Hasen und die Zahl der Füchse mit g, h bzw. f , so kann man folgendes Modell als SysgDGL aufstellen.

$$\begin{aligned} g'(t) &= 1 - h(t)g(t), \\ h'(t) &= h(t)[g(t) - f(t)] - c_3h(t), \\ f'(t) &= f(t)h(t) - c_1f(t) - c_2\sqrt{f(t)}, \quad c_1 + c_2 = c = \text{const} > 0, \quad c_i \geq 0. \end{aligned} \tag{1.92}$$

Mögliche AB sind $g(0) = h(0) = 1, f(0) = 0.1$.

Die erste Gleichung beschreibt das Wachstum der Grasmenge und das "Gefressenwerden von Gras" mit einer Rate, die proportional zum Produkt aus Hasen und Gras ist. In der zweiten Gleichung für Hasen ist die Wachstumsrate ebenfalls mit $h(t)g(t)$ gegeben. Zusätzlich geht die Zahl der Füchse "hasenmindernd" mit $-h(t)f(t)$ sowie die Hasenjagd mit $-c_3h(t)$ ein. In der Gleichung der Füchse ist ihr Wachstum von den vorhandenen Hasen abhängig, während der negative Term

das Jägerverhalten simulieren soll.

Der Abschuss erfolgt proportional zur Anzahl der Füchse bzw. Hasen.

(a) Simulationsläufe

Das SysgDGl tendiert zu Schwingungen. Diese werden jedoch mit der Zeit herausgedämpft, und es stellt sich ein Gleichgewicht ein.

Die Abbildungen sind mit *Mathematica* erstellt worden.

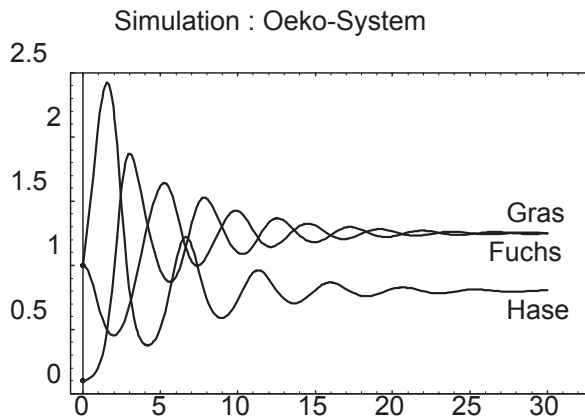


Abb. 1.23

Datei *oeko14.eps*,

Ökosystem

Gras, Hase, Fuchs,

$c = c_1 = 0.8$,

$c_2 = 0, c_3 = 0$,

Gleichgewichtslage
 $g(\infty) = f(\infty) = \frac{5}{4}$,

t $h(\infty) = \frac{4}{5}$

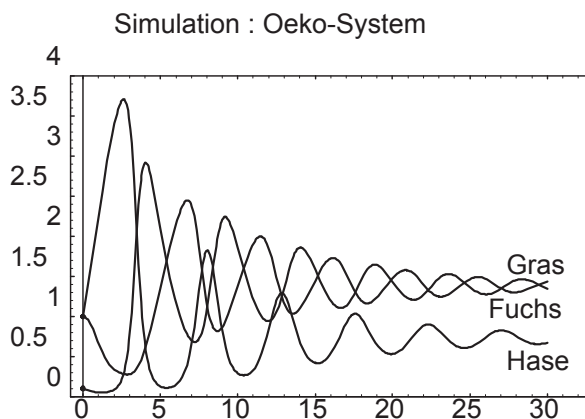


Abb. 1.24

Datei *oeko24.eps*,

Ökosystem

Gras, Hase, Fuchs,

$c = 0.8, c_1 = 0.3$,

$c_2 = 0.5, c_3 = 0$,

Gleichgewichtslage

$g(\infty) = \frac{85-5\sqrt{145}}{18} = 1.377$,

$f(\infty) = g(\infty)$,

t $h(\infty) = \frac{17+5\sqrt{145}}{40} = 0.726$

Im letzten Fall haben wir wegen $1 > \sqrt{f(t)} > f(t)$ am Anfang eine höhere Abschussquote der Füchse als im ersten Beispiel. Die Zahl der Füchse wird so stark dezimiert, sodass es zu einer richtigen Hasenplage kommt. Die Folge dazu ist wieder ein starker Rückgang der Grasmenge. Insgesamt sind die Ausschläge größer.

(b) Stationäres Gleichgewicht, Gleichgewichtslage

Bei hinreichend großer Zeit stellt sich für die Anzahlen der Populationen ein Gleichgewicht ein. Man berechnet die Gleichgewichtsgrößen $g_\infty = g(t = \infty)$, h_∞ und f_∞ aus den Bedingungen des stationären Verhaltens $g' = h' = f' = 0$. Dabei kann man auch den Typ der Gleichgewichtslage untersuchen.

(c) Numerik

Die numerische Lösung des SysgDgl 1. Ordnung

$$y' = F(t, y), \quad y: \mathbb{R} \rightarrow \mathbb{R}^3,$$

kann im einfachsten Fall mittels des PZV (1.66) bei geeigneten AB erfolgen. Man testet verschiedene Schrittweiten, kontrolliert den Verlauf der Näherungslösungen und erzeugt grafische Darstellungen.

1.7 Medizin

1. Modell der Diabeteserkennung [13]

$$\begin{aligned} \dot{u} &= F_1(u, v) + J(t), & u(t) &\text{Konzentration der Glukose im Blut,} \\ \dot{v} &= F_2(u, v), & v(t) &\text{Insulinkonzentration,} \\ & & J(t) &\text{externe Rate des Anstiegs der Blutzuckerkonzentration.} \end{aligned} \quad (1.93)$$

2. Epidemiologie [13]

Es wird ein einfaches Modell zur zeitlichen Verbreitung von Epidemien ohne Betrachtung von Migrationen der Gesamtpopulation vorgestellt.

Fragen sind z.B.: Entsteht eine Epidemie oder nicht? Wann stirbt sie aus?

$$\begin{aligned} \dot{y}_1 &= -ry_1y_2, & y_1 &\text{aufnahmefähige Population,} \\ \dot{y}_2 &= ry_1y_2 - \gamma y_2, & y_2 &\text{infizierte Population,} \\ \dot{y}_3 &= \gamma y_2, & y_3 &\text{resistente bzw. gestorbene Population.} \end{aligned} \quad (1.94)$$

3. Modell eines Immunsystems [63]

Das Modell der Abwehr des Immunsystems gegen Krebs beschreibt die zeitliche und räumliche Entwicklung der Zellular-Population $X(t, r)$, $M_0(t, r)$, $M_1(t, r)$, wobei $X(t, r)$ die Konzentration der normalen Zellen sowie $M_{0,1}(t, r)$ die Konzentrationen der zytotoxischen T-Zellen (T-Killerzellen, Zellen des Immunsystems zur Abwehr von Krankheitserregern) bedeuten.

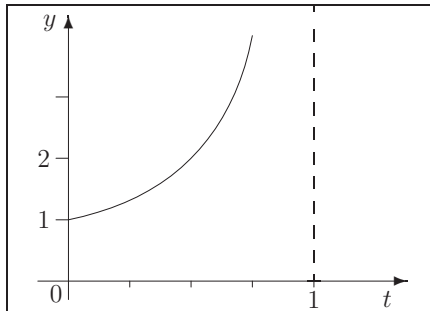
4. Zellwachstum

Die gDGL für explosionsartiges Wachstum von Zellen hat die Form

$$y' = y^2, \quad y(0) = c > 0, \quad t \geq 0. \quad (1.95)$$

Beschreibung der Lösung

$$\begin{aligned} \frac{y'}{y^2} &= -\left(\frac{1}{y}\right)' = 1, \quad y \neq 0, \\ \frac{1}{y} &= -t + C, \quad \text{AB beachten,} \\ y(t) &= \frac{1}{-t+1/c}. \end{aligned}$$

**Abb. 1.25**Datei *bss9.pic*,

Wachstumsfunktion

$$y(t) = \frac{1}{-t+1}, \quad c = 1,$$

Polstelle bei $t = 1$

5. Tumorwachstum [13]

Die einfache Dynamik des Zellwachstums kann beschrieben werden durch die gDGL

$$y' = \lambda y, \quad y(t_0) = y_0, \quad \lambda > 0 \text{ Wachstumsrate.} \quad (1.96)$$

Gemäß ihrer Lösung

$$y(t) = y_0 e^{\lambda(t-t_0)} \quad (1.97)$$

wachsen die sich teilenden Zellen exponentiell an, wobei sich das Zellvolumen $y(t)$ während eines jeden Zeitintervalls der Länge $\ln(2)/\lambda$ verdoppelt.

Empirische Untersuchungen haben gezeigt, dass die Zunahme des Zellvolumens eher der Bedingung

$$y(t) = y_0 e^{\lambda/\alpha(1-e^{-\alpha t})}, \quad \lambda, \alpha > 0, \quad (1.98)$$

entspricht (Gompertzsche Beziehung, Gompertzsche Wachstumskurve), die für $t \rightarrow \infty$ den Grenzwert $y_0 e^{\lambda/\alpha}$ hat.

Die dazugehörige gDGL ist

$$y' = \lambda e^{-\alpha t} y, \quad (1.99)$$

die auf zweierlei Art interpretiert werden kann.

a) $y' = (\lambda e^{-\alpha t}) y$

Das Abbremsen des Zellwachstums hat ihre Ursache in der mit der Zeit fallenden Wachstumsrate $\lambda e^{-\alpha t}$.

b) $y' = \lambda (e^{-\alpha t} y)$

Die durchschnittlich zur Zellbildung nötige Zeit bleibt konstant, aber die Wachstumshemmung basiert auf dem Verlust sich reproduzierender Zellen $e^{-\alpha t} y$.

1.8 Weitere Beispiele

Wir notieren nun weitere gDGI bzw. SysgDGI bzw. entsprechende AWP. Dabei können sich einige Beispiele der bisherigen Abschnitte durchaus wiederholen. In vielen Fällen werden die exakten oder bzw. allgemeinen Lösungen angegeben.

1.8.1 Skalare gDGI 1. Ordnung

$$y' = f(x, y) \text{ bzw. } f(x, y, y') = 0, \quad y = y(x), \quad \text{AB } y(x_0) = y_0$$

Notation auch mit $x = t, \dot{y} = y'$

1. $y' = \lambda y$, übliche Test-DGI, $\lambda \in \mathbb{C}, \mathbb{C}^-, \mathbb{R}$

Lösung: $y = ce^{\lambda x}$

$$y' = \lambda y, \quad y(0) = 1$$

Lösung: $y = e^{\lambda x}$

2. $y' = 1 + y$

Lösung: $y = -1 + ce^x$

$$y' = 1 \pm y, \quad y(0) = 1$$

Lösung: $y = \mp 1 + \left\{ \begin{array}{c} 2 \\ 1 \end{array} \right\} e^x$

3. $-\frac{1}{y'} = 1 + y$, orthogonale Trajektorien zu $y' = 1 + y$

Lösung: $y = -1 \pm \sqrt{c - 2x}$

$$-\frac{1}{y'} = 1 + y, \quad y(0) = 1$$

Lösung: $y = -1 + \sqrt{4 - 2x}$

4. $y' = \frac{1}{x + y}$

Lösung: $ce^y - (x + y + 1) = 0, \quad y - \ln(|x + y + 1|) = c,$
 $y = -x - 1 - \text{LambertW}(-ce^{-x-1})$

$$y' = \frac{1}{x + y}, \quad y(0) = 1$$

Lösung: $2e^{y-1} - (x + y + 1) = 0$

5. $y' = \frac{2}{y}$, Entfernung von Schnittpunkt T der Normalen an einen Kurvenpunkt $P(x, y)$ mit der x -Achse von der Abszisse von $P = 2$

Lösung: $y^2 = 4x + c$

$$y' = \frac{2}{y}, \quad y(0) = 1$$

Lösung: $y = \sqrt{4x + 1}$

6. $y' = y(1 - y)$, spezielle logistische DGL

Lösung: $y = \frac{ce^x}{1 + ce^x} = \frac{e^x}{d + e^x} = \frac{1}{1 + de^{-x}}$

$y' = y(1 - y)$, $y(0) = y_0$

Lösung: $y = \frac{y_0}{y_0 + (1 - y_0)e^{-x}}$

$y' = y(1 - y)$, $y(0) = 0$

Stationäre Lösung: $y = 0$

$y' = y(1 - y)$, $y(0) = 1$

Stationäre Lösung: $y = 1$

7. $y' = py(m - y)$, $p, m > 0$, $y(x) \leq m$, allgemeine logistische DGL

Lösung: $y = \frac{me^{mpx}}{mc + e^{mpx}}$

$y' = py(m - y)$, $y(0) = y_0$, $p, m > 0$, $y(x) \leq m$

Lösung: $y = \frac{y_0me^{mpx}}{m + y_0(e^{mpx} - 1)}$

$y' = py(m - y)$, $y(0) = 1$

Lösung: $y = \frac{me^{mpx}}{e^{mpx} + m - 1} = \frac{m}{1 + (m - 1)e^{-mpx}}$

8. $y' = \frac{2xy}{x^2 - y^2}$, Kurve mit Tangentenabschnitt = x -Achsenabschnitt $|TP| = |OP|$

Lösung: $\left(y - \frac{c}{2}\right)^2 + x^2 = \left(\frac{c}{2}\right)^2$ und einige Sonderfälle

$y' = \frac{2xy}{x^2 - y^2}$, $y(1) = 1$ bzw. $y(0) = 2$

Lösung: $(y - 1)^2 + x^2 = 1$

9. $y' = x^2 + y^2$

Lösung: $y = -x \frac{c \operatorname{BesselJ}(-3/4, x^2/2) + \operatorname{BesselY}(-3/4, x^2/2)}{c \operatorname{BesselJ}(1/4, x^2/2) + \operatorname{BesselY}(1/4, x^2/2)}$

$y' = x^2 + y^2$, $y(0) = 0$

Lösung: $y(0) = 0$, sonst $y = -x \frac{-\operatorname{BesselJ}(-3/4, x^2/2) + \operatorname{BesselY}(-3/4, x^2/2)}{-\operatorname{BesselJ}(1/4, x^2/2) + \operatorname{BesselY}(1/4, x^2/2)}$

10. $y' = y(2 - y)$

Lösung: $y = \frac{2ce^{2x}}{2 + ce^{2x}} = \frac{2e^{2x}}{2d + e^{2x}} = \frac{2}{1 + 2de^{-2x}}$

$y' = y(2 - y)$, $y(0) = 3$

Lösung: $y = \frac{6}{3 - e^{-2x}}$

11. $y' = \frac{y}{x}$
 Lösung: $y = cx$
 $y' = \frac{y}{x}$, $y(1) = 1$
 Lösung: $y = x$
12. $-\frac{1}{y'} = \frac{y}{x}$, orthogonale Trajektorien zu $y' = \frac{y}{x}$, Kreisschar
 Lösung: $y^2 + x^2 = c$
 $-\frac{1}{y'} = \frac{y}{x}$, $y(0) = 1$
 Lösung: $y^2 + x^2 = 1$
13. $\frac{1}{y'} = \frac{y}{x}$, Hyperbelschar
 Lösung: $y^2 - x^2 = c$
 $\frac{1}{y'} = \frac{y}{x}$, $y(0) = 1$
 Lösung: $y^2 - x^2 = 1$
14. $y' = \frac{-y+x}{y+x}$
 Lösung: $y^2 + 2yx - x^2 = c$
 $y' = \frac{-y+x}{y+x}$, $y(1) = 1$
 Lösung: $y^2 + 2yx - x^2 = 2$
15. $\frac{2y'}{x^2 + y^2} - 5y^3 = 0$
16. $\frac{x}{x-y'} + x + 2y^2 = x^2y$
17. $4(y')^2 - e^{y'} = x$
18. $(y')^3 - 2(y')^2 + y' = xy^2$
19. $y' = -(x+2)y^3$
 Lösung: $y = \pm \frac{1}{\sqrt{(x+2)^2 + c}}$, $\frac{1}{y^2} - x^2 - 4x - c = 0$, $y = 0$
 $y' = -(x+2)y^3$, $y(0) = 1$
 Lösung: $\frac{1}{y^2} - x^2 - 4x - 1 = 0$
20. $y' = 2 \left(\frac{y+2}{x+y-1} \right)^2$

$$21. \quad y' = \frac{3y - 7x + 1}{3x - 7y + 11}$$

$$\text{Lösung: } (y - 2)^2 - (x - 1)^2 = c \left(\frac{y - x - 1}{y + x - 3} \right)^{3/7}, \quad c \in \mathbb{R} \setminus \{0\}$$

$$22. \quad y' + p(x)y + q(x)y^n = 0, \quad \text{Bernoullische DGI, } n = 0, 1 \text{ einfache Sonderfälle}$$

Substitution $u = y^{1-n}$ liefert

$$(1 - n)y^{-n}y' + p(x)(1 - n)y^{1-n} + q(x)(1 - n) = 0$$

$$u' + p(x)(1 - n)u + q(x)(1 - n) = 0, \quad \text{lineare gDGI}$$

$$23. \quad xy' - 4y = x^2\sqrt{y}$$

$$\text{Lösung: } y = 0, \quad \sqrt{y} - x^2/2(\ln(x) + c) = 0$$

$$xy' - 4y = x^2\sqrt{y}, \quad y(1) = 1$$

$$\text{Lösung: } y = x^4(\ln(x)^2/4 + \ln(x) + 1)$$

$$24. \quad xy' + y = xy^2 \ln(x)$$

$$\text{Lösung: } y = \frac{2}{x(c - \ln(x)^2)}$$

$$xy' + y = xy^2 \ln(x), \quad y(1) = 1$$

$$\text{Lösung: } y = \frac{2}{x(2 - \ln(x)^2)}$$

$$25. \quad y' - \frac{xy}{2(x^2 - 1)} - \frac{x}{2y} = 0$$

$$u = y^2, \quad u' - \frac{x}{x^2 - 1}u = x$$

$$\text{Lösung: } y^2 = x^2 - 1 + c\sqrt{|x^2 - 1|}, \quad |x| \neq 1$$

$$26. \quad y' - \frac{xy}{2(x^2 - 1)} - \frac{x}{2y} = 0, \quad y(0) = 1$$

$$\text{Lösung: } y^2 = x^2 - 1 + 2\sqrt{|x^2 - 1|}$$

$$27. \quad y' = x^3y^2 + xy$$

$$\text{Lösung: } y = \frac{1}{2 - x^2 + ce^{-x^2/2}}$$

$$y' = x^3y^2 + xy, \quad y(1) = 1$$

$$\text{Lösung: } y = 0, \quad y = \frac{1}{2 - x^2}$$

$$28. \quad x^2y' - y^2 - xy = 0$$

$$\text{Lösung: } y = 0, \quad y = \frac{x}{c - \ln(x)}$$

$$x^2y' - y^2 - xy = 0, \quad y(1) = 1$$

$$\text{Lösung: } y = 0, \quad y = \frac{x}{1 - \ln(x)}$$

29. $x^2 y' + (1-x)y = 0$

Lösung: $y = 0, \quad y = cx e^{1/x}$

$x^2 y' + (1-x)y = 0, \quad y(1) = 1$

Lösung: $y = 0, \quad y = x e^{1/x-1}$

$x^2 y' + (1-x)y = 0, \quad y(0) = 0$

Lösung: $y = 0$ für $x \geq 0$, sonst $y = dx e^{1/x}$, nicht eindeutig

30. $y' = \frac{1}{1 + \tan^2(y)}$

Lösung: $y = \arctan(x+c)$

$y' = \frac{1}{1 + \tan^2(y)}, \quad y(0) = 0$

Lösung: $y = \arctan(x)$

31. $y' = -2xy - 2x^3$

Lösung: $y = 1 - x^2 + ce^{-x^2}$

$y' = -2xy + 2x^3, \quad y(0) = 0$

Lösung: $y = 1 - x^2 - e^{-x^2}$

32. $y' = -y - xy^2$

Lösung: $y = \frac{1}{ce^x - x - 1}$

$y' = -y - xy^2, \quad y(0) = 1$

Lösung: $y = \frac{1}{2e^x - x - 1}$

33. $y' = 2(y+1)\tan(x)$

Lösung: $y = \frac{2c - \cos(2x)}{1 + \cos(2x)}$

$y' = 2(y+1)\tan(x), \quad y(0) = 1$

Lösung: $y = \frac{3 - \cos(2x)}{1 + \cos(2x)}$

34. $y' = x - y^2$, Riccati-DGI

Lösung: $y = \frac{c \operatorname{AiryAi}(1, x) + \operatorname{AiryBi}(1, x)}{c \operatorname{AiryAi}(x) + \operatorname{AiryBi}(x)}$

$y' = x - y^2, \quad y(0) = 0$

Lösung: $y = \frac{\sqrt{3} \operatorname{AiryAi}(1, x) + \operatorname{AiryBi}(1, x)}{\sqrt{3} \operatorname{AiryAi}(x) + \operatorname{AiryBi}(x)}$

$y' = x - y^2, \quad y(0) = -\frac{1}{2}$

Lösung: $y = \frac{c \operatorname{AiryAi}(1, x) + \operatorname{AiryBi}(1, x)}{c \operatorname{AiryAi}(x) + \operatorname{AiryBi}(x)}, \quad c = \frac{\pi 3^{5/6} + 3^{5/3} \Gamma^2(2/3)}{-\pi 3^{1/3} + 3^{7/6} \Gamma^2(2/3)}$

35. $y' = -y + 10 \sin(3x)$

Lösung: $y = c e^{-x} - 3 \cos(3x) + \sin(3x)$

36. $y' = \text{sign}(\sin(x)) - y$

Lösung: $y = \text{sign}(\sin(x)) + c e^{-x}$

$y' = \text{sign}(\sin(x)) - y, \quad y(1) = 1$

Lösung: $y = \text{sign}(\sin(x))$

37. $y' = 1 - y^2$, spezielle Riccati-DGI

Transformation der gDGI auf eine Gleichung 2. Ordnung mittels $y = \frac{u'}{u}$

Lösung: $y = \tanh(x + c), \quad \tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

38. $y' = 1 - y^2, \quad y(0) = 0, \quad 0 \leq x \leq 1$, spezielle Riccati-DGI mit AB

Lösung: $y = \tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

39. $y' = 1 - y^2, \quad y(0) = -\frac{1}{2}, \quad 0 \leq x \leq 1$, spezielle Riccati-DGI mit AB

Lösung: $y = \tanh(x - \text{artanh}(1/2))$

40. $v' = c + k_2 v^2, \quad c > 0, \quad k_2 < 0$, spezielle Riccati-DGI, Bewegungsgleichung

Lösung: $v = \sqrt{ck_2}/k_2 \tan(\sqrt{ck_2}(t + C))$

41. $v' = c + k_2 v^2, \quad v(0) = v_0, \quad c > 0, \quad k_2 < 0$, spezielle Riccati-DGI mit AB

Lösung: $v = \sqrt{ck_2}/k_2 \tan(t \sqrt{ck_2} + \arctan(k_2 v_0 / \sqrt{ck_2}))$

43. $y'(x^2 - 4x + 5) = (x - 2)y$

Lösung: $y = c \sqrt{x^2 - 4x + 5}$

$y'(x^2 - 4x + 5) = (x - 2)y, \quad y(0) = 1$

Lösung: $y = \sqrt{(x^2 - 4x + 5)/5}$

44. $y' = 2|x|, \quad y(-1) = -1$

Lösung: $y = x|x|$

$y' = 2|x|, \quad y(1) = -1$

Lösung: $y = x|x| - 2$

45. $y' = 2x, \quad y(1) = -1$

Lösung: $y = x^2 - 2$

$y' = -2x, \quad y(1) = -1$

Lösung: $y = -x^2$

46. $y' = -200y + 2000 - (1991 + 199x)e^{-x}, \quad y(0) = 10$

Lösung: $y = 10(1 + e^{-200x}) - (10 + x)e^{-x}$

47. $y' = 10 \text{sign}(\sin(10x)) \cos(10x), \quad y(0) = 0$

Lösung: $y = |\sin(10x)|$

48. $y' = -\frac{3}{x}y + \frac{2}{x^3}$

Lösung: $y = \frac{c}{x^3} + \frac{2}{x^2}$

$y' = -\frac{3}{x}y + \frac{2}{x^3}, \quad y(1) = 1$

Lösung: $y = -\frac{1}{x^3} + \frac{2}{x^2}$

49. $y' = 1 + \frac{y}{2x}$

Lösung: $y = 2x + c\sqrt{x}$

$y' = 1 + \frac{y}{2x}, \quad y(1) = 3$

Lösung: $y = 2x + \sqrt{x}$

50. $y' = 1 + \frac{y}{x}$

Lösung: $y = x(c + \ln(x))$

$y' = 1 + \frac{y}{x}, \quad y(1) = 1$

Lösung: $y = x(1 + \ln(x))$

51. $y' = x - y$

Lösung: $y = -1 + x + ce^{-x}$

$y' = x - y, \quad y(0) = 1$

Lösung: $y = -1 + x + 2e^{-x}$

52. $y' = -y \frac{c_1 + x}{c_0 + c_1x + c_2x^2}, \quad c_i \in \mathbb{R}$

Lösung: $y = c(c_0 + c_1x + c_2x^2)^{-1/(2c_2)}$.

$$e^{-(2c_2-1)c_1/(c_2\sqrt{4c_0c_2-c_1^2}) \arctan((c_1+2c_2x)/\sqrt{4c_0c_2-c_1^2})}$$

$y' = -y \frac{c_1 + x}{c_0 + c_1x + c_2x^2}, \quad c_i \in \mathbb{R}, \quad y(0) = \frac{1}{2}$

Lösung: $y = \frac{1}{2}c_0^{1/(2c_2)}(c_0 + c_1x + c_2x^2)^{-1/(2c_2)}$.

$$e^{(2c_2-1)c_1/(c_2\sqrt{4c_0c_2-c_1^2}) (\arctan(c_1/\sqrt{4c_0c_2-c_1^2}) - \arctan((c_1+2c_2x)/\sqrt{4c_0c_2-c_1^2}))}$$

53. $y' = -y^2 + \frac{y}{x}$

Lösung: $y = \frac{2x}{x^2 + 2c}$

$y' = -y^2 + \frac{y}{x}, \quad y(1) = 2$

Lösung: $y = \frac{2}{x}$

$$54. \quad y' = y - \frac{2x}{y}$$

$$\text{Lösung: } y = \pm \sqrt{1 + 2x + ce^{2x}}$$

$$y' = y - \frac{2x}{y}, \quad y(0) = 1$$

$$\text{Lösung: } y = \sqrt{1 + 2x}$$

$$55. \quad y' = (2y \ln(x) - 1) \frac{y}{x}$$

$$\text{Lösung: } y = \frac{1}{2(1 + \ln(x)) + cx}$$

$$y' = (2y \ln(x) - 1) \frac{y}{x}, \quad y(1) = 0.5$$

$$\text{Lösung: } y = \frac{1}{2(1 + \ln(x))}$$

$$56. \quad y' = -200xy^2$$

$$\text{Lösung: } y = \frac{1}{c + 100x^2}$$

$$y' = -200xy^2, \quad y(-1) = 1/101$$

$$\text{Lösung: } y = \frac{1}{1 + 100x^2}$$

$$57. \quad y' = \frac{1}{y + 10^{-p}}, \quad p \in \mathbb{R}$$

$$\text{Lösung: } y = -10^{-p} \pm \sqrt{10^{-2p} + 2x + 2c}$$

$$y' = \frac{1}{y + 10^{-p}}, \quad p \in \mathbb{R}, \quad y(0) = 0$$

$$\text{Lösung: } y = -10^{-p} + \sqrt{10^{-2p} + 2x}$$

$$58. \quad y' = e^{-\sin(x)} - y \cos(x)$$

$$\text{Lösung: } y = (x + c)e^{-\sin(x)}$$

$$y' = e^{-\sin(x)} - y \cos(x), \quad y(0) = 0$$

$$\text{Lösung: } y = xe^{-\sin(x)}$$

$$y' = y \cos(x), \quad y(0) = 1$$

$$\text{Lösung: } y = e^{\sin(x)}$$

$$59. \quad y' = k\sqrt{y}, \quad y(0) = 0, \quad 0 \leq x, \quad k > 0$$

$$\text{Lösung: Sei } \tau \geq 0.$$

$$y = \begin{cases} 0, & \text{falls } 0 \leq x \leq \tau, \\ \frac{1}{4}k^2(x - \tau)^2, & \text{falls } \tau < x \end{cases}$$

bzw. Nulllösung

$$60. \quad y' = -2xy^2$$

$$\text{Lösung: } y = \frac{1}{c + x^2}$$

$$y' = -2xy^2, \quad y(0) = 1$$

$$\text{Lösung: } y = \frac{1}{1 + x^2}$$

$$61. \quad y' = 10 \left(y - \frac{x^2}{1 + x^2} \right) + \frac{2x}{(1 + x^2)^2}, \quad y(0) = y_0$$

$$\text{Lösung: } y = \frac{x^2}{1 + x^2}$$

$$y' = 10 \left(y - \frac{x^2}{1 + x^2} \right) + \frac{2x}{(1 + x^2)^2}, \quad y(0) = 0$$

$$\text{Lösung: } y = y_0 e^{10x} + \frac{x^2}{1 + x^2}$$

$$62. \quad y' = \frac{1}{1 + 4x^2} + ay^2, \quad y(0) = 0, \quad x \in [0, 4], \quad a = -8, 0.4$$

Lösung mittels hypergeometrischer Funktionen

$$63. \quad y' = \frac{1 - x^2 - y^2}{1 + x^2 + xy}, \quad y(0) = 0, \quad x \in [0, 10]$$

$$64. \quad y' = \frac{2x}{y^2}$$

$$\text{Lösung: } y^3 - 3x^2 - c = 0$$

$$y' = \frac{2x}{y^2}, \quad y(0) = 1$$

$$\text{Lösung: } y = \sqrt[3]{3x^2 + 1}$$

$$65. \quad y' = -\frac{1}{\sqrt{y}}$$

$$\text{Lösung: } y^{3/2} + 3x/2 - c = 0$$

$$y' = -\frac{1}{\sqrt{y}}, \quad y(0) = 1$$

$$\text{Lösung: } y = (1 - 3x/2)^{2/3}$$

$$66. \quad y' = y^2$$

$$\text{Lösung: } y = \frac{1}{c - x}$$

$$y' = y^2, \quad y(0) = 1$$

$$\text{Lösung: } y = \frac{1}{1 - x}$$

$$67. \quad y' = \sin\left(\frac{1}{y}\right) - 2, \quad y(0) = y_0$$

$$68. \quad y' = y - x^3 + 3x - 2, \quad y(0) = y_0$$

Lösung: $y = (y_0 - 5)e^x + x^3 + 3x^2 + 3x + 5$

$y' = y - x^3 + 3x - 2, \quad y(0) = 5$

Lösung: $y = x^3 + 3x^2 + 3x + 5$

69. $y' = \sin(y)$

Lösung: $y = \arctan\left(\frac{2ce^x}{1+c^2e^{2x}}, \frac{1-c^2e^{2x}}{1+c^2e^{2x}}\right), \quad \arctan(y, z) = -i \ln\left(\frac{z+iy}{\sqrt{z^2+y^2}}\right)$

$y' = \sin(y), \quad y(0) = 1$

Lösung: $y = \arctan\left(\frac{2e^x \sin(1)(1 - \cos(1))}{\sin(1)^2 + (e^x)^2(2 - 2\cos(1) - \sin(1)^2)}, \frac{\sin(1)^2 + (e^x)^2(-2 + 2\cos(1) + \sin(1)^2)}{\sin(1)^2 + (e^x)^2(2 - 2\cos(1) - \sin(1)^2)}\right)$

$y' = \sin(y), \quad y(0) = k\pi, \quad k \text{ ganzzahlig}$

Stationäre Lösungen: $y = k\pi$

70. $y' = x \cos(y)$

Lösung: $y = \arctan\left(\frac{c^2e^{x^2} - 1}{c^2e^{x^2} + 1}, \frac{2c^2e^{x^2/2}}{c^2e^{x^2} + 1}\right), \quad \arctan(y, z) = -i \ln\left(\frac{z+iy}{\sqrt{z^2+y^2}}\right)$

$y' = x \cos(y), \quad y(0) = y_0$

$y' = x \cos(y), \quad y(-2) = 1$

Lösung: $y = \arctan\left(\frac{(1+\sin(1))e^{x^2-4} - 1 + \sin(1)}{(1+\sin(1))e^{x^2-4} + 1 - \sin(1)}, \frac{2\cos(1)e^{x^2/2-2}}{(1+\sin(1))e^{x^2-4} + 1 - \sin(1)}\right)$

$y' = x \cos(y), \quad y(0) = \pi/2$

Stationäre Lösung: $y = \pi/2$

71. $y' = x + y^n, \quad y(0) = y_0, \quad n \text{ natürlich}$

72. $y' = e^y \sin(x)$

Lösung: $y = -\ln(\cos(x) - c)$

$y' = e^y \sin(x), \quad y(x_0) = y_0$

Lösung: $y = y_0 - \ln(1 + e^{y_0}(\cos(x) - \cos(x_0)))$

73. $y' = xy$

Lösung: $y = ce^{x^2/2}$

$y' = xy, \quad y(0) = 1$

Lösung: $y = e^{x^2/2}$

74. $y' = 2|y|$

Lösung: $y = e^{2x+c}$, falls $y > 0$; $y = e^{c-2x}$, falls $y < 0$

$y' = 2y, \quad y(1) = 1$

Lösung: $y = e^{2x-2}$

$$y' = -2y, \quad y(1) = -1$$

Lösung: $y = -e^{2-2x}$

$$75. \quad y' = \frac{y-2}{x(2x-1)}$$

Lösung: $y = c_1 \frac{2x-1}{x} + \frac{1}{x}$

$$y' = \frac{y-2}{x(2x-1)}, \quad y(1) = 1$$

Lösung: $y = \frac{1}{x}$

$$76. \quad y' = y^2 - x^6$$

Lösung: $y = -x^3 \frac{c \operatorname{BesselI}(-7/8, x^4/4) - \operatorname{BesselK}(7/8, x^4/4)}{c \operatorname{BesselI}(1/8, x^4/4) + \operatorname{BesselK}(1/8, x^4/4)}$

Maple-Notation der Bessel-Funktionen

$$y' = y^2 - x^6, \quad y(0) = -0.1$$

Lösung: $y = -x^3 \frac{4.701307519771402 \operatorname{BesselI}\left(\frac{-7}{8}, \frac{1}{4} x^4\right) - \operatorname{BesselK}\left(\frac{7}{8}, \frac{1}{4} x^4\right)}{4.701307519771402 \operatorname{BesselI}\left(\frac{1}{8}, \frac{1}{4} x^4\right) + \operatorname{BesselK}\left(\frac{1}{8}, \frac{1}{4} x^4\right)}$

$$y' = -y^2 + x^2$$

Lösung: $y = x \frac{c \operatorname{BesselI}(-3/4, x^2/2) - \operatorname{BesselK}(3/4, x^2/2)}{c \operatorname{BesselI}(1/4, x^2/2) + \operatorname{BesselK}(1/4, x^2/2)}$

$$77. \quad y' = 1 + y^2$$

Lösung: $y = \tan(x + c)$

$$y' = 1 + y^2, \quad y(1) = 1$$

Lösung: $y = \tan\left(x + \frac{\pi}{4} - 1\right)$

$$78. \quad y' = 3y - 5x$$

Lösung: $y = \frac{5}{9} + \frac{5}{3}x + c e^{3x}$

$$y' = 3y - 5x, \quad y(1) = 2$$

Lösung: $y = \frac{5}{9} + \frac{5}{3}x - \frac{2}{9} e^{3(x-1)}$

$$79. \quad y' = e^y \sin^2(x) \cos(x)$$

Lösung: $y = \ln(12/(\sin(3x) - 3 \sin(x) - 12c))$

$$y' = e^y \sin^2(x) \cos(x), \quad y(0) = y_0$$

Lösung: $y = y_0 + \ln(12) - \ln(e^{y_0}(\sin(3x) - 3 \sin(x)) + 12)$

1.8.2 Systeme von zwei gDGL 1. Ordnung

$$u' = f(x, u, v) \text{ bzw. } f(x, u, v, u') = 0, \quad u = u(x), \quad \text{AB } u(x_0) = u_0$$

$$v' = g(x, u, v) \text{ bzw. } g(x, u, v, v') = 0, \quad v = v(x), \quad v(x_0) = v_0$$

$$\text{Notation auch mit } x = t, \quad \dot{u} = u', \quad \dot{v} = v', \quad y = (y_1, y_2)^T = (u, v)^T$$

1. Klassisches Räuber-Beute-Modell von Lotka-Volterra

$$\dot{y}_1 = a y_1(1 - y_2), \quad y_1(0) = 3, \quad a = 10 > 0, \quad 0 \leq x \leq 5$$

$$\dot{y}_2 = -y_2(1 - y_1), \quad y_2(0) = 1$$

2. Räuber-Beute-Modell

$$\dot{y}_1 = 1.2y_1 - y_1^2 - \frac{y_1 y_2}{y_1 + 0.2}, \quad y_1(0) = 1, \quad 0 \leq x \leq 30$$

$$\dot{y}_2 = \frac{1.5y_1 y_2}{y_1 + 0.2} - y_2, \quad y_2(0) = 0.75$$

Man teste auch mit den AB $y_1(0) = 0.75, y_2(0) = 0.25$.

3. Abrollen einer Kugel auf einer Falllinie eines parabolischen Zylinders

$$\dot{u} = v, \quad u(0) = 30$$

$$\dot{v} = -\frac{u}{2025 + u^2}(31532 + v^2), \quad v(0) = 0$$

4. $\dot{u} = v$

$$\dot{v} = -u$$

$$\text{Lösung: } u = c_1 \sin(x) + c_2 \cos(x)$$

$$v = c_1 \cos(x) - c_2 \sin(x)$$

5. $\dot{u} = -u + 25v, \quad u(0) = 0$

$$\dot{v} = -2v, \quad v(0) = v_0 > 0$$

$$\text{Lösung: } u = 25v_0(-e^{2x} + e^x)$$

$$v = v_0 e^{-2x}$$

6. System mit moderater Nichtlinearität

$$\dot{u} = -u + 25v - \sqrt{u^2 + v^2} v, \quad u(0) = 0$$

$$\dot{v} = -2v + \sqrt{u^2 + v^2} u, \quad v(0) = v_0 > 0$$

7. $\dot{u} = 2u - 3v, \quad u(0) = 1$

$$\dot{v} = u - 2v, \quad v(0) = 0$$

$$\text{Lösung: } u = \cosh(x) + 2 \sinh(x)$$

$$v = \sinh(x)$$

8. $\dot{u} = v, \quad u(0) = \frac{1}{2}\pi$

$$\dot{v} = -\frac{2 + v^2}{8 - 2 \cos(u)} \sin(u), \quad v(0) = 0$$

9. $\dot{u} = 2u + 4v + e^{ax}, \quad u(0) = 0, \quad a \in \mathbb{R}$

$$\dot{v} = 4u + 2v + e^{ax}, \quad v(0) = 0$$

Lösung: (a) $a \neq -2, a \neq 6$

$$u = \frac{1}{a-6}(e^{ax} - e^{6x})$$

$$v = \frac{1}{a-6}(e^{ax} - e^{6x})$$

(b) $a = -2$

$$u = \frac{1}{8}(e^{6x} - e^{-2x})$$

$$v = \frac{1}{8}(e^{6x} - e^{-2x})$$

(c) $a = 6$

$$u = xe^{6x}, \quad v = xe^{6x}$$

10.
$$y' = \begin{pmatrix} 1 & 1 \\ \omega^2 & 1 \end{pmatrix} y, \quad y = y(t), \quad \omega > 0$$

Lösung: $y_1 = c_1 e^{(1+\omega)t} + c_2 e^{(1-\omega)t}$

$$y_2 = \omega(c_1 e^{(1+\omega)t} - c_2 e^{(1-\omega)t})$$

11.
$$y' = \begin{pmatrix} 0 & 1 \\ -1 & -\frac{1}{100} \end{pmatrix} y$$

Lösung: $y_1 = e^{-t/200} [c_1 \sin(t \sqrt{39999}/200) + c_2 \cos(t \sqrt{39999}/200)]$

$$y_2 = 1/200 e^{-t/200} [c_1(\sqrt{39999} \cos(t \sqrt{39999}/200) - \sin(t \sqrt{39999}/200)) - c_2(\sqrt{39999} \sin(t \sqrt{39999}/200) + \cos(t \sqrt{39999}/200))]$$

12. $y_1' = y_2, \quad y_1(0) = 0$

$$y_2' = -y_1 - \frac{1}{100} y_2, \quad y_2(0) = 1$$

Lösung:

$$y_1 = 200/\sqrt{39999} e^{-t/200} \sin(t \sqrt{39999}/200)$$

$$= 1.0000125 e^{-0.005t} \sin(0.9999875 t)$$

$$y_2 = e^{-t/200} [\cos(t \sqrt{39999}/200) - 1/\sqrt{39999} \sin(t \sqrt{39999}/200)]$$

$$= e^{-0.005t} [\cos(0.9999875 t) - 0.005000062501 \sin(0.9999875 t)]$$

13. Pendel ohne/mit Reibung

$$y' = \begin{pmatrix} 0 & 1 \\ -1 & -k \end{pmatrix} y, \quad y = y(t), \quad k \geq 0$$

Lösung: $y_1 = c_1 e^{(-k+\sqrt{k^2-4})/2} + c_2 e^{(-k-\sqrt{k^2-4})/2}$

$$y_2 = c_1 (-k + \sqrt{k^2-4})/2 e^{(-k+\sqrt{k^2-4})/2} +$$

$$c_2 (-k - \sqrt{k^2-4})/2 e^{(-k-\sqrt{k^2-4})/2}$$

Fallunterscheidung $k < 2, k = 2, k > 2$

$$y' = \begin{pmatrix} 0 & 1 \\ -1 & -100 \end{pmatrix} y, \quad y = y(t), \quad k = 100$$

$$\text{Lösung: } y_1 = c_1 e^{t(-50+7\sqrt{51})} + c_2 e^{t(-50-7\sqrt{51})}, \quad y_2 = y_1'$$

$$y' = \begin{pmatrix} 0 & 1 \\ -1 & -100 \end{pmatrix} y, \quad y(0) = (0, 1)^T, \quad k = 100$$

$$\text{Lösung: } y_1 = \frac{\sqrt{51}}{714} \left(e^{t(-50+7\sqrt{51})} - e^{t(-50-7\sqrt{51})} \right), \quad y_2 = y_1'$$

14. Pendel

$$\dot{y}_1 = y_2, \quad y_1(0) = y_{10}$$

$$\dot{y}_2 = -y_1 - ky_2, \quad y_2(0) = y_{20}$$

$$\text{Lösung: } k = 10/3, \quad y_{10} = y_{20} = 2$$

$$y_1 = 3e^{-t/3} - e^{-3t}$$

$$y_2 = -e^{-t/3} + 3e^{-3t}$$

$$15. \quad y_1' = x - y_2, \quad y_1(0) = 1$$

$$y_2' = -x + y_1 y_2, \quad y_2(0) = 1$$

16. Dynamische Kennlinie des Asynchronmotors

$$\beta \dot{u} = x - u + xv, \quad u(0) = u_0, \quad \beta \in \mathbb{R}^+$$

$$\beta \dot{v} = -xu - v, \quad v(0) = v_0$$

17. Steifes System

$$y' = Ay = \begin{pmatrix} -29\,998 & -39\,996 \\ 29\,997/2 & 19\,997 \end{pmatrix} y$$

$$\text{Lösung: } y_1 = c_1 e^{-x} + c_2 e^{-10000x}$$

$$y_2 = -\frac{3}{4}c_1 e^{-x} - \frac{1}{2}c_2 e^{-10000x}$$

$$y' = Ay, \quad y(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\text{Lösung: } y_1 = -6e^{-x} + 7e^{-10000x}$$

$$y_2 = \frac{9}{2}e^{-x} - \frac{7}{2}e^{-10000x}$$

18. Steifes System

$$y' = Ay = \begin{pmatrix} -216\,686.5 & 39\,346.1 \\ 0 & -1.999\,95 \end{pmatrix} y$$

$$\text{Lösung: } y_1 = c_1 e^{-433373/2x} + c_2 \frac{786922000}{4333690001} e^{-39999/20000x}$$

$$y_2 = c_2 e^{-39999/20000x}$$

$$y' = Ay, \quad y(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\text{Lösung: } y_1 = \frac{786922000}{4333690001} \left(e^{-39999/20000x} - e^{-433373/2x} \right)$$

$$y_2 = e^{-39999/20000x}$$

19. Autonomes System abgeleitet aus der Riccati-DGI $y' = x - y^2$, $y(0) = 0$

$$\dot{y}_1 = y_2 - y_1^2, \quad y_1(0) = 0$$

$$\dot{y}_2 = 1, \quad y_2(0) = 0$$

$$\text{Lösung: } y_1 = y(x)$$

$$y_2 = x$$

20. Verhulst-DGI

$$\dot{y}_1 = -y_1$$

$$\dot{y}_2 = 1 - y_1^2 - y_2^2$$

$$\text{Lösung: } y_1 = c_1 e^{-x}$$

$$y_2 = \dots$$

21. $\dot{u} = -v + u(s - u^2 - v^2)$, $u(0) = \sqrt{s}$, $s \geq 0$

$$\dot{v} = u + v(s - u^2 - v^2), \quad v(0) = 0$$

$$\text{Lösung: } u = \sqrt{s} \cos(t)$$

$$v = \sqrt{s} \sin(t)$$

22. Hopf-DGI

$$\dot{u} = -v + u(1 - u^2 - v^2)$$

$$\dot{v} = u + v(1 - u^2 - v^2)$$

$$\text{Lösung: } u = 0, \text{ spezielle}$$

$$v = 0$$

23. $\dot{u} = \omega v + \omega u(\lambda - (u^2 + v^2 - 1)^2)$, $u(0) = \sqrt{1 + \sqrt{\lambda}}$

$$\dot{v} = -\omega u + \omega v(\lambda - (u^2 + v^2 - 1)^2), \quad v(0) = 0$$

$$\omega = \pi/10, \quad \lambda \in [0, 1]$$

$$\text{Lösung: } u = \sqrt{1 + \sqrt{\lambda}} \cos(\omega t)$$

$$v = -\sqrt{1 + \sqrt{\lambda}} \sin(\omega t)$$

24. Räuber-Beute-Modell

$$\dot{u} = u(a - bv), \quad u(0) = u_0, \quad a, b, c, d > 0$$

$$\dot{v} = -v(c - du), \quad v(0) = v_0$$

25. Gedämpfte Schwingung abgeleitet aus der gDGI 2. Ordnung

$$\dot{x}_1 = x_2, \quad x_1(0) = 1$$

$$\dot{x}_2 = -\frac{D}{m}x_1 - \frac{b}{m}x_2, \quad x_2(0) = 1$$

$$m > 0, \quad D > 0, \quad |b| \ll 1, \quad \delta = \frac{b}{2m}, \quad \omega_d = \sqrt{\frac{D}{m} - \delta^2}$$

$$\text{Lösung: } x_1 = C e^{-\delta t} \sin(\omega_d t + \Phi), \quad \Phi = \operatorname{arccot}\left(\frac{1+\delta}{\omega_d}\right), \quad C = \frac{1}{\sin(\Phi)}$$

$$x_2 = \dot{x}_1$$

26. Aus gDGI 2. Ordnung

$$y_1' = y_2, \quad y_1(0) = 1$$

$$y_2' = y_1 - y_2, \quad y_2(0) = 0$$

$$\text{Lösung: } y_1 = \frac{1}{10}(5 + \sqrt{5}) e^{x(-1+\sqrt{5})/2} + \frac{1}{10}(5 - \sqrt{5}) e^{x(-1-\sqrt{5})/2}$$

$$y_2 = y_1'$$

27. Aus gDGI 2. Ordnung

$$y_1' = y_2, \quad y_1(0) = 0$$

$$y_2' = y_1 y_2 + 1, \quad y_2(0) = s = -0.5$$

28. $y_1' = x(y_2 - y_1)$

$$y_2' = x(y_2 + y_1)$$

Lösung:

$$y_1 = c_1 \sinh(x^2 \sqrt{2}/2) + c_2 \cosh(x^2 \sqrt{2}/2)$$

$$y_2 = c_1 \sinh(x^2 \sqrt{2}/2) + c_2 \cosh(x^2 \sqrt{2}/2) +$$

$$+ \sqrt{2} (c_1 \cosh(x^2 \sqrt{2}/2) + c_2 \sinh(x^2 \sqrt{2}/2))$$

$$y_1' = x(y_2 - y_1), \quad y_1(0) = 1$$

$$y_2' = x(y_2 + y_1), \quad y_2(0) = 1$$

Lösung:

$$y_1 = \cosh(x^2 \sqrt{2}/2)$$

$$y_2 = \cosh(x^2 \sqrt{2}/2) + \sqrt{2} \sinh(x^2 \sqrt{2}/2)$$

1.8.3 Systeme von n gDGI 1. Ordnung

$n \geq 1, \quad x \in [x_0, x_1], \quad y_i = y_i(x)$

$$y_1' = f_1(x, y_1, y_2, \dots, y_n), \quad \text{AB } y_1(x_0) = y_{1,0}$$

$$y_2' = f_2(x, y_1, y_2, \dots, y_n), \quad y_2(x_0) = y_{2,0}$$

.....

$$y_n' = f_n(x, y_1, y_2, \dots, y_n), \quad y_n(x_0) = y_{n,0}$$

Notation auch mit $x = t, \quad \dot{y}_i = y_i'$

1. Zweispezimodell, $n = 2$

$$\dot{u} = au(1 - v), \quad u(0) = 1, \quad a = 2$$

$$\dot{v} = -cv(1 - u), \quad v(0) = 3, \quad c = 1$$

2. System, $n = 4$

$$y_1' - 2y_2' + y_3' = -2, \quad y_1(0) = 1$$

$$-y_1' + 3y_2' - 2y_3' + y_4' = 3 + x, \quad y_2(0) = -1$$

$$y_3' - y_4' = 0, \quad y_3(0) = -3$$

$$3y_4' - 5y_1' - 2y_3 = 0, \quad y_4(0) = 2$$

$$\begin{aligned} \text{Lösung: } y_1 &= e^{-2x} + 2 \sin(x) \\ y_2 &= \sin(x) - \cos(x) + x \\ y_3 &= -e^{-2x} - 2 \cos(x) \\ y_4 &= 2e^{-2x} + 2 \sin(x) \end{aligned}$$

3. System, $n = 3$

$$\begin{aligned} \dot{y}_1 &= -y_1 - 24y_2, & y_1(0) &= 2 \\ \dot{y}_2 &= -25y_2, & y_2(0) &= 1 \\ \dot{y}_3 &= 125y_2 - 150y_3, & y_3(0) &= 0 \end{aligned}$$

$$\begin{aligned} \text{Lösung: } y_1 &= e^{-25x} + e^{-x} \\ y_2 &= e^{-25x} \\ y_3 &= e^{-25x} - e^{-150x} \end{aligned}$$

4. Lorenz-Modell der Wärmeströmung in einer Flüssigkeitsschicht, $n = 3$

$$\begin{aligned} y_1' &= -\sigma y_1 + \sigma y_2, & y_1(0) &= y_{10} \\ y_2' &= -y_1 y_3 + r y_1 - y_2, & y_2(0) &= y_{20} \\ y_3' &= y_1 y_2 - b y_3, & y_3(0) &= y_{30} \end{aligned}$$

σ Prandtl-Zahl, r reduzierte Rayleigh-Zahl, b Wellenzahl

5. Zeitliche Verbreitung von Epidemien ohne Migration, $n = 3$, $y_i = y_i(t)$

r, γ Parameter

$$\begin{aligned} \dot{y}_1 &= -r y_1 y_2, & y_1 & \text{aufnahmefähige Population} \\ \dot{y}_2 &= r y_1 y_2 - \gamma y_2, & y_2 & \text{infizierte Population} \\ \dot{y}_3 &= \gamma y_2, & y_3 & \text{resistente bzw. gestorbene Population} \end{aligned}$$

6. Rössler-System, $n = 3$

$$\begin{aligned} \dot{y}_1 &= -y_2 - y_3, & y_1(0) &= y_{10} \\ \dot{y}_2 &= y_1 + a y_2, & y_2(0) &= y_{20} \\ \dot{y}_3 &= b y_1 - c y_3 + y_1 y_3, & y_3(0) &= y_{30} \end{aligned}$$

7. Drei-Wellen-Wechselwirkung, $n = 3$

$$\begin{aligned} \dot{y}_0 &= y_0 + 2y_0 y_1^2 \sin(\vartheta) \\ \dot{y}_1 &= -y_1 - y_0^2 y_1 \sin(\vartheta) \\ \dot{y}_2 &= 2(y_1^2 - y_0^2) - 2\delta + 2(2y_1^2 - y_0^2) \cos(\vartheta) \end{aligned}$$

8. Räuber-Beute-Modell

$$\begin{aligned} \dot{y}_1 &= 10y_1(1 - y_2), & y_1(0) &= 3, \quad 0 \leq x \leq 5 \\ \dot{y}_2 &= -y_2(1 - y_1), & y_2(0) &= 1 \end{aligned}$$

Man teste dazu Runge-Kutta-Verfahren (RKV) mit verschiedenen Schrittweiten, z.B. $h = 0.025 \dots 0.09$.

9. Räuber-Beute-Modell

$$\dot{y}_1 = 1.2y_1 - y_1^2 - \frac{y_1 y_2}{y_1 + 0.2}, \quad y_1(0) = 1, \quad 0 \leq x \leq 30$$

$$\dot{y}_2 = \frac{1.5y_1 y_2}{y_1 + 0.2} - y_2, \quad y_2(0) = 0.75$$

Man teste auch die AB $y_1(0) = 0.75$, $y_2(0) = 0.25$.

Die Schrittweite von RKV sei z.B. $h = 0.1$.

10. Steifes System, $n = 3$

$$\dot{y}_1 = -0.5y_1 + 32.6y_2 + 35.7y_3, \quad y_1(0) = 4, \quad x \in [0, 1]$$

$$\dot{y}_2 = -48y_2 + 9y_3, \quad y_2(0) = 13$$

$$\dot{y}_3 = 9y_2 - 72y_3, \quad y_3(0) = 1$$

$$\text{Lösung: } y_1 = -12e^{-45x} + e^{-75x} + 15e^{-0.5x}$$

$$y_2 = 12e^{-45x} + e^{-75x}$$

$$y_3 = 4e^{-45x} - 3e^{-75x}$$

11. Steifes System, $n = 3$

Die anfängliche Steifheit nimmt ab für $x \gg 0$.

$$\dot{y}_1 = -0.1y_1 + 100y_2y_3, \quad y_1(0) = 4, \quad x \in [0, 10]$$

$$\dot{y}_2 = 0.1y_1 - 100y_2y_3 - 500y_2^2, \quad y_2(0) = 0$$

$$\dot{y}_3 = 500y_2^2 - 0.5y_3, \quad y_3(0) = 0.5$$

Man teste RKV mit den Schrittweiten $h = 10^{-4} \dots 0.05$.

12. System, $n = 3$

$$\dot{y}_1 = -21y_1 + 19y_2 - 20y_3, \quad y_1(0) = 1$$

$$\dot{y}_2 = 19y_1 - 21y_2 + 20y_3, \quad y_2(0) = 0$$

$$\dot{y}_3 = 40y_1 - 40y_2 - 40y_3, \quad y_3(0) = -1$$

$$\text{Lösung: } y_1 = \frac{1}{2}[e^{-2x} + e^{-40x}(\sin(40x) + \cos(40x))]$$

$$y_2 = \frac{1}{2}[e^{-2x} - e^{-40x}(\sin(40x) + \cos(40x))]$$

$$y_3 = e^{-40x}[\sin(40x) - \cos(40x)]$$

13. Steifes System, $n = 3$

$$\dot{y}_1 = -\frac{1}{100}(y_1 - y_2), \quad y_1(0) = 0$$

$$\dot{y}_2 = y_1 - y_2 - y_1 y_3, \quad y_2(0) = 1$$

$$\dot{y}_3 = y_1 y_2 - 100y_3, \quad y_3(0) = 1$$

1.8.4 Skalare gDGI 2. und höherer Ordnung

$y'' = f(x, y, y')$ bzw. $f(x, y, y', y'') = 0$, $y = y(x)$, AB $y(x_0) = y_0$, $y'(x_0) = y'_0$

Notation auch mit $x = t$, $\dot{y} = y'$, $\ddot{y} = y''$

1. $y'' = 0$

Lösung: $y = c_1 + c_2 x$

2. $y'' - k^2y = 0, \quad k > 0$

Lösung: $y = c_1e^{kx} + c_2e^{-kx}$

3. $y'' - k^2y = 0, \quad k > 0, \quad y(0) = y_0, \quad y'(0) = y'_0$

Lösung: $y = \frac{1}{2} \left(y_0 + \frac{y'_0}{k} \right) e^{kx} + \frac{1}{2} \left(y_0 - \frac{y'_0}{k} \right) e^{-kx}$

$y = e^{kx} + e^{-kx}, \quad \text{falls } y_0 = 2, \quad y'_0 = 0$

4. Pendel

$y'' + k^2y = 0, \quad k > 0$

Lösung: $y = c_1 \sin(kx) + c_2 \cos(kx)$

5. Pendel mit AB

$y'' + k^2y = 0, \quad k > 0, \quad y(0) = y_0, \quad y'(0) = y'_0$

Lösung: $y = \frac{y'_0}{k} \sin(kx) + y_0 \cos(kx)$

$y = \frac{y'_0}{k} \sin(kx), \quad \text{falls } y_0 = 0$

$y = \sin(x), \quad \text{falls } y_0 = 0, \quad y'_0 = 1, \quad k = 1$

$y = c_1 \sin(kx) + y_0 \cos(kx), \quad \text{falls die AB } y'(0) \text{ fehlt}$

6. Fadenpendel

$y'' + ay' + by = 0, \quad D = a^2 - 4b \quad \text{Diskriminante}$

Lösung: (a) $D > 0$

$y = c_1e^{k_1x} + c_2e^{k_2x}$

$k_1 = -\frac{1}{2}(a - \sqrt{D}), \quad k_2 = -\frac{1}{2}(a + \sqrt{D}),$

z.B. falls $b < 0$, dann $D > 0$

(b) $D = 0$

$y = c_1e^{k_1x} + c_2xe^{k_1x}$

$k_1 = -a/2$

z.B. falls $b = 0, \quad a = 0$, dann $D = 0$

(c) $D < 0$

$y = e^{-\frac{ax}{2}} (c_1 \sin(\frac{1}{2}\sqrt{|D|x}) + c_2 \cos(\frac{1}{2}\sqrt{|D|x}))$

$k_1 = -\frac{a}{2} + \frac{1}{2}\sqrt{-D} \quad i, \quad k_2 = -\frac{a}{2} - \frac{1}{2}\sqrt{-D} \quad i$

z.B. falls $b \gg 0$, dann $D < 0$;

$a > 0 \rightarrow$ gedämpftes Pendel, $a < 0 \rightarrow$ erregtes Pendel

7. $y'' + ky' + y = 0, \quad y(0) = y_0, \quad y'(0) = y'_0, \quad |k| < 2, \quad D = k^2 - 4 < 0 \quad \text{Diskriminante}$

Lösung: $y = e^{-\frac{kx}{2}} \left(\frac{2y'_0 + ky_0}{\sqrt{4 - k^2}} \sin(\frac{1}{2}\sqrt{4 - k^2}x) + y_0 \cos(\frac{1}{2}\sqrt{4 - k^2}x) \right)$

8. $y'' - 2y' = x$

Lösung: $y = -\frac{1}{4}x^2 - \frac{1}{4}x + c_1 + c_2e^{2x}$

9. $y'' - 2y' = 3e^{2x} + (x^2 - 3x + 1)e^x$

Lösung: $y = \frac{3}{2}xe^{2x} + (-x^2 - 3x - 3)e^x + c_1 + c_2e^{2x}$

10. $y'' - 2y' = -\frac{1}{4}x \cos\left(\frac{x}{2}\right) + 3 - (x-1)e^{2x}$

Lösung: $y = \left(\frac{1}{17}x + \frac{152}{289}\right) \cos\left(\frac{x}{2}\right) + \left(\frac{4}{17}x - \frac{4}{289}\right) \sin\left(\frac{x}{2}\right) +$
 $-\frac{3}{2}x + \left(-\frac{1}{4}x^2 + \frac{3}{4}x\right) e^{2x} + c_1 + c_2e^{2x}$

11. $y'' + 5y' + 6y = 0$

Lösung: $y = c_1e^{-2x} + c_2e^{-3x}$

$y'' + 5y' + 6y = 0, \quad y(0) = 0, \quad y'(0) = 1$

Lösung: $y = e^{-2x} - e^{-3x}$

$y'' + 5y' + 6y = 6x^2 + 10x$

Lösung: $y = c_1e^{-2x} + c_2e^{-3x} - \frac{1}{3} + x^2$

$y'' + 5y' + 6y = 6x^2 + 10x, \quad y(0) = 0, \quad y'(0) = 1$

Lösung: $y = 2e^{-2x} - \frac{5}{3}e^{-3x} - \frac{1}{3} + x^2$

12. Resonanz zwischen Eigenschwingung und Störung

$y'' + 6y' + 5y = 3e^{-x}$

Lösung: $y = c_1e^{-x} + c_2e^{-5x} + \frac{3}{4}xe^{-x}$

13. $y'' + 6y' + 25y = 3 \sin(4x), \quad y(0) = y'(0) = 0$

Lösung: $y = \frac{1}{73}[e^{-3x}(3 \sin(4x) + 8 \cos(4x)) + 3 \sin(4x) - 8 \cos(4x)]$

14. $y'' - 4y' + 4y = 0$

Lösung: $y = (c_1 + c_2x)e^{2x}$

15. $y'' + y' + \frac{1}{4}y = 0$

Lösung: $y = (c_1 + c_2x)e^{-x/2}$

16. Schwingungs-DGL ohne Einwirkung äußerer Kräfte

$y'' + 2hy' + k^2y = 0$

Lösung: $y = c_1e^{(-h+\sqrt{h^2-k^2})x} + c_2e^{(-h-\sqrt{h^2-k^2})x}$

$y'' + 2hy' + k^2y = 0, \quad h = 50, \quad k = 1$

Lösung: $y = c_1e^{(-50+7\sqrt{51})x} + c_2e^{(-50-7\sqrt{51})x}$

$$y'' + 100y' + y = 0, \quad y(0) = 0, \quad y'(0) = 1$$

$$\text{Lösung: } y = \frac{\sqrt{51}}{714} \left(e^{(-50+7\sqrt{51})x} - e^{(-50-7\sqrt{51})x} \right)$$

17. Erzwungene Schwingung

$$y'' + 2\rho y' + \omega^2 y = a \sin(\omega_1 x), \quad y(0) = 0, \quad y'(0) = \omega$$

$$\begin{aligned} \text{Lösung: } y = & \frac{1}{2} \frac{\sqrt{\rho^2 - \omega^2} (\omega\omega_1^2 + 2\omega\rho^2 - \omega^3 - 2\omega\rho\sqrt{\rho^2 - \omega^2} + \omega_1 a)}{\rho^2\omega_1^2 - \omega^2\omega_1^2 + 2\rho^4 - 3\rho^2\omega^2 + \omega^4 - 2\rho^3\sqrt{\rho^2 - \omega^2} + 2\omega^2\rho\sqrt{\rho^2 - \omega^2}} \\ & \cdot e^{(-\rho + \sqrt{\rho^2 - \omega^2})x} \\ & - \frac{1}{2} \frac{2\omega\rho^2 + 2\omega\rho\sqrt{\rho^2 - \omega^2} + \omega\omega_1^2 - \omega^3 + \omega_1 a}{\sqrt{\rho^2 - \omega^2} (2\rho^2 + 2\rho\sqrt{\rho^2 - \omega^2} - \omega^2 + \omega_1^2)} e^{(-\rho - \sqrt{\rho^2 - \omega^2})x} \\ & + \frac{a[(\omega^2 - \omega_1^2) \sin(\omega_1 x) - 2\omega_1\rho \cos(\omega_1 x)]}{\omega_1^4 + \omega_1^2(4\rho^2 - 2\omega^2) + \omega^4} \end{aligned}$$

$$18. \quad y\ddot{y} = \dot{x}^2 + acy^2 - cy\dot{y} + cy^2\dot{y} - acy^3, \quad a = 2, \quad c = 1, \quad y(0) = 1, \quad \dot{y}(0) = \dot{y}_0$$

$$19. \quad y\ddot{y} = \dot{y}^2 + acy^2 + ay\dot{y} - ay^2\dot{y} - acy^3, \quad a = 2, \quad c = 1, \quad y(0) = 3, \quad \dot{y}(0) = \dot{y}_0$$

20. Besselsche DGI

$$x^2 y'' + xy' + (x^2 - k^2)y = 0, \quad k \in \mathbb{Z}$$

$$\text{Lösung: } y = J_0(x), \quad \text{falls } k = 0, \quad y(0) = 1, \quad y'(0) = 0, \quad \text{Bessel-Funktion 1. Art}$$

$$y = J_1(x), \quad \text{falls } k = 1, \quad y(0) = 0, \quad y'(0) = 0.5$$

21. Pendelgleichungen

$$y'' + \sin(y) = 0$$

$$y'' + \sin(y) = 0, \quad y(0) = 0, \quad y'(0) = 2$$

$$y'' + \sin(y) = 0, \quad y(0) = 0, \quad y'(0) = 0$$

$$\text{Lösung: } y = 0$$

$$22. \quad y'' - 3y' + 2y = 0, \quad y(0) = 1, \quad y'(0) = 2$$

$$\text{Lösung: } y = e^{2x}$$

$$23. \quad y'' + 3y' + 2y = 0, \quad y(0) = 1, \quad y'(0) = -1$$

$$\text{Lösung: } y = e^{-x}$$

$$24. \quad y'' + 2y' + y = xe^{-x}$$

$$\text{Lösung: } y = (c_1 + c_2 x)e^{-x} + \frac{1}{6}x^3 e^{-x}$$

$$25. \quad y'' + 2y' + y = -11 \cos(2x) + 2 \sin(2x)$$

$$\text{Lösung: } y = (c_1 + c_2 x)e^{-x} + \cos(2x) - 2 \sin(2x)$$

$$26. \quad y'' + 2y' + y = 25 \sin(2x)$$

$$\text{Lösung: } y = (c_1 + c_2 x)e^{-x} - 4 \cos(2x) - 3 \sin(2x)$$

27. $\ddot{y}(y-1) - 2\dot{y}^2 = 0$

Lösung: $y = 1 - \frac{c_1}{x - c_2}, y = c$

28. $x^2yy'' - (y - xy')^2 = 0$

$$y = e^u, \quad x^2u'' + 2xu' - 1 = 0$$

$$u' = v, \quad x^2v' + 2xv - 1 = 0$$

$$v = \frac{1}{x} - \frac{c_1}{x^2}, \quad u = \ln(c_2x) + \frac{c_1}{x}$$

Lösung: $y = c_2xe^{\frac{c_1}{x}}$

29. Abrollen einer Kugel auf der Falllinie eines parabolischen Zylinders

$$\ddot{y} + \frac{y}{2025 + y^2}(31532 + \dot{y}^2) = 0, \quad y(0) = 30, \quad \dot{y}(0) = 0,$$

30. Rollpendel

$$\left(\frac{\Theta}{m} + (r^2 + s^2 - 2rs \cos(\phi)) \right) \ddot{\phi} + (r\dot{\phi}^2 + g)s \sin(\phi) = 0$$

31. Airys Gleichung

$$y'' - xy = 0, \quad y(0) = y_0, \quad y'(0) = y'_0$$

Lösung: $y = \frac{1}{6\Gamma(2/3)}(-2 \cdot 3^{5/6}y'_0\pi - 3 \cdot 3^{2/3}y_0\Gamma(2/3)^2)\text{AiryAi}(x) + 3^{1/3}(2y'_0\pi + 3^{5/6}y_0\Gamma(2/3)^2)\text{AiryBi}(x)$

32. $y'' - y = 2x, \quad y(0) = 0, \quad y'(0) = 0$

Lösung: $y = e^x - e^{-x} - 2x$

33. $y'' + 5y' + 4y = 1 - e^x, \quad y(0) = y'(0) = 0$

Lösung: $y = \frac{1}{4} - \frac{1}{3}xe^{-x} - \frac{2}{9}e^{-x} - \frac{1}{36}e^{-4x}$

34. $y'' - 3y' + 5y = 1, \quad y(0) = 0, \quad y'(0) = 1$

Lösung: $y = \frac{1}{5} + e^{\frac{3}{2}x} \left(-\frac{1}{5} \cos\left(\frac{\sqrt{11}}{2}x\right) + \frac{13}{55}\sqrt{11} \sin\left(\frac{\sqrt{11}}{2}x\right) \right)$

35. $y'' + 12y' + 20y = 0, \quad y(0) = 4, \quad y'(0) = 0$

Lösung: $y = 5e^{-2x} - e^{-10x}$

36. $y'' + 12y' + 20y = 0, \quad y(0) = \frac{1}{2}, \quad y'(0) = -1$

Lösung: $y = \frac{1}{2}e^{-2x}$

37. $y'' + \omega^2y - \varepsilon f(x, y, y') = 0, \quad \omega \in \mathbb{R}^+, \quad \varepsilon \in \mathbb{R}, \quad y(0) = y_0, \quad y'(0) = y'_0$

Lösung für $\varepsilon = 0$: $y = a \sin(\omega x + \phi)$

38. $y'' + y' + \frac{1}{4}y = 0$

Lösung: $y = (c_1 + c_2x)e^{-x/2}$

39. $y'' + y - \frac{y^3}{6} = 2 \sin(2.78535x), \quad y(0) = 0, \quad y'(0) = 0$

40. $y'' + y - \varepsilon y^3 = 0$, $\varepsilon \in \mathbb{R}$, $y(0) = 1$, $y'(0) = 0$

41. $y'' + \frac{2}{x}y' + y^p = 0$, $p \in \mathbb{R}^+$, $y(0) = 1$, $y'(0) = 0$

Lösung für $p = 5$: $y = \frac{1}{\sqrt{1+x^2/3}}$

42. $\ddot{y} + y^3 = 0$, $y(0) = 1$, $\dot{y}(0) = 0$

43. $\ddot{y} - 6y^2 = 0$, $y(0) = 1$, $\dot{y}(0) = 0$, $x \in [0, 1]$

$\ddot{y} - 2y^3 - xy = 0$, $y(0) = 1$, $\dot{y}(0) = 0$, $x \in [0, 1]$

$\ddot{y} - 6y^2 = x$, $y(0) = 1$, $\dot{y}(0) = 0$, $x \in [0, 1]$

44. $\ddot{y} + 2000\dot{y} + 10^6y = 0$

Lösung: $y = (c_1 + c_2x)e^{-1000x}$

45. $\ddot{y} + 100001\dot{y} + 10^5y = \sin(x)$

Lösung: $y = c_1e^{-x} + c_2e^{-100000x} - \frac{100001}{20000000002} \cos(x) + \frac{99999}{20000000002} \sin(x)$

46. $\ddot{y} + 100.1\dot{y} + 1000y = x^2$

Lösung: $y = c_1e^{x(-1001+3\sqrt{66889})/20} + c_2e^{-x(1001+3\sqrt{66889})/20} + \frac{902001}{50000000000} - \frac{1001}{5000000}x + \frac{1}{1000}x^2$

47. $\ddot{y} + 100.1\dot{y} + 1000y = -x^2$

Lösung: $y = c_1e^{x(-1001+3\sqrt{66889})/20} + c_2e^{-x(1001+3\sqrt{66889})/20} - \frac{902001}{50000000000} + \frac{1001}{5000000}x - \frac{1}{1000}x^2$

48. $\ddot{y} - 100.1\dot{y} + 1000y = x^3$

Lösung: $y = c_1e^{x(1001+3\sqrt{66889})/20} + c_2e^{x(1001-3\sqrt{66889})/20} + \frac{2408409003}{500000000000000} + \frac{2706003}{50000000000}x + \frac{3003}{10000000}x^2 + \frac{1}{1000}x^3$

49. Van der Polsche DGI

$\ddot{y} - \varepsilon(1 - y^2)\dot{y} + y = 0$, $y(0) = y_0$, $\dot{y}(0) = \dot{y}_0$, $t \geq 0$, $\varepsilon \geq 0$

Der Parameter ε legt die Einschwingdauer fest.

Lösung für $\varepsilon = 0$: $y(t) = \dot{y}_0 \sin(t) + y_0 \cos(t)$

AB $y(0) = 0$, $\dot{y}(0) = -0.05$

$y(0) = 2$, $\dot{y}(0) = 0$

50. Rotation eines Pendels mit der Winkelgeschwindigkeit ω

$\ddot{y} + \sin(y) - \frac{1}{2}\omega^2 \sin(2y) = 0$, $y(0) = \frac{\pi}{4}$, $\dot{y}(0) = 0$, $\omega \in [0, 10]$

51. Pendel mit kleiner Reibung, $y = y(t)$

$\ddot{y} + \frac{3}{25}\dot{y} + 2y = 0$, $y(0) = 1$, $\dot{y}(0) = 0$

Lösung: $y = e^{-3t/50} [\cos(t\sqrt{4991}/50) + 3/\sqrt{4991} \sin(t\sqrt{4991}/50)]$

52. $y'' + y' - y = 0$

Lösung: $y = c_1 e^{x(-1+\sqrt{5})/2} + c_2 e^{x(-1-\sqrt{5})/2}$

$y'' + y' - y = 0, \quad y(0) = 1, \quad y'(0) = 0$

Lösung: $y = \frac{1}{10}(5 + \sqrt{5}) e^{x(-1+\sqrt{5})/2} + \frac{1}{10}(5 - \sqrt{5}) e^{x(-1-\sqrt{5})/2}$

53. $y'' + y' - y + 1 = 0$

Lösung: $y = 1 + c_1 e^{x(-1+\sqrt{5})/2} + c_2 e^{x(-1-\sqrt{5})/2}$

$y'' + y' - y + 1 = 0, \quad y(0) = 1, \quad y'(0) = 0$

Lösung: $y = 1$

54. $my'' + by' + Dy = 0, \quad m > 0, \quad D > 0, \quad |b| \ll 1$

Lösung: $y = Ce^{-\delta x} \sin(\omega_d x + \Phi), \quad \delta = \frac{b}{2m}, \quad \omega_d = \sqrt{\frac{D}{m} - \delta^2}$

$my'' + by' + Dy = 0, \quad y(0) = 1, \quad y'(0) = 1$

Lösung: $y = Ce^{-\delta x} \sin(\omega_d x + \Phi), \quad \Phi = \operatorname{arccot}\left(\frac{1+\delta}{\omega_d}\right), \quad C = \frac{1}{\sin(\Phi)}$

55. $y'' - yy' - 1 = 0, \quad y(0) = 0, \quad y'(0) = s = -0.5$

56. $y'' - 6y - x = 0$

Lösung: $y = -x/6 + c_1 e^{x\sqrt{6}} + c_2 e^{-x\sqrt{6}}$

$y'' - 6y - x = 0, \quad y(0) = 1, \quad y'(0) = 0$

Lösung: $y = -x/6 + (36 + \sqrt{6})/72 e^{x\sqrt{6}} + (36 - \sqrt{6})/72 e^{-x\sqrt{6}}$

57. $y''' + \frac{1}{2}y''^3 = 0, \quad y(0) = 1, \quad y'(0) = 3, \quad y''(0) = 1$

Lösung: $y = \frac{4}{3}(x+1)^{3/2} + x - \frac{1}{3}$ mittels Ansatz $y'' = z$

58. $y''' - 2y'' - y' + 2y = 4x$

Lösung: $y = 1 + 2x + c_1 e^x + c_2 e^{-x} + c_3 e^{2x}$

$y''' - 2y'' - y' + 2y = 4x, \quad y(1) = 1, \quad y'(1) = 1, \quad y''(1) = 1$

Lösung: $y = 1 + 2x - 3e^{x-1} + e^{2x-2}$

59. $y^{(4)} + 3y''' + 3y'' + y' = 10e^{-2x}$

Lösung: $y = 5e^{-2x} - e^{-x}[c_1 + c_2(x+1) + c_3(x^2 + 2x + 2)] + c_4$

$y^{(4)} + 3y''' + 3y'' + y' = 10e^{-2x}, \quad y(0) = 1, \quad y'(0) = 3, \quad y''(0) = 1, \quad y'''(0) = 0$

Lösung: $y = 5e^{-2x} - e^{-x}(22 + 9x + \frac{15}{2}x^2) + 18$

60. $y^{(5)} - y = x$

Lösung: $y = -x + c_1 e^x +$

$+ c_2 e^{(-1+\sqrt{5})/4 x} \cos(\sqrt{10+2\sqrt{5}}/4 x) + c_3 e^{(-1+\sqrt{5})/4 x} \sin(\sqrt{10+2\sqrt{5}}/4 x)$
 $+ c_4 e^{-(1+\sqrt{5})/4 x} \cos(\sqrt{10-2\sqrt{5}}/4 x) + c_5 e^{-(1+\sqrt{5})/4 x} \sin(\sqrt{10-2\sqrt{5}}/4 x)$

61. $y^{(5)} - y^2 = x$

Kapitel 2

Differentialgleichungen mit Maple

Auch wenn man einige wichtige Typen von Differentialgleichungen analytisch lösen kann, so muss doch festgestellt werden, dass es für die meisten Gleichungen keine Möglichkeit für eine geschlossene Lösung gibt. Dort werden dann numerische Verfahren eingesetzt.

In [119] findet der Leser einen Zugang, wie ausgehend von der mathematischen Betrachtung von Differentialgleichungen der Übergang zu ihrer analytischen und numerischen Lösung mittels des CAS Maple erfolgt. Es werden jedoch die grundlegenden Kenntnisse in der Theorie der Differentialgleichungen vorausgesetzt.

In diesem Abschnitt sollen daran anknüpfend unter verschiedenen Aspekten gDGL und SysgDGL mit Maple behandelt werden. Das betrifft

- die analytische Lösung von gDGL und AWP,
- die Lösungsdarstellung,
- die Definitionsbereiche von Lösungen,
- die Eindeutigkeit der Lösung eines AWP,
- die Situation von mehreren Lösungen eines AWP,
- ausgewählte SysgDGL und
- spezielle gDGL.

Dabei werden zum besseren Verständnis für den Leser die Hinweise zu den Differentialgleichungen in der Maple-Online-Hilfe auszugsweise zitiert sowie die in Maple vorliegenden Kommandos systematisch einbezogen und grob erläutert. Ein größeres Verständnis für die darin implementierten Algorithmen und numerischen Löser für gDGL wird durch detaillierte Untersuchungen in den nachfolgenden Kapiteln angestrebt, zumindest in wichtigen Gesichtspunkten wie die Konstruktion der Verfahren sowie ihre Eigenschaften wie z.B. Genauigkeit, Stabilität und Effizienz.

Es werden dazu zahlreiche Beispiele und Berechnungen vorgeführt. Dabei ist unser Ziel, dass sich sowohl der Schwierigkeitsgrad der Probleme vergrößern als auch die gewonnenen Erkenntnisse zunehmen sollen. Der Leser wird bzw. kann nach dem Studium einiger Beispiele durchaus zum nächsten Abschnitt übergehen, wird aber möglicherweise aus gegebenem Anlass später dann zu den weiteren Beispielen zurückkehren.

Aus der Online-Hilfe der CAS [67], [68] sowie aus anderen Quellen und Programmen kann der Leser weitere Beispiele beliebig ergänzen.

2.1 Analytische Lösung von Differentialgleichungen

Zunächst verwenden wir die Maple-Funktion `dsolve`. Sie umfasst zahlreiche Zugänge zur Lösung von skalaren gDGL und SysgDGL (vergl. `Help` zu `dsolve`) u.a. mit der Bestimmung einer geschlossenen Form der Lösung, der Lösung eines AWP, der Suche nach Lösungen in Reihenform, der Berechnung numerischer Lösungen oder der Lösung mittels Integraltransformationen.

`dsolve` - solve ordinary differential equations (ODEs)

Calling Sequence

```
dsolve(ODE)
dsolve(ODE,y(x),extra_args)
dsolve({ODE,ICs},y(x),extra_args)
dsolve({sysODE,ICs},{funcs},extra_args)
```

Parameters

```
ODE      - ordinary differential equation
y(x)     - indeterminate function of one variable, required if ODE contains
           derivatives of more than one such function
ICs      - initial conditions
{sysODE} - set with a system of ODEs
{funcs}  - set with indeterminate functions
extra_args - (optional) depends on the type of problem being solved
```

Description

As a general ODE solver, `dsolve` handles different types of ODE problems. These include the following.

- Looking for closed form solutions for a single ODE - see below, or a system of ODEs - see `dsolve,system`.
- Solving ODEs or a system of them with given initial conditions (boundary value problems). See `dsolve,ICs`.
- Looking for formal power series solutions to a linear ODE with polynomial coefficients. See `dsolve,formal_series`.
- Looking for formal solution to a linear ODE with polynomial coefficients. See `dsolve,formal_solution`.
- Looking for solutions using integral transforms (Laplace, Fourier). See `dsolve,integral_transform`.
- Looking for numerical (see `dsolve,numeric`) or series solutions (see `dsolve,series`) to ODEs or systems of them.

In the case of a single ODE, `dsolve` tries to solve it using either classification methods or symmetry methods. Using symmetry methods, `dsolve` first looks for the generators of symmetry groups of the given ODE, and then uses this information to integrate it, or at least reduce its order. Classification methods are used when the ODE matches a recognizable pattern (that is, for which a solving method is already implemented), and symmetry methods are reserved for the non-classifiable cases.

To see what method is being used to solve a given ODE, you can assign the following (see `infolevel`):

```
> infolevel[dsolve] := 3;
```

The ODE Analyzer, a new interactive interface, has been created for numeric and symbolic solutions of ODE and ODE systems. The interface is easy to use, with many common options. In addition to the computation of solutions and solution values, plots can also be generated. The interface is suitable as an educational tool; it can provide the corresponding Maple commands needed to compute the solution or plot. For information on launching the interface (and command-line options), see `dsolve/interactive`. For an illustrated description of the interface, see `worksheet/interactive/dsolve`.

See also `dsolve/details` and

```
dsolve/algorithms, dsolve/education, dsolve/formal_series, dsolve/formal_solution, dsolve/hypergeometric,
dsolve/ICs, dsolve/integrating_factors, dsolve/integrating_factors_for_LODEs, dsolve/interactive,
dsolve/intrans, dsolve/Lie, dsolve/linear, dsolve/numeric, dsolve/piecewise, dsolve/references,
dsolve/series, dsolve/system, int or Int, intat or Intat,
Maple packages for differential equations: DEtools, odeadvisor, odeadvisor/types, PDEtools
```


In den folgenden Beispielen möge der Leser die Ergebnisse stets seiner eigenen Bewertung unterziehen und dabei auch die Kommentare zu den Kommandos beachten.

Beispiel 2.1 Lösung einer gDGL oder eines AWP

(1) $y' = 3x^2$, AB $y(0) = 1$

```
> dgl:=diff(y(x),x)=3*x^2:
dsolve(dgl,y(x));
dsolve({dgl,y(0)=1},y(x));
```

$$y(x) = x^3 + _C1$$

$$y(x) = x^3 + 1$$

(2) $y' = ax + b$, AB $y(0) = 1$

```
> dgl:=diff(y(x),x)=ax+b:
dsolve(dgl,y(x));
dsolve({dgl,y(0)=1},y(x));
```

$$y(x) = \frac{1}{2}ax^2 + bx + _C1$$

$$y(x) = \frac{1}{2}ax^2 + bx + 1$$

(3) $y' = 1 + y/x$, AB $y(1) = 1$ bzw. $y(0) = 1, 0$

```
> dgl:=diff(y(x),x)=1+y(x)/x:
dsolve(dgl,y(x));
dsolve({dgl,y(1)=1},y(x));
dsolve({dgl,y(0)=1},y(x)); # keine Loesung und kein Ergebnis
dsolve({dgl,y(0)=0},y(x)); # _C1 bleibt, x ln(x)->0 fuer x->0
yex:=unapply(rhs(%),x);
limit(yex(x),x=0); # limit(yex(x),x=0,right/left);
yex(0); # Nulldivision
```

$$y(x) = (\ln(x) + _C1)x$$

$$y(x) = (\ln(x) + 1)x$$

$$y(x) = (\ln(x) + _C1)x$$

$$yex := x \rightarrow (\ln(x) + _C1)x$$

$$0$$

Error, (in ln) numeric exception: division by zero

(4) $y' = \ln(x)$, AB $y(0) = 0$ bzw. $y(0) = 2$

```
> dgl:=diff(y(x),x)=1+y(x)/x:
dsolve(dgl,y(x));
dsolve({dgl,y(0)=0},y(x)); # x ln(x)->0 fuer x->0
dsolve({dgl,y(0)=2},y(x));
yex:=unapply(rhs(%),x);
limit(yex(x),x=0); # limit(yex(x),x=0,right/left);
yex(0); # Nulldivision
```

$$y(x) = x \ln(x) - x + _C1$$

$$y(x) = x \ln(x) - x$$

$$y(x) = x \ln(x) - x + 2$$

$$yex := x \rightarrow x \ln(x) - x + 2$$

$$2$$

Error, (in ln) numeric exception: division by zero

(5) $y' \cos(x)^2/4 - y = 0$, AB $y(0) = 0, 1$ bzw. $y(\pi/2) = 0, 1$

```
> dgl:=diff(y(x),x)*cos(x)^2/4-y(x)=0:
dsolve(dgl,y(x));
dsolve({dgl,y(0)=0},y(x));      # 0
dsolve({dgl,y(0)=1},y(x));
dsolve({dgl,y(Pi/2)=0},y(x));   # _C1 bleibt im rechten Loesungsanteil
yex:=unapply(rhs(%),x);
limit(yex(x),x=Pi/2);           # 0, obwohl in Formel _C1 steht
                                # limit(yex(x),x=Pi/2,right/left);
yex(Pi/2);                       # 0

# Kontrolle des rechten Loesungsanteils explizit
limit(exp(4*sin(x)/cos(x)),x=Pi/2,right); # 0
limit(exp(4*sin(x)/cos(x)),x=Pi/2,left);  # infinity
limit(exp(4*sin(x)/cos(x)),x=Pi/2);      # undefined
dsolve({dgl,y(Pi/2)=1},y(x));           # keine Loesung und kein Ergebnis
```

$$y(x) = -C1 e^{\frac{4 \sin(x)}{\cos(x)}}$$

$$y(x) = 0$$

$$y(x) = e^{\frac{4 \sin(x)}{\cos(x)}}$$

$$y(x) = \begin{cases} 0 & x \leq \frac{1}{2}\pi \\ -C1 e^{\frac{4 \sin(x)}{\cos(x)}} & \frac{1}{2}\pi < x \end{cases}$$

$$yex := x \rightarrow \text{piecewise} \left(x \leq \frac{1}{2}\pi, 0, \frac{1}{2}\pi < x, -C1 e^{\frac{4 \sin(x)}{\cos(x)}} \right)$$

$$0$$

$$0$$

$$0$$

$$\infty$$

undefined

(6) $y''' + y' + \sin(x)^2 = 0$, AB $y(0) = 0$, $y'(0) = 0$, $y''(0) = 0$

```
> dgl:=diff(y(x),x$3)+diff(y(x),x$1)+sin(x)^2=0:
dsolve(dgl,y(x));
dsolve({dgl,y(0)=0,D(y)(0)=0,(D@@2)(y)(0)=0},y(x));

y(x) = -cos(x)_C2 + sin(x)_C1 - 1/12 sin(2x) - 1/2 x + _C3
y(x) = 2/3 sin(x) - 1/12 sin(2x) - 1/2 x
```

(7) $y' = 1/(x+1)$, AB $y(0) = 0, 1$, $y(-1) = 0, 1$

```
> dgl:=diff(y(x),x)=1/(x+1):
dsolve(dgl,y(x));
dsolve({dgl,y(0)=0},y(x));
dsolve({dgl,y(0)=1},y(x));
dsolve({dgl,y(-1)=0},y(x));   # keine Loesung und kein Ergebnis
dsolve({dgl,y(-1)=1},y(x));   # keine Loesung und kein Ergebnis
```

$$y(x) = \ln(1+x) + -C1$$

$$y(x) = \ln(1+x)$$

$$y(x) = \ln(1+x) + 1$$

(8) $y' = e^x + 28x^3 + e$, AB $y(0) = 0, 1$

```
> dgl:=diff(y(x),x)=exp(x)+28*x^3+exp(1):
dsolve(dgl,y(x));
dsolve({dgl,y(0)=0},y(x));
dsolve({dgl,y(0)=1},y(x));
```

$$y(x) = e^x + 7x^4 + ex + _CI$$

$$y(x) = e^x + 7x^4 + ex - 1$$

$$y(x) = e^x + 7x^4 + ex$$

(9) $y' = py(m - y)$, AB $y(0) = 0, -1, 1/2, 1$, logistische Gleichung

```
> p:='p': m:='m':
dgl:=diff(y(x),x)=p*y(x)*(m-y(x)):
dsolve(dgl,y(x)); eval(rhs(%),[m=1,p=2]);
dsolve({dgl,y(0)=0},y(x)); # 0

dsolve({dgl,y(0)=-1},y(x)); eval(rhs(%),[m=1,p=2]);
dsolve({dgl,y(0)=1/2},y(x)); eval(rhs(%),[m=1,p=2]);
dsolve({dgl,y(0)=1},y(x)); eval(rhs(%),[m=1,p=2]); # 1
```

$$y(x) = \frac{m}{1+e^{-m p x} _CI m}$$

$$\frac{1}{1+e^{-2x} _CI}$$

$$y(x) = 0$$

$$y(x) = -\frac{m}{-1+e^{-m p x} + e^{-m p x} m}$$

$$-\frac{1}{-1+2e^{-2x}}$$

$$y(x) = \frac{m}{1+2e^{-m p x} m - e^{-m p x}}$$

$$\frac{1}{1+e^{-2x}}$$

$$y(x) = \frac{m}{1+e^{-m p x} m - e^{-m p x}}$$

$$1$$

(10) $y' = axy$, AB $y(0) = 0, 1, -1$

```
> a:='a':
dgl:=diff(y(x),x)=a*x*y(x);
dsolve(dgl,y(x)); eval(rhs(%),[a=1]);
dsolve({dgl,y(0)=0},y(x)); # 0
```

```
dsolve({dgl,y(0)=1},y(x)); eval(rhs(%),[a=1]);
```

```
dsolve({dgl,y(0)=-1},y(x)); eval(rhs(%),[a=1]);
```

$$y(x) = _CI e^{\frac{1}{2} a x^2}$$

$$-_CI e^{\frac{1}{2} x^2}$$

$$y(x) = 0$$

$$y(x) = e^{\frac{1}{2} a x^2}$$

$$e^{\frac{1}{2} x^2}$$

$$y(x) = -e^{\frac{1}{2} a x^2}$$

$$-e^{\frac{1}{2} x^2}$$

$$(11) \quad y' = ay + b, \quad \text{AB } y(0) = 0, 1, -1$$

```
> a:='a': b:='b':
    dgl:=diff(y(x),x)=a*y(x)+b:
    dsolve(dgl,y(x));          eval(rhs(%),[a=1,b=0]);
    dsolve({dgl,y(0)=0},y(x));
    dsolve({dgl,y(0)=1},y(x)); eval(rhs(%),[a=1,b=1]);
    dsolve({dgl,y(0)=-1},y(x)); eval(rhs(%),[a=1,b=1]); # -1
    y(x) = - $\frac{b}{a} + e^{ax}$ _C1
    ex_C1
    y(x) = - $\frac{b}{a} + \frac{e^{ax}b}{a}$ 
    y(x) = - $\frac{b}{a} + e^{ax}(1 + \frac{b}{a})$ 
    -1 + 2ex
    y(x) = - $\frac{b}{a} + e^{ax}(-1 + \frac{b}{a})$ 
    -1
```

Beispiel 2.2 Lösung eines SysgDgl 1. Ordnung mit zwei gDgl bzw. AWP

$$(1) \quad \begin{aligned} y' &= z, & \text{AB } y(0) &= 1 \\ z' &= y, & z(0) &= 1 \end{aligned}$$

```
> dgl1:=diff(y(x),x)=z(x):
    dgl2:=diff(z(x),x)=y(x):
    dsolve({dgl1,dgl2},{y(x),z(x)});
    dsolve({dgl1,dgl2,y(0)=1,z(0)=1},{y(x),z(x)});
    {y(x) = _C1ex + _C2e-x, z(x) = _C1ex - _C2e-x}
    {y(x) = ex, z(x) = ex}
```

$$(2) \quad \begin{aligned} y' &= z, & \text{AB } y(0) &= 0, 1 \\ z' &= y, & z(0) &= 0, 1 \end{aligned}$$

```
> dgl1:=diff(y(x),x)=z(x):
    dgl2:=diff(z(x),x)=y(x):
    dsolve({dgl1,dgl2},{y(x),z(x)});
    dsolve({dgl1,dgl2,y(0)=0,z(0)=0},{y(x),z(x)}); # 0, 0
    dsolve({dgl1,dgl2,y(0)=1,z(0)=0},{y(x),z(x)}); # ohne Loesung/Ergebnis
    dsolve({dgl1,dgl2,y(0)=0,z(0)=1},{y(x),z(x)}); # "
    dsolve({dgl1,dgl2,y(0)=1,z(0)=1},{y(x),z(x)}); # "
    [{z(x) = 2 WeierstrassP(x + _C1, 0, _C2)}, {y(x) =  $\frac{1}{3} \frac{d}{dx} z(x)$ }]
    {y(x) = 0, z(x) = 0}
```

$$(3) \quad \begin{aligned} y' &= xy/2, & \text{AB } y(1) &= 1, \text{ verschiedene } x_0 \text{ in der AB, separierte gDgl} \\ z' &= 2z, & z(0) &= 1 \end{aligned}$$

```
> dgl1:=diff(y(x),x)=x*y(x)/2:
    dgl2:=diff(z(x),x)=2*z(x):
    dsolve({dgl1,dgl2},{y(x),z(x)});
    dsolve({dgl1,dgl2,y(1)=1,z(0)=1},{y(x),z(x)});
    {y(x) = _C2e $\frac{1}{4}x^2$ , z(x) = _C1e2x}
    {y(x) = e $\frac{1}{4}(x^2-1)$ , z(x) = e2x}
```

$$(4) \quad y' = xy/2 + z, \quad \text{AB} \quad y(0) = 0, 1 \\ z' = 2z \quad \quad \quad z(0) = 0, 1$$

```
> dgl1:=diff(y(x),x)=x*y(x)/2+z(x):
dgl2:=diff(z(x),x)=2*z(x):
dsolve({dgl1,dgl2},{y(x),z(x)});
dsolve({dgl1,dgl2,y(1)=0,z(0)=0},{y(x),z(x)});
dsolve({dgl1,dgl2,y(1)=1,z(0)=0},{y(x),z(x)});
dsolve({dgl1,dgl2,y(1)=0,z(0)=1},{y(x),z(x)});
{y(x) = -C2*sqrt(pi)*erf(1/2*x - 2)*e^(1/4*x^2+4) + e^(1/4*x^2+4)*_C1, z(x) = -C2*e^(2*x)}
{y(x) = 0, z(x) = 0}
{y(x) = e^(1/4*x^2), z(x) = 0}
{y(x) = sqrt(pi)*erf(1/2*x - 2)*e^(1/4*x^2+4) + e^(1/4*x^2)*sqrt(pi)*erf(2)*e^4, z(x) = e^(2*x)}
```

$$(5) \quad y' = 2z, \quad \text{AB} \quad y(0) = 1, 2 \\ z' = y + z \quad \quad \quad z(0) = 1$$

```
> dgl1:=diff(y(x),x)=2*z(x):
dgl2:=diff(z(x),x)=y(x)+z(x):
dsolve({dgl1,dgl2},{y(x),z(x)});
dsolve({dgl1,dgl2,y(1)=1,z(0)=1},{y(x),z(x)});
dsolve({dgl1,dgl2,y(1)=2,z(0)=1},{y(x),z(x)});
{y(x) = -C1*e^(2*x) + -C2*e^(-x), z(x) = -C1*e^(2*x) - 1/2*_C2*e^(-x)}
{y(x) = e^(2*x), z(x) = e^(2*x)}
{y(x) = 4/3*e^(2*x) + 2/3*e^(-x), z(x) = 4/3*e^(2*x) - 1/3*e^(-x)}
```

$$(6) \quad y' = -z, \quad \text{AB} \quad y(0) = 0, 1 \\ z' = y \quad \quad \quad z(0) = 0, 1$$

```
> dgl1:=diff(y(x),x)=-z(x):
dgl2:=diff(z(x),x)=y(x):
dsolve({dgl1,dgl2},{y(x),z(x)});
dsolve({dgl1,dgl2,y(0)=0,z(0)=0},{y(x),z(x)});
dsolve({dgl1,dgl2,y(0)=1,z(0)=1},{y(x),z(x)});
{y(x) = -C1*sin(x) + -C2*cos(x), z(x) = -_C1*cos(x) + -C2*sin(x)}
{y(x) = 0, z(x) = 0}
{y(x) = -sin(x) + cos(x), z(x) = cos(x) + sin(x)}
```

$$(7) \quad y' = z + x, \quad \text{AB} \quad y(0) = 0, 1 \\ z' = y - x \quad \quad \quad z(0) = 0, 1$$

```
> dgl1:=diff(y(x),x)=z(x)+x:
dgl2:=diff(z(x),x)=y(x)-x:
dsolve({dgl1,dgl2},{y(x),z(x)});
dsolve({dgl1,dgl2,y(0)=0,z(0)=0},{y(x),z(x)});
dsolve({dgl1,dgl2,y(0)=1,z(0)=1},{y(x),z(x)});
{y(x) = e^x*_C2 + e^(-x)*_C1 - 1 + x, z(x) = e^x*_C2 - e^(-x)*_C1 + 1 - x}
{y(x) = -1 + e^(-x) + x, z(x) = 1 - e^(-x) - x}
{y(x) = e^x + e^(-x) - 1 + x, z(x) = e^x - e^(-x) + 1 - x}
```

(8) Zweispezimodell

$$y' = ay(1 - bz), \quad \text{AB } y(0) = y_0, 0, 1, \text{ Parameter geg.}$$

$$z' = -cz(1 - dy), \quad z(0) = z_0, 0, 1$$

```
> dgl1:=diff(y(x),x)=-4*y(x)-z(x):
dgl2:=diff(z(x),x)=y(x)+2*z(x):
dsolve({dgl1,dgl2},{y(x),z(x)});
dsolve({dgl1,dgl2,y(0)=y0,z(0)=z0},{y(x),z(x)});
dsolve({dgl1,dgl2,y(0)=0,z(0)=0},{y(x),z(x)});
dsolve({dgl1,dgl2,y(0)=1,z(0)=1},{y(x),z(x)});
```

$$\{y(x) = -C_1 e^{-(1+2\sqrt{2})x} + C_2 e^{-(1+2\sqrt{2})x},$$

$$z(x) = -3C_1 e^{-(1+2\sqrt{2})x} - 2C_1 e^{-(1+2\sqrt{2})x} \sqrt{2} - 3C_2 e^{-(1+2\sqrt{2})x} + 2C_2 e^{-(1+2\sqrt{2})x} \sqrt{2}\}$$

$$\{y(x) = -\frac{1}{8}\sqrt{2}(-2\sqrt{2}y_0 + 3y_0 + z_0)e^{-(1+2\sqrt{2})x} + \frac{1}{8}\sqrt{2}(2\sqrt{2}y_0 + 3y_0 + z_0)e^{-(1+2\sqrt{2})x},$$

$$z(x) = \frac{3}{8}\sqrt{2}(-2\sqrt{2}y_0 + 3y_0 + z_0)e^{-(1+2\sqrt{2})x} + \frac{1}{2}(-2\sqrt{2}y_0 + 3y_0 + z_0)e^{-(1+2\sqrt{2})x}$$

$$- \frac{3}{8}\sqrt{2}(2\sqrt{2}y_0 + 3y_0 + z_0)e^{-(1+2\sqrt{2})x} + \frac{1}{2}(2\sqrt{2}y_0 + 3y_0 + z_0)e^{-(1+2\sqrt{2})x}\}$$

$$\{y(x) = 0, z(x) = 0\}$$

$$\{y(x) = (\frac{1}{2} - \frac{1}{2}\sqrt{2})e^{-(1+2\sqrt{2})x} + (\frac{1}{2} + \frac{1}{2}\sqrt{2})e^{-(1+2\sqrt{2})x},$$

$$z(x) = -3(\frac{1}{2} - \frac{1}{2}\sqrt{2})e^{-(1+2\sqrt{2})x} - 2(\frac{1}{2} - \frac{1}{2}\sqrt{2})e^{-(1+2\sqrt{2})x} \sqrt{2} - 3(\frac{1}{2} + \frac{1}{2}\sqrt{2})e^{-(1+2\sqrt{2})x} + 2(\frac{1}{2} + \frac{1}{2}\sqrt{2})e^{-(1+2\sqrt{2})x} \sqrt{2}\}$$

Beispiel 2.3 Bestimmung des Fundamentalsystems und evtl. der partikulären Lösung(1) $y'''' + ay'' = 0$, a Parameter

```
> a:='a':
dgl:=diff(y(x),x$4)+a*diff(y(x),x$2)=0:
dsolve({dgl},y(x),output=basis);
> dgl:=diff(y(x),x$4)+0*diff(y(x),x$2)=0:
dsolve({dgl},y(x),output=basis);
```

$$[1, x, \sin(\sqrt{a}x), \cos(\sqrt{a}x)]$$

$$[1, x, x^2, x^3]$$
(2) $5y'''' + y'' + 2 = 0$

```
> dgl:=diff(5*y(x),x$4)+diff(y(x),x$2)+2=0:
dsolve({dgl},y(x),output=basis);
```

$$[[1, x, \sin(\frac{1}{5}\sqrt{5}x), \cos(\frac{1}{5}\sqrt{5}x)], -x^2]$$
(3) $5y'' + y'/2 + y = 0$

```
> dgl:=diff(y(x),x$2)+1/2*diff(y(x),x)+y(x)=0:
dsolve({dgl},y(x),output=basis);
```

$$[e^{-\frac{1}{4}x} \cos(\frac{1}{4}\sqrt{15}x), e^{-\frac{1}{4}x} \sin(\frac{1}{4}\sqrt{15}x)]$$
(4) $y'''(x^2y)' = 0$

```
> dgl:=diff(y(x),x$3)*diff(x^2*y(x),x)=0:
dsolve({dgl},y(x),output=basis);
```

$$[1, x, x^2]$$

(5) $y''' + y' + \sin(x)^2 = 0$

```
> dgl:=diff(y(x),x$3)+diff(y(x),x$1)+sin(x)^2=0:
dsolve({dgl},y(x),output=basis);
```

$$[[1, \sin(x), \cos(x)], -\frac{1}{12} \sin(2x) - \frac{1}{2}x]$$

(6) $y' - x^2 = 0$, $y'' - y' + x - 1 = 0$, $y''' + 9y'' = 0$

```
> dgl:=diff(y(x),x)-x^2=0:
dsolve({dgl},y(x),output=basis);
> dgl:=diff(y(x),x$2)-diff(y(x),x)+x-1=0:
dsolve({dgl},y(x),output=basis);
> dgl:=diff(y(x),x$3)+9*diff(y(x),x$2)=0:
dsolve({dgl},y(x),output=basis);
```

$$[[1], \frac{1}{3}x^3]$$

$$[[1, e^x], \frac{1}{2}x^2]$$

$$[1, x, e^{-9x}]$$

(7) $y'' + ay' = 0$, $y'' + ay' + by = 0$, $y'' - \omega^2 y = 0$, $y'' - 4y' + 4y = 0$

```
> a:='a': b:='b':
dgl:=diff(y(x),x$2)+a*diff(y(x),x)=0:
dsolve({dgl},y(x),output=basis);
> dgl:=diff(y(x),x$2)+a*diff(y(x),x)+b*y(x)=0:
dsolve({dgl},y(x),output=basis);
> omega:='omega':
dgl:=diff(y(x),x$2)-omega^2*y(x)=0:
dsolve({dgl},y(x),output=basis);
> dgl:=diff(y(x),x$2)-4*diff(y(x),x)+4*y(x)=0:
dsolve({dgl},y(x),output=basis);
```

$$[1, e^{-ax}]$$

$$[e^{(-\frac{1}{2}a + \frac{1}{2}\sqrt{a^2-4b})x}, e^{(-\frac{1}{2}a - \frac{1}{2}\sqrt{a^2-4b})x}]$$

$$[e^{\omega x}, e^{-\omega x}]$$

$$[e^{2x}, e^{2x}x]$$

(8) $y'''''' + y'''' = 0$, $y'''' + y''' = 27$, $y'''' + 4y''' + 2y'' - 4y' + 8y + 16y = 0$

```
> dgl:=diff(y(x),x$6)+diff(y(x),x$5)=0:
dsolve({dgl},y(x),output=basis);
> dgl:=diff(y(x),x$5)+diff(y(x),x$3)=27:
dsolve({dgl},y(x),output=basis);
> dgl:=diff(y(x),x$5)+4*diff(y(x),x$4)+2*diff(y(x),x$3)-
4*diff(y(x),x$2)+8*diff(y(x),x)+16*y(x)=0:
dsolve({dgl},y(x),output=basis);
```

$$[1, x, x^2, x^3, x^4, e^{-x}]$$

$$[[1, x, x^2, \sin(x), \cos(x)], \frac{9}{2}x^3]$$

$$[e^{-2x}, e^{-2x}x, e^{-2x}x^2, e^x \sin(x), e^x \cos(x)]$$

2.2 Lösungsdarstellung von Differentialgleichungen

Ausgehend vom Ergebnis der Maple-Funktion `dsolve`, welche die analytische Lösung der gDGL bzw. des AWP liefert, kann man zunächst überprüfen, ob die Funktion wirklich der gDGL genügt und/oder die AB erfüllt.

Dazu notiert man am besten die gDGL als $F(x, y, y', \dots) = 0$, setzt die Lösung auf der linken Seite der gDGL ein (Kommando `subs`), um den Vergleich mit Null zu machen. Eventuell führen erst Vereinfachungen der Formel mit `eval`, `simplify`, `expand` zum Ziel.

Analog testet man die Erfüllung der AB, indem man die Lösung an dieser Stelle mit dem `subs`-Befehl auswertet und wenn notwendig noch vereinfacht.

Gegeben sei als Testbeispiel das AWP (siehe auch Abschnitt 1.8.1 Nr. 68)

$$y' - y + x^3 - 3x + 2 = 0, \quad y(0) = 1. \quad (2.1)$$

Die sichere Vorgehensweise ist die folgende.

```
> dgl:=diff(y(x),x)-y(x)+x^3-3*x+2:
dsolve({dgl=0},y(x));      # dsolve([dgl=0],y(x));
dsolve(dgl=0,y(x));
f:=rhs(%);
# Kontrolle der Richtigkeit der Loesung y(x) durch Einsetzen in die gDGL
dgl_ein:=simplify(subs(y(x)=f,dgl));
eval(dgl_ein);             # expand(dgl_ein); simplify(dgl_ein);
```

$$\{y(x) = 5 + 3x + 3x^2 + x^3 + e^x_C1\}$$

$$y(x) = 5 + 3x + 3x^2 + x^3 + e^x_C1$$

$$f := 5 + 3x + 3x^2 + x^3 + e^x_C1\}$$

$$dgl_ein := \frac{\partial}{\partial x}(5 + 3x + 3x^2 + x^3 + e^x_C1) - 3 - 6x - 3x^2 - e^x_C1$$

$$0$$

```
> dgl_ab:=dsolve({dgl=0,y(0)=1},y(x)):
f1:=rhs(dgl_ab);
# Kontrolle der Richtigkeit der Loesung y(x) durch Einsetzen
subs(y(x)=f1,dgl);
eval(%);
# Kontrolle der Richtigkeit der AB y(0)=1 durch Einsetzen
subs(x=0,f1);
simplify(%);
```

$$f1 := 5 + 3x + 3x^2 + x^3 - 4e^x$$

$$\frac{d}{dx}(5 + 3x + 3x^2 + x^3 - 4e^x) - 3 - 6x - 3x^2 + 4e^x_C1$$

$$0$$

$$5 - 4e^0$$

$$1$$

```
> # Andere AB fuer weitere Trajektorien
f2:=rhs(dsolve({dgl=0,y(0)=1/2},y(x))):
f3:=rhs(dsolve({dgl=0,y(0)=3/2},y(x))):
```


Um gDGL und ihre Lösungen besser zu charakterisieren, kann man evtl. das Richtungsfeld der gDGL verwenden.

Dazu werden die Funktionen `dfield` und `quiver` in MATLAB sowie die Maple-Befehle `fieldplot`, `dfieldplot`, `DEplot` genutzt (auch im Abschnitt 4.1 kurz erläutert).

Zu empfehlen sind die Kommandos `dfieldplot` und `DEplot`. `fieldplot` dient allgemein der Darstellung von Vektorfeldern und kann möglicherweise wegen fehlender Skalierung das Richtungsfeld nicht so elegant anzeigen. Der Befehl `DEplot` kann neben dem Richtungsfeld auch numerische Lösungen der gDGL erzeugen und ausgeben. Weitere Kommandos können diese Daten noch verarbeiten.

In jedem Fall sollte man auf die Wahl des Darstellungsbereiches und die zahlreichen Optionen achten, um anschauliche und überzeugende Ergebnisse/Grafiken zu präsentieren.

2.2.1 Richtungsfeld der Differentialgleichung

`plots[fieldplot]` - plot a 2-D vector field

Calling Sequences

```
fieldplot(f,x=r1,y=r2,options)
fieldplot(f,r1,r2,options)
```

Parameters

```
f      - Vector field to be plotted
x      - first coordinate
r1     - range of first coordinate
y      - second coordinate
r2     - range of second coordinate
options - options that control the behavior of fieldplot
```

Description

- . A typical call to the `fieldplot` function is `fieldplot(f,x=a..b,y=c..d)`, where `f` is a list of two expressions in `x` and `y`. `a..b` and `c..d` specifies the horizontal and vertical ranges of the field. The result is a two dimensional vector field with the vector evaluated at `(x,y)` located at this point.
- . It is possible to input procedures, rather than expressions, as the components of a vector field. A typical call in this case is `fieldplot(f,a..b,c..d)`, where `f` is a list of two procedures in two variables.
- . It is also possible to input a 2 dimensional `VectorField` from the `VectorCalculus` package to be plotted, which works with either calling sequence as the coordinates are given as part of the field vector.
- . A call to `fieldplot` produces a PLOT data structure, which is then printed. For information on these data structures, see `plot[structure]`. Remaining arguments are interpreted as options which are specified as equations of the form `option = value`. In particular, the following optional arguments are available.
- . The `arrows` option allows you to specify the style of arrow used to represent a vector. The possible values are `LINE`, `THIN`, `SLIM`, and `THICK`, with `THIN` being the default.

- . The `grid` option allows you to set the size of the two dimensional grid use to plot the field vectors. The default is `[20,20]`.

- . The `fieldstrength` option allows finer control of how the size of the drawn vectors corresponds to the strength of the field. The following options are available:

`maximal` or `maximal(v)`: The size of the arrow corresponds to the ratio of the field strength at the given point to the maximal field strength at all points in the grid. The value `v` (by default 1) is a scaling factor used to adjust the ratio. Values smaller than 1 reduce the size of all arrows, while values larger than 1 increase the size of all arrows. Note: In cases where $1 < v$, the drawing of arrows that are larger than the maximum size is controlled through the `clipping` option (see below).

`average` or `average(v)`: The size of the arrow corresponds to the ratio of the field strength at the given point to the average field strength at all points in the grid. The value `v` (by default is 1/2) multiplies the computed average to determine the maximum arrow size. The drawing of arrows that are larger than the maximum size is controlled through the `clipping` option (see below). This option is most useful for fields that have large regions with small values, as it allows scaling up to view the field direction for those regions.

`fixed` or `fixed(v)`: The size of all arrows is fixed to `v` (by default 0.9) times the maximum arrow size for the grid. The value `v` must be a positive value no larger than 1. Use of this option results in a direction plot for the field.

`log` or `log[v]`: The size of the arrows is rescaled according to the formula $\text{size} = \exp(\ln(\text{strength}/\text{max})/(1+v))$ where `strength` is the strength of the field at the point, and `max` is the maximum strength of the field on the grid. The value `v` (by default 2) must be larger than -1. Values of `v` larger than zero have the effect of reducing the variation of the arrow length throughout the field, while values smaller than zero increase the variation. Using the value `v = 0` produces an identical plot to the default `maximal` plot. This option is most useful for fields that have either very large or very small variations, for example, an inverse square law (see the examples below).

`scaleto(v)`: The size of the arrows is scaled so that the maximum `fieldstrength` is assumed to be `v` (which must be a positive numeric value). This option is most useful when constructing an animation of `fieldplots` where the maximal field strength is not consistent between frames of the animation, but the relative field strength between the frames is relevant.

- . The `clipping` boolean option specifies how to draw arrows that have too large a magnitude. If set to `true`, arrows with too large a magnitude are drawn at a fixed maximum size without an arrow head. If set to `false`, arrows size is not bounded, so the arrows are drawn as dictated by their magnitude, even when these arrows would obscure other arrows in the plot. By default, this option is `false` if a coordinate transform is applied (via the `coords` argument), and `true` otherwise. This option is only useful if the `fieldstrength` option is used to increase the size of the arrows, or the field has been transformed. These are the only cases in which arrows may exceed the desired length.

- . Use with `Coordinate Transforms`
Some discussion of the behavior of `fieldplot` when used in combination with the `coords` options is required.

Under a coordinate transform, both the field and the location of the arrows are adjusted. Options like view and axes labels apply only to the plot after the transform has been applied. A significant issue arises when scaling the arrows that describe the vector field after the transform has been applied. Specifically, the transform may map many points on the grid to nearly the same area in the transformed region.

The fieldplot command uses an averaging technique to determine a suitable rescaling of the field vectors after the transform. In some cases, this may not be what is desired, so the fieldstrength and clipping options can be used to adjust the plot. In addition, arrows that are located at transform singularities are not drawn if they have a component in a direction that becomes undefined at the singularity.

As an example, a purely radial field will always have arrows drawn in polar coordinates, but the field $[0,1]$ will have no arrows drawn at $r=0$.

- . The remaining options are the same as those available for the plot command. See plot[options] for further information.
- . The command with(plots,fieldplot) allows the use of the abbreviated form of this command.

Aus der gDGl 1. Ordnung (2.1) kann das Vektorfeld $[1, y'] = [1, y - x^3 + 3x - 2]$ erzeugt werden. Man bemerkt, dass bei entsprechender Gebietsauswahl eine anschauliche Darstellung des Richtungsfeldes, d.h. der Pfeile, erfolgt.

```
> v:=[1,y-x^3+3*x-2];
  fieldplot(v,x=-2..2,y=-2..5); # Farbe black default
> fieldplot(v,x=0..1,y=0..1,color=black,arrows=medium);
```

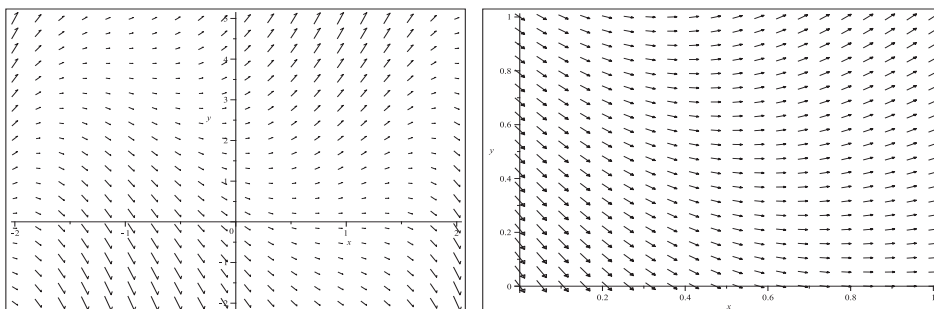


Abb. 2.1 Dateien *richt_field_01.ps*, *richt_field_02.ps*,

Richtungsfeld zur gDGl $y' - y + x^3 - 3x + 2 = 0$ in $[-2, 2] \times [-2, 5]$
bzw. $[0, 1] \times [0, 1]$, Kommando fieldplot

Richtungsfeld $[1, y']$ gemeinsam mit Lösungen $y(x)$ des AWP

```
> v:=[1,y-x^3+3*x-2]:
  p1:=fieldplot(v,x=-2..2,y=-2..5,color=black):
  p2:=plot([f1,f2,f3],x=-2..2,thickness=2,color=[black,green,blue]):
  p3:=pointplot([[0,1],[0,1/2],[0,3/2]],color=[black,green,blue],
                symbol=solidcircle,symbolsize=16):
  display(p1,p2,p3,axes=frame);
```

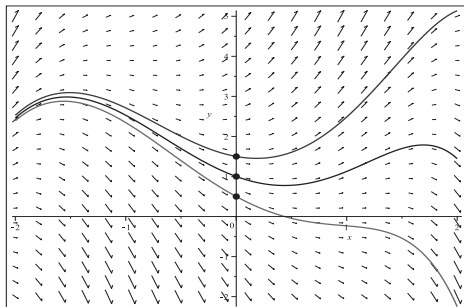


Abb. 2.2

Datei *richt_field_03.ps*,

Richtungsfeld zum AWP (2.1)

in $[-2, 2] \times [-2, 5]$

mit den exakten Lösungen

zu den AB $y(0) = \frac{1}{2}, 1, \frac{3}{2}$,Kommandos `fieldplot`, `dsolve`, `plot`

DEtools[dfieldplot] - plot direction field to a system of DEs

Calling Sequences

```
dfieldplot(deqns, vars, trange, yrange, xrange, options)
```

Parameters

```
deqns - list or set of first order ordinary differential equations
vars   - list or set of dependent variables
trange - range of the independent variable
xrange - range of the first dependent variable
yrange - range of the second dependent variable
options - (optional) equations of the form keyword=value
```

Description

- . Given either a system of two first order autonomous differential equations, or a single first order differential equation, `dfieldplot` produces a direction field plot. There can be only one independent variable.
- . For plotting solution curves, see `DEplot` or `phaseportrait`.
- . The direction field presented consists of a grid of arrows tangential to solution curves. For each grid point, the arrow centered at (x, y) will have slope dy/dx . For system of two first order autonomous differential equations this slope is computed using $(dy/dt)/(dx/dt)$, where these two derivatives are specified in the first argument to `dfieldplot`.
Note: A system is determined to be autonomous when all terms and factors, other than the differential, are free of the independent variable. For more information, see `DEtools[autonomous]`. For the single first order differential equation, the slope is simply dy/dx .
- . Optional equations for `dfieldplot` are in the `DEplot` help page.
- . The `xrange` and `yrange` parameters must be specified as follows.
`x(t)=x1..x2, y(t)=y1..y2` or `x=x1..x2, y=y1..y2`.
By default, integration along a solution curve stops one mesh point after the specified range is exceeded.
This may be overridden by the `obsrange` option.

Hinweise zu den optionalen Parametern `options` findet man in `DEplot`.

Auch hier ist die Wahl des entsprechenden Gebiets sowie der verschiedenen Optionen für das Richtungsfeld wichtig.

```
> dfieldplot(dgl=0,y,x=-2..2,y=-2..5); # Farbe red
pl1:=dfieldplot(dgl,y,x=-2..2,y=-2..5,color=black):
display(pl1);
pl2:=dfieldplot(dgl,y,x=-2..2,y=-2..5,color=black,
arrows=large,dirgrid=[10,10]):
display(pl2);
```

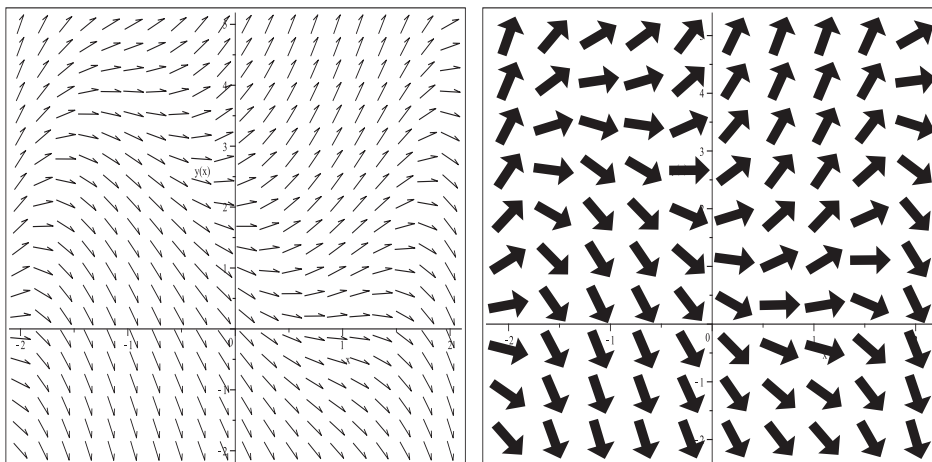


Abb. 2.3 Dateien *richt_field_04.ps*, *richt_field_05.ps*,
Richtungsfeld zur gDGL $y' - y + x^3 - 3x + 2 = 0$ in $[-2, 2] \times [-2, 5]$,
Kommando `dfieldplot`

DEtools[DEplot] - plot solutions to a system of DEs

Calling Sequences

```
DEplot(deqns,vars,trange,options)
DEplot(deqns,vars,trange,inits,options)
DEplot(deqns,vars,trange,xrange,yrange,options)
DEplot(deqns,vars,trange,inits,xrange,yrange,options)
DEplot(dproc,vars,trange,number,xrange,yrange,options)
```

Parameters

`deqns` - list or set of first order ordinary differential equations,
or a single differential equation of any order

`dproc` - a Maple procedure representation for first order ordinary
differential equations, or a single differential equation
of any order

`vars` - dependent variable, or list or set of dependent variables

`number` - equation of the form 'number'=integer indicating the number
of differential equations when `deqns` is given as a function
(`dproc`) instead of expressions

`inits` - set or list of lists; initial conditions for solution curves

`trange` - range of the independent variable

`xrange` - range of the first dependent variable

`yrange` - range of the second dependent variable

`options` - (optional) equations of the form `keyword=value`

Das Kommando `DEplot` dient i. Allg. zur Darstellung des Richtungsfeldes der gDGL mit numerischen Lösungen von AWP. Die Beschreibung des Kommandos erfolgt im Abschnitt 2.2.3, wenn es auch um numerische Lösungen geht.

Werden keine AB notiert, so wird nur das Richtungsfeld gezeigt, vorausgesetzt, dass der vertikale Bereich (Ausdehnung) `xrange` angegeben ist.

Die Vielzahl der Optionen gestattet eine geeignete Auswahl, sodass auch bei komplizierten gDGL anschauliche Ergebnisse erzeugt werden können.

```
> # wenn keine AB, dann muss der y-Bereich angegeben sein
p11a:=DEplot({dgl=0}, [y(x)], x=-2..2, axes=frame): # Fehler
```

Error, (in DEtools/DEplot) all dependent variable ranges must be specified.

```
> DEplot(dgl=0, y(x), x=-2..2, y=-2..5, axes=frame): # Farbe red
```

```
> p11:=DEplot(dgl, y(x), x=-2..2, y=-2..5, axes=frame, color=black,
  arrows=mediumfill): # default dirgrid=[20,20]
display(p11);
p12:=DEplot({dgl=0}, [y(x)], x=-2..2, y=-2..5, axes=frame, color=black,
  arrows=comet, dirgrid=[15,20]):
display(p12);
```

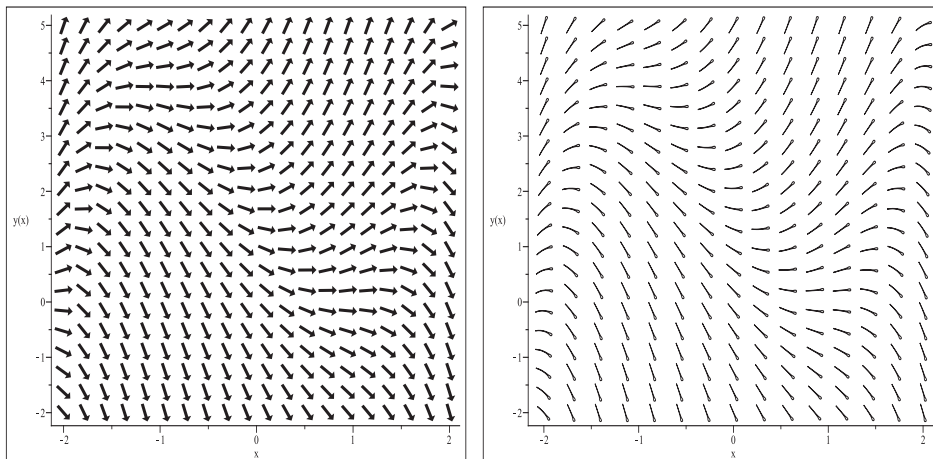


Abb. 2.4 Dateien *richt_field_06.ps*, *richt_field_07.ps*,

Richtungsfeld zur gDGL $y' - y + x^3 - 3x + 2 = 0$ in $[-2, 2] \times [-2, 5]$,

Kommando `DEplot`

2.2.2 Numerische Lösung

Neben dem Richtungsfeld der gDGl sind natürlich auch numerische Lösungen des AWP von Interesse. In Maple sind dazu die Kommandos `dsolve`, `DEplot`, `phaseportrait` vorhanden.

Mit `dsolve` kann die gDGl numerisch gelöst werden und geeignet weiterverarbeitet werden. Eine Kombination mit Befehlen für die Erzeugung des Richtungsfeldes ist natürlich möglich. Die Ergebnisse können auch grafisch verarbeitet werden, z.B. mit `odeplot`.

In jedem Fall sollte man auch hier auf die Wahl des Darstellungsbereiches und die zahlreichen Optionen achten, um aussagekräftige Resultate zu erhalten.

`dsolve/numeric` - find numerical solution of ODEs

Calling Sequences

```
dsolve(odesys,numeric,vars,options)
dsolve(numeric,procopts,options)
```

Parameters

```
odesys - set or list; ordinary differential or differential-algebraic
         equation(s) and initial/boundary conditions
numeric - name; instruct dsolve to find a numerical solution
vars     - (optional) any indeterminate function of one variable, or a set
         or list of them, representing the unknowns of the ODE problem
procopts - (required if odesys is not present) options that specify a
         procedure defined system (procedure, initial, start, number,
         and procvars). For more information, see dsolve[numeric,IVP].
options  - (optional) equations of the form keyword = value
```

Description

- . The `dsolve` command with the `numeric` or `type=numeric` option finds a numerical solution for the ODE or ODE system. If the optional `equation method=numericmethod` is provided (where `numericmethod` is one of `rkf45`, `ck45`, `rosenbrock`, `bvp`, `rkf45_dae`, `ck45_dae`, `rosenbrock_dae`, `dverk78`, `lsode`, `gear`, `taylorseries`, `mebdfi`, or `classical`), `dsolve` uses that method to obtain the numerical solution.
Both initial and boundary value problems can be numerically solved, as well as initial differential algebraic problems. For a system input, the type of problem is automatically detected, but for some DAE problems it may be necessary to specify the method.
By default, a procedure is returned that can be used to obtain solution values if given the value of the independent variable.
- . The default for initial value problems (IVP) is a Runge-Kutta Fehlberg method that produces a fifth order accurate solution (For more information, see `rkf45` or `numeric,IVP`).
For boundary value problems (BVP), a finite difference technique with Richardson extrapolation is used (For more information, see `numeric,BVP`).
For differential-algebraic IVP problems (DAE), a modification of the `rkf45` method is used (For more information, see `dae_extension`).
- . All IVP methods can be used for complex-valued IVPs with a real-valued independent variable (though for the default stiff and nonstiff IVP methods, it may be necessary to specify that the problem is complex via the `complex` option). None of the BVP or DAE methods can currently be used for complex-valued problems, requiring the system be converted to a real system before calling `dsolve`.

- . The IVP and DAE methods have additional capabilities for the returned solution procedure. Specifically, it is possible to query the last computed solution value (useful for problems with singularities), query the initial data, and change the initial data for some solution procedures (DAE methods support a subset of this). For more information, see `numeric,IVP` and `numeric,DAE`. When the set of dependent variables is specified as a list, then this specifies the order of the dependent variables as they appear in the output. The order is otherwise alphabetical in the dependent variable name.

Options

- . The high-level options, most common to IVPs, BVPs, and DAEs are as follows.

'output'	= keyword or array
'stiff'	= boolean
'events'	= list
'event_pre'	= keyword
'event_maxiter'	= integer
'range'	= numeric..numeric
'abserr'	= numeric or list
'relerr'	= numeric
'known'	= name or list of names
'optimize'	= boolean

'output' = keyword or array

Keyword that can take the values `procedurelist`, `listprocedure`, `operator`, or `piecewise`, or an array or `Array` that gives the values of the independent variable at which solution values are desired.

The default keyword is `procedurelist`, which gives the output from `dsolve` as a procedure. This procedure accepts the value of the independent variable as an argument, and it returns a list of the solution values of the form `variable=value`, where the left-hand sides are the names of the independent variable, the dependent variable(s) and their derivatives (for higher order equations), and the right-hand sides are the corresponding computed solution values.

The `listprocedure` keyword gives the output as a list of equations of the form `variable=procedure`, where the left-hand sides are the names of the independent variable, the dependent variable(s) and derivatives, and the right-hand sides are procedures that can be used to compute the corresponding solution components. This output form is most useful when the returned procedure is to be used later with, for example, `fsolve`. The `operator` keyword gives the output as a list of equations of the form `operator=procedure`, where the left-hand sides are operators that can be applied to an independent variable value to give the function or derivative evaluated at a point, and the right-hand sides are procedures that can be used to compute the corresponding solution components. This output form is most useful for shortcut evaluation forms where the returned list is to be evaluated at a point (for example, `[result](1)`).

The `piecewise` keyword is available only for non-stiff and stiff default IVP and DAE methods (`rkf45`, `ck45`, `rosenbrock`, `rkf45_dae`, `ck45_dae`, and `rosenbrock_dae`) and the `taylorseries` method. It provides output as a list of equations of the form `variable=pwfunc`, where the left-hand sides are the names of the independent variable, the dependent variable(s) and derivatives, and the right-hand sides are piecewise polynomial functions describing the corresponding solution components. These piecewise functions are obtained from the method interpolants for each step of the computation. In addition, the form of the piecewise polynomials can be specified using an index on the piecewise output request. Specification of `output=piecewise[horner]` provides output in terms of horner-form polynomials (the default for `taylorseries`), while specification of `output=piecewise[polynomial]` provides output in terms of standard polynomial form (the default for all other methods).

Use of this output form requires that the 'range' argument be used to specify to `dsolve/numeric` the desired range of the piecewise function. If an array is used, it must be a vector of floats that defines the values of the independent variable at which solution values are desired. When this option is given, a 2 x 1 matrix is returned (instead of a solution procedure). The [1,1] entry is an array containing the name of the independent variable and the names of the dependent variable(s) and derivatives. The [2,1] entry is a matrix. The first column of this matrix is a copy of the output vector, that is, the values of the independent variable. The other columns are values of the dependent variable(s) and derivatives corresponding to the vector of names in [1,1]. Row *i* of this matrix is the vector containing both the value of the independent variable and the values of the dependent variables evaluated at element *i* of the output array.

Note: If an Array is used, then the output is the same as for an array except the newer Array and Matrix datatypes are used in the output. For a large number of output points and/or solution components, the output data may not be directly visible because there is a default size at which the representation of an Array or Matrix is displayed instead of an inline display of the data itself. For more information, see `interface(rtablesize)`.

'stiff'= boolean

Boolean that is used only for IVP and DAE. Setting `stiff=true` without selecting a method specifies that the default stiff methods (`rosenbrock` or `rosenbrock_dae`) must be used instead of the default non-stiff methods (`rkf45` or `rkf45_dae`). When the method is also specified, a consistency check is performed to verify that the method matches with the 'stiff' value. For more information on the phenomena of stiffness, see `dsolve[Stiffness]`.

'events'= list

'event_pre'= keyword

'event_maxiter'= integer

Arguments that are available only for real-valued IVPs or DAEs using the non-stiff and stiff methods, `rkf45`, `ck45`, `rosenbrock`, `rkf45_dae`, `ck45_dae`, or `rosenbrock_dae`, and are used to specify and control the behavior of events. These options are discussed in detail in `dsolve[Events]`.

'range'= numeric...numeric

Values that specify the range of the independent variable over which solution values are desired.

For IVPs and DAEs this option is used only by the non-stiff and stiff methods (`rkf45`, `ck45`, `rosenbrock`, `rkf45_dae`, `ck45_dae`, `rosenbrock_dae`) and the `taylorseries` method. It has two purposes for procedure-type output. If 'range' is used, then the call to `dsolve` computes the solution over the desired range before returning, storing that solution for later calls to the returned procedure, which then compute the return values through interpolation.

Notes:

- When computing a numerical solution for a problem that has large regions where the solution is changing slowly, and small regions where the solution is changing rapidly, use of 'range' combined with the `refine` option of `odeplot` (`rkf45*`, `ck45*`, and `rosenbrock*` only) allows better plotting of the details of the solution. If 'range' is not used, then the call to `dsolve` returns a procedure that does not store the computed solution, but rather computes the solution whenever a point is requested.
- For long-time integration problems it is suggested that 'range' not be used, as the storage of the entire solution can consume a fair bit of memory.

For the BVP method, this option is only needed when the BVP method is to be used to solve an IVP with a global error bound, as for BVP the information can be inferred from the boundary conditions in `deqns`.

For more information, see `numeric,BVP`.

'abserr'= numeric or list

Numeric value that gives a limit on the absolute error tolerance for a successful step in the IVP and DAE cases, and a measure of the maximum error between the computed solution and the exact solution in the BVP case (in all but exceptional systems). It is supported by all methods except the classical methods (as all classical methods are implemented with a fixed step size and no error control). The list form of abserr is available for the rkf45*, ck45*, rosenbrock*, and mebdfi IVP and DAE methods, and allows specification of absolute error tolerances that are different for each dependent variable, or for each solution component. More detail on this can be found in the dsolve[numeric,IVP] page and the dsolve[Error_Control] page.

'relerr'= numeric

Numeric value that gives a limit on the relative error tolerance for a successful step for an IVP or DAE. This option works in conjunction with 'abserr'. It is supported by all methods except classical, taylorseries, and bvp. Defaults for 'abserr' and 'relerr' are specific to each method, but it should be noted that the default error tolerance for the non-stiff methods rkf45 and ck45 is now fixed (rather than controlled by the value of Digits). For a detailed discussion about error control, see dsolve[Error_Control].

'known'= name or list of names

Provides a list of functions that should be considered as known when examining the system. This allows for specification of a system containing user-defined functions. Each user defined function must always return numerical values when given numeric input, and must return unevaluated when given name input. For an illustration of the use of this option, see the Examples section below.

Note: In most cases, use of a system with a user-defined function prevents the use of evalhf within dsolve, so obtaining a solution using this facility will run noticeably slower than specification of the system in terms of known functions (when it is possible to express the system in this way).

All methods except taylorseries, rosenbrock, and rosenbrock_dae support this option directly, but for these two methods additional information is required. The key problem is that these two methods require information on the derivatives of the input ODE system, which is not directly available when functions are procedure defined. It is still possible to use these methods with 'known', as long as the derivatives of the function are defined via use of one or more 'diff/' rules. The rosenbrock method requires only a single derivative, but the taylorseries method requires derivatives to arbitrary order, so with taylorseries it is useful only if some finite derivative of the 'known' functions can be represented as a function instead of a procedure, in which case, all derivatives up to that function derivative must have a diff rule defined. The rosenbrock_dae method may require more than a single derivative, as the DAE preprocessing may need to differentiate the input system. For taylorseries, rosenbrock, and rosenbrock_dae use of multiple argument functions, or functions that also depend on the dependent variables or derivatives of the problem are tricky, and have some limitations.

The details of this are deferred to the respective help pages.

'optimize'= boolean

Boolean that tells dsolve to optimize the input system for computation. This adds an up-front cost, with the possible reward that the core computation itself will be faster. This is false by default for IVP and BVP problems, and true by default for DAE problems (which typically derive

the most benefit from optimization). Systems containing conditional evaluation functions (such as `piecewise`) should not use `optimize=true`, as this will sometimes result in a slower computation. This option is available only for system defined problems, and for all methods except `taylorseries` (which uses its own form of optimization).

- . All other options, including options for specification of procedures for the evaluation of the ODE system, and options specific to IVP or BVP are discussed in `dsolve[numeric,IVP]` and `dsolve[numeric,BVP]`, or the help pages specific to each method.

Notes

- . When `Digits<=evalhf(Digits)`, numerical solutions are computed in double precision (hardware floats). This is the default, as on initialization Maple sets `Digits` to 10. Explicit setting of `Digits>evalhf(Digits)` causes `dsolve` to compute numerical solutions using Maple floats instead of hardware floats. The precision of a computation is fixed in the call to `dsolve`, so for procedure outputs, once the procedure is created, further changes to the setting of `Digits` should have no effect on the computed solution.
Numerical solutions can be used in combination with `fsolve`, but the `Digits` setting on the call to `fsolve` must be lower (usually by 5 or more) than the precision for the `dsolve` procedure (because `fsolve` requires a higher accuracy on the residual than the current `Digits` setting).
The `rkf45`, `ck45`, `rosenbrock`, and `taylorseries` methods, when computing solution values with an interpolant, work around this problem by computing the interpolant with the current setting of `Digits` when called from `fsolve`. An example of the problem is presented in the last example below.
- . Results can be plotted by using the `odeplot` function in the `plots` package.
- . ODE systems can also be interactively solved and plotted with the `dsolve[interactive]` Maplet interface.

plots[odeplot] - 2-D or 3-D plot of output from dsolve

Calling Sequence

```
odeplot(dsn,vars,range,options)
```

Parameters

```
dsn      - output from a call to dsolve(...,numeric)
vars     - (optional list) axes and functions to plot
range    - (optional) range; range of the independent variable
options  - (optional) equations that specify plot options;
           see ?plot[options] and ?plot3d[option]
```

Kommandos dsolve/numeric und odeplot mit Ausgabevarianten

```
> dsol1 :=dsolve({dgl=0,y(0)=1},y(x),type=numeric):
dsol2 :=dsolve([dgl=0,y(0)=1/2],y(x),numeric):
dsol2a:=dsolve({dgl=0} union {y(0)=1/2},y(x),numeric):
dsol3 :=dsolve({dgl=0,y(0)=3/2},y(x),'numeric',method=gear):
p1:=odeplot(dsol1,[x,y(x)],-2..2,color=black,thickness=3):
p2:=odeplot(dsol2,[x,y(x)],-2..2,color=green,thickness=3):
p3:=odeplot(dsol3,[x,y(x)],-2..2,color=blue,thickness=3,numpoints=100):
display(p1,p2,p3,view=[-2..2,-2..5]);
```

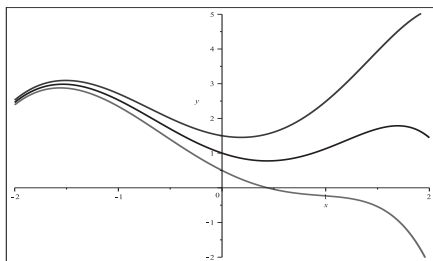


Abb. 2.5

Datei *richt_field_10.ps*,

numerische Lösungen zum AWP (2.1)

in $[-2, 2] \times [-2, 5]$

bei den AB $y(0) = \frac{1}{2}, 1, \frac{3}{2}$,

Kommandos

`dsolve/numeric` und `odeplot`

Achtung: Bei `dsolve/numeric` mit der Option `range` hat man bzgl. der numerischen Verfahren nur die Wahl der Standardmethode `rkf45`.

Eine Option von `dsolve/numeric` betrifft den Output der erzeugten Prozedur.

`output=procedurelist`

Das ist die Standardeinstellung.

Sie erzeugt eine einzige Prozedur, deren Ergebnis zu jedem Wert der unabhängigen Variablen eine Liste der Form $[x=\dots, y(x)=\dots, z(x)=\dots, \dots]$ oder auch z.B.

$[t=\dots, x(t)=\dots, y(t)=\dots, \dots]$ ist.

Die Standardmethode ist das Verfahren von Runge-Kutta-Fehlberg 4(5).

```
> ic1:=y(0)=1;
> dsol4:=dsolve({dgl=0,ic1},y(x),numeric); # default output=procedurelist
                dsol4 := proc(x,rkf45)...end proc
```

Übliche sowie sonstige Weiterverarbeitung

```
> dsol4(0),dsol4(1),dsol4(2);
[rhs(dsol4(0)[1]),rhs(dsol4(0)[2])];
[x = 0., y(x) = 1.], [x = 1., y(x) = 1.1268726346382616], [x = 2., y(x) = 1.443777067254847]
[0., 1.]

> pl4:=odeplot(dsol4, [x, y(x)], x=-2..2, color=black, thickness=3,
view=[-2..2, 0..3]):
display(pl4);
sq:=seq([0.05*k, rhs(dsol4(0.05*k)[2])], k=-40..40):
pl4a:=plot(sq, x=-2..2, color=black, thickness=3, view=[-2..2, 0..3]):
display(pl4a);
```

Die Prozedurliste mit variablem Argument lässt sich nicht ohne Weiteres auswerten.

Nur im Zusammenhang mit einer Funktionsdefinition sind deren Berechnung und grafische Darstellung möglich.

```
> dsol4(x), op(1,dsol4(x));
op(2,dsol4(x)); # Fehler
rhs(op(1,dsol4(x))); # Fehler
rhs(op(2,dsol4(x))); # Fehler
                dsol4(x), x
```

```

Error, improper op or subscript selector
Error, invalid input: rhs received x, which is not valid for its
    1st argument, expr
Error, improper op or subscript selector

> xx:=x->rhs(op(1,dsol4(x)));
yy:=x->rhs(op(2,dsol4(x)));
xx(x);
yy(x);
                                xx := x → rhs(op(1, dsol4(x)))
                                yy := x → rhs(op(2, dsol4(x)))
Error, (in xx) invalid input: rhs received x, which is not valid for
    its 1st argument, expr
Error, (in yy) improper op or subscript selector

> # definierte Argumente
xx(0),yy(0);
xx(1),yy(1);
xx(2),yy(2);
                                0., 1.
                                1., 1.12687263463826
                                2., 1.44377706725485

> # Operatorform der Funktion zulaessig
p14b:=plot(yy,-2..2,color=black,thickness=3,view=[-2..2,0..3]):
> # Damit funktioniert die "normale" Darstellung, 'yy(x)' wie Parameterform
p14b:=plot('yy(x)', 'x'=-2..2,color=black,thickness=3,view=[-2..2,0..3]):
p14b:=plot('yy(x)', x=-2..2,color=black,thickness=3,view=[-2..2,0..3]):
display(p14b);

> # Funktionsform nicht anwendbar
x:='x':
plot(yy(x),x=-2..2);
Error, (in yy) improper op or subscript selector

```

output=listprocedure

Hier wird eine Liste von Prozeduren erzeugt, deren Elemente als einzelne Prozeduren weiter verarbeitet werden können. Es gibt also mehr Möglichkeiten der Verarbeitung als Funktion und Grafik mittels `subs`, `eval` oder `assign`.

```

> dsol5:=dsolve({dgl,ic1},numeric,output=listprocedure);
                                dsol5 := [x = proc(x)...end proc, y(x) = proc(x)...end proc]

> # diese Darstellung ungeeignet, vergleiche mit dsol4(0)
dsol5(0);
                                [x(0) = 0., (y(x))(0) = 1.]

> dsol5x:=subs(dsol5,x);
dsol5y:=subs(dsol5,y(x));
dsol5y(x);
dsol5x(0),dsol5x(1),dsol5x(2);
dsol5y(0),dsol5y(1),dsol5y(2);

```

```

dsol5x := proc(x)..end proc
dsol5y := proc(x)..end proc
          dsol5y(x)
          0., 1., 2.
1., 1.12687263463826, 1.44377706725485

> fx5:=eval(x,dsol5);
  fy5:=eval(y(x),dsol5);
  fy5(x);
  fx5(0),fx5(1),fx5(2);
  fy5(0),fy5(1),fy5(2);

          fx5 := proc(x)..end proc
          fy5 := proc(x)..end proc
                fy5(x)
                0., 1., 2.
1., 1.12687263463826, 1.44377706725485

> printf('      x          y(x)\n');
  for x from 0 to 2 by 0.4 do
    printf(' %4.1f %9.6f %9.6f\n',x,dsol5y(x),fy5(x));
  end do;

x          y(x)
0.0  1.000000  1.000000
0.4  0.776701  0.776701
0.8  0.929836  0.929836
1.2  1.367532  1.367532
1.6  1.763871  1.763871
2.0  1.443777  1.443777

> x:='x':
  int(fy5(x),x=0..2);
  evalf(%);

          
$$\int_0^2 fy5(x) dx$$

2.44377657788381

> # Vergleich
p15a:=plot(fy5(x),x=-2..2,color=black,thickness=3,view=[-2..2,0..3]):
p15a:=plot('fy5(x)',x=-2..2,color=black,thickness=3,view=[-2..2,0..3]):
p15a:=plot('fy5(x)', 'x'=-2..2,color=black,thickness=3,view=[-2..2,0..3]):
display(p15a);

> p15:=odeplot(dsol5, [x,y(x)],x=-2..2,color=black,thickness=3,
  view=[-2..2,0..3]):
display(p15);
#p15b:=plot(dsol5y(x),x=-2..2,color=black,thickness=3,numpoints=100,
#  view=[-2..2,0..3]):
p15b:=plot(fy5(x),x=-2..2,color=black,thickness=3,numpoints=100,
  view=[-2..2,0..3]):
display(p15b);

```

output=operator

```
> dsol6:=dsolve({dgl,ic1},numeric,output=operator);
          dsol6 := [x = proc(x)...end proc, y(x) = proc(x)...end proc]

> dsol6(0),dsol6(1),dsol6(2);
dsol6(1)[2];
rhs(dsol6(1)[2]);

[x = 0, y(0) = 1.], [x = 1., y(1) = 1.12687263463826], [x = 2., y(2) = 1.44377706725485]
          y(1) = 1.12687263463826
          1.12687263463826

> fy6:=unapply(rhs(dsol6(x)[2]),x);
fy6(0),fy6(1),fy6(2);

          fy6 := x -> y(x)
          1., 1.12687263463826, 1.44377706725485

> pl6:=odeplot(dsol6, [x,y(x)],x=-2..2,color=black,thickness=3,
          view=[-2..2,0..3]):
display(pl6);
pl6a:=plot(fy6(x),x=-2..2,color=black,thickness=3,numpoints=100,
          view=[-2..2,0..3]):
display(pl6a);
```

output=piecewise

```
> dsol7:=dsolve({dgl,ic1},numeric,output=piecewise,range=0..2);

          dsol7 := [x = x, y(x) = {[undefined, x < 0.],
          [0.999215299421484 - 0.943719362778975 x + 1.02698142908849 (x - 0.0277621231571285)2 + ...

> # Fehler
# odeplot(dsol7, [x,y(x)],x=0..2,color=black,thickness=3):
> dsol7[1];
dsol7[2]:
          rhs(dsol7[2]):
          # lange Formeln
          rhs(dsol7[2]),x);
          fy7:=unapply(rhs(dsol7[2]),x);
          fy7(0),fy7(1),fy7(2);

          x = x
          fy7 := x -> piecewise(x < 0., undefined,
          x ≤ 0.0555242463142570086, 0.9992152994214841 - 0.9437193627789752 x + ...
          ...(x - 1.9647357710489568)4, undefined)
          1., 1.12687263463826, 1.44377706725485

> pl7:=plot(fy7(x),x=0..2,color=black,thickness=3,view=[0..2,0..3]):
display(pl7);
```

output=array

```
> dsol8:=dsolve({dgl,ic1},numeric,output=array([seq(0.1*k,k=0..20)]);
eval(dsol8[1,1]): # 1.Zeile
eval(dsol8[2,1]): # 2.Zeile/Block
```

$$dsol8 := \begin{bmatrix} & & x & y(x) \\ & & & \\ & 0. & & 1. \\ 0.1000000000000000 & & & 0.910316448054921 \\ 0.2000000000000000 & & & 0.842389025602480 \\ 0.3000000000000000 & & & 0.797565007640731 \\ \dots & & & \dots \\ & 1. & & 1.12687263463826 \\ 1.1000000000000000 & & & 1.24433642656435 \\ \dots & & & \dots \\ 1.9000000000000000 & & & 1.64542363967616 \\ & 2. & & 1.44377706725485 \end{bmatrix}$$

```
> pl8:=odeplot(dsol8,[x,y(x)],color=black,thickness=3):
pl8a:=odeplot(dsol8,style=point,color=black,
symbol=solidcircle,symbolsize=16):
display(pl8,pl8a,view=[0..2,0..3]);
```

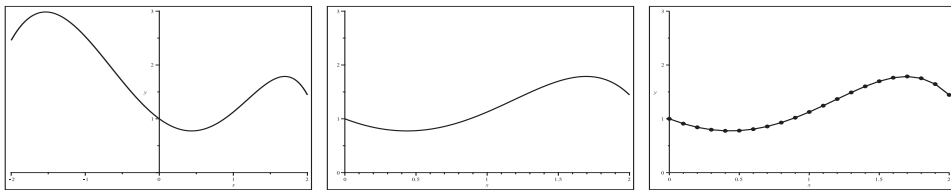


Abb. 2.6 Dateien *richt_field_11.ps* (zu p14,p14a,p14b,p15,p15a,p15b,p16,p16a), *richt_field_12.ps* (zu p17), *richt_field_13.ps* (zu p18,p18a), numerische Lösung zum AWP $y' - y + x^3 - 3x + 2 = 0$, $x \in [-2, 2]$ bzw. $[0, 2]$, mit der AB $y(0) = 1$, Kommandos `dsolve`, `odeplot`

Überblick über weitere ausgewählte Verfahren

- (1) Klassische Verfahren mit konstanter Schrittweite

```
dsolve(deqns,vars,type=numeric,method=classical) oder
dsolve(deqns,vars,type=numeric,method=classical[<Methode>])
```

- (2) (Eingebettete) explizite Runge-Kutta-Verfahren mit automatischer Schrittweitensteuerung

```
dsolve(deqns,vars,type=numeric,method=rkf45),
dsolve(type=numeric,method=rkf45,procopts),
dsolve(deqns,vars,type=numeric,method=dverk78),
dsolve(type=numeric,method=dverk78,procopts),
dsolve(deqns,vars,type=numeric,method=ck45) und
dsolve(type=numeric,method=ck45,procopts)
```


- (3) Explizite Extrapolationsverfahren von Gear und Bulirsch-Stoer mit automatischer Schrittweitensteuerung
`dsolve(deqns, vars, type=numeric, method=gear)` oder
`dsolve(deqns, vars, type=numeric, method=gear [<Methode>])`
- (4) Implizite Rosenbrock-Verfahren der Ordnung 3(4) für steife Systeme
`dsolve(deqns, vars, type=numeric, method=rosenbrock)` und
`dsolve(type=numeric, method=rosenbrock, procopts)`
- (5) Implizite Mehrschrittverfahren für steife Systeme (Livermore Stiff ODE Solver), auch mit strukturierter rechter Seite
`dsolve(deqns, vars, type=numeric, method=lsode)` oder
`dsolve(deqns, vars, type=numeric, method=lsode [<Methode>])`
- (6) Prädiktor-Korrektor-Mehrschrittverfahren mit automatischer Schrittweitensteuerung für steife und nichtsteife Systeme (ältere Maple-Versionen)
`dsolve(deqns, vars, type=numeric, method=mgear)` oder
`dsolve(deqns, vars, type=numeric, method=mgear [<Methode>])`

Ein Nutzer braucht viel Erfahrung im Umgang mit gDGL und SysgDGL, um entsprechende Verfahren auszuwählen. Diese erfordern u.a. Anpassungen der Toleranzen und geeignete Verfahrensparameter.

Für ein AWP wählen wir einige numerische Verfahren aus und vergleichen diese.

Gegeben sei als Testbeispiel das AWP (siehe auch Abschnitt 1.8.1 Nr. 2, 36)

$$y' + y = \text{sign}(\sin(\pi t)), \quad y(0) = 2. \quad (2.2)$$

Es beschreibt die Rechteckanregung eines RC-Kreises (Schaltungen, die aus einem ohmschen Widerstand R und einem Kondensator C aufgebaut sind; RC-Glieder sind lineare, zeitinvariante Systeme).

Wir betrachten zunächst Lösungen dieses und anderer AWP sowie das Richtungsfeld. Da die rechte Seite der gDGL Unstetigkeitsstellen an ganzzahligen Argumenten $t = k$ hat und damit auch Sprungstellen im Richtungsfeld auftreten werden, ist einiges zu beachten.

In den offenen Intervallen $(k, k + 1)$, evtl. mit AB dort, kann man die gDGL bzw. das AWP lösen. Das Kommando `dsolve` liefert jedoch auch globale Lösungen zu AWP, die jedoch die Unstetigkeiten der rechten Seite “übernehmen“, damit nicht überall differenzierbar und als Lösungen nicht zu akzeptieren sind. Maple kann das Problem lösen, aber “auf seine Weise“.

```
> dg11:=diff(y(t),t)+y(t)=signum(sin(Pi*t)):
  init1:=y(0)=2:
> # explizite Loesung
  dsolve(dg11,y(t));
  dsolve({dg11,init1},y(t));
  dg11_ab:=dsolve({dg11,init1},y(t));
```

$$y(t) = \text{signum}(\sin(\pi t)) + e^{-t} \cdot C1$$

$$y(t) = \text{signum}(\sin(\pi t)) + 2e^{-t}$$

$$\text{dgl1_ab} := y(t) = \text{signum}(\sin(\pi t)) + 2e^{-t}$$

```
> f:=t->1+exp(-t);      # Loesung von y'+y=1 in [0,1] mit AB y(0)=2

> f1:=rhs(dgl1_ab);
f2:=rhs(dsolve({dgl1,y(0)=1},y(t)));
f3:=rhs(dsolve({dgl1,y(0)=0},y(t)));
f4:=rhs(dsolve({dgl1,y(0)=-1},y(t)));

      f := t -> 1 + e^{-t}
      f1 := signum(sin(pi*t)) + 2e^{-t}
      f2 := signum(sin(pi*t)) + e^{-t}
      f3 := signum(sin(pi*t))
      f4 := signum(sin(pi*t)) - e^{-t}
```

Bei der grafischen Ausgabe des Richtungsfeldes mit den obigen unstetigen Lösungsfunktionen gibt es eine unklare Fehlermeldung, wenn man die Genauigkeitseinstellung `Digits` verwendet, und zwar bei zu großer Stellenanzahl.

Mit der Option `discont=true` kann man an den Unstetigkeitsstellen die Sprünge vollziehen.

```
> Digits:=16:      # >=16 -> Fehler
> # Unstetigkeiten der rechten Seite, Spruenge im Richtungsfeld
pl11:=dfieldplot(dgl1,y,t=-2..4.1,y=-4..8,color=gray,arrows=small,
  dirgrid=[30,30]):
pl12:=plot([f1,f2,f3,f4],t=-2..4.1,thickness=2,
  color=[black,green,blue,magenta]):
pl13:=pointplot([[0,2],[0,1],[0,0],[0,-1]],
  color=[black,green,blue,magenta],
  symbol=solidcircle,symbolsize=16):
display(pl11,pl12,pl13,axes=normal);
```

Error: Error creating plot:com.maplesoft.client.dag.RtableDagData
cannot be cast to com.maplesoft.client.dag.RtableDoubleData

```
> Digits:=15:      # <=15 -> o.k.
# Unstetigkeiten der rechten Seite, Spruenge im Richtungsfeld
pl11:=dfieldplot(dgl1,y,t=-2..4.1,y=-4..8,color=gray,arrows=small,
  dirgrid=[30,30]):
# pl12:=plot([f1,f2,f3,f4],t=-2..4.1,thickness=2,
#   color=[black,green,blue,magenta],discont=true):
pl12:=plot([f1,f2,f3,f4],t=-2..4.1,thickness=2,
  color=[black,green,blue,magenta]):
pl13:=pointplot([[0,2],[0,1],[0,0],[0,-1]],
  color=[black,green,blue,magenta],
  symbol=solidcircle,symbolsize=16):
display(pl11,pl12,pl13,axes=normal);
```

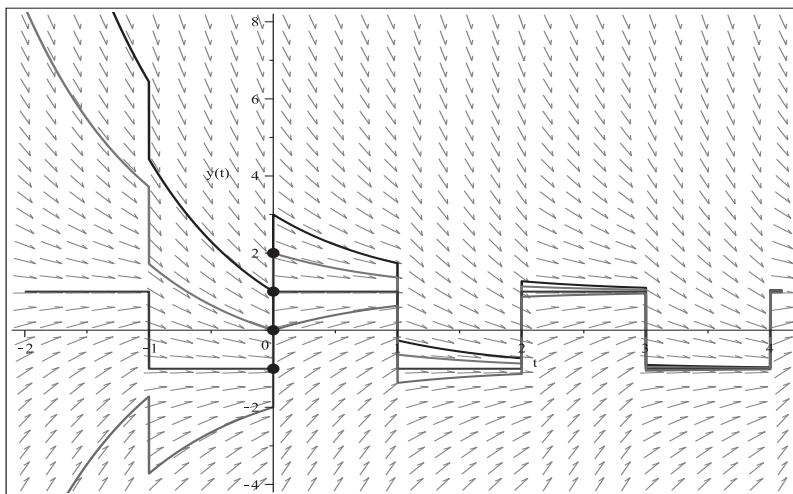


Abb. 2.7 Datei *rc_kreis_01.ps*,

Richtungsfeld und Lösungen zum AWP $y' + y = \text{sign}(\sin(\pi t))$, $t \in [-2, 4]$,
mit den AB $y(0) = 2, 1, 0, -1$, Kommandos `dsolve`, `dfieldplot`

Wir werden nun die gDGl (2.2) intervallweise exakt lösen, wobei der Endwert in einem Intervall die AB für das nächste Intervall definiert. Damit entsteht zumindest eine stetige Gesamtlösung, auch wenn die Differenzierbarkeit an den Intervallenden weiterhin nicht vorliegt.

Der Grund für diese Berechnung ist, dass die numerischen Verfahren mit ihren reellwertigen Schrittweiten i . Allg. diese Stellen nicht treffen und überspringen und damit eine stetige Näherungslösung erzeugen können.

```
> # in [0,1], [2,3]
dgl01:=diff(y(t),t)+y(t)=1:
dsolve(dgl01,y(t));

# in [1,2], [3,4]
dgl02:=diff(y(t),t)+y(t)=-1:
dsolve(dgl02,y(t));

y(t) = 1 + e-t _C1
y(t) = -1 + e-t _C1

> init01:=y(0)=2:
dgl1_01:=dsolve({dgl01,init01},y(t));
# im ersten Intervall [0,1] wie f(t)=1+exp(-t)
init02:=y(1)=1+exp(-1):
dgl1_02:=simplify(dsolve({dgl02,init02},y(t)));
init03:=y(2)=-1+exp(-2)+2*exp(-1):
dgl1_03:=simplify(dsolve({dgl01,init03},y(t)));
init04:=y(3)=1-2*exp(-1)+2*exp(-2)+exp(-3):
dgl1_04:=simplify(dsolve({dgl02,init04},y(t)));
```

```

      dgl1_01 := y(t) = 1 + e-t
      dgl1_02 := y(t) = -1 + 2e-t+1 + e-t
      dgl1_03 := y(t) = 1 - 2e-t+2 + e-t + 2e-t+1
      dgl1_04 := y(t) = -1 + 2e-t+3 - 2e-t+2 + 2e-t+1 + e-t

> p101:=plot(rhs(dgl1_01),t=0..1,color=black,thickness=3):
  p102:=plot(rhs(dgl1_02),t=1..2,color=blue,thickness=3):
  p103:=plot(rhs(dgl1_03),t=2..3,color=green,thickness=3):
  p104:=plot(rhs(dgl1_04),t=3..4,color=red,thickness=3):
  p105:=dfieldplot(dgl1,y,t=0..4.1,y=-0.5..2.5,color=black,
    arrows=small,dirgrid=[40,30]):
  display(p105,p101,p102,p103,p104,view=[0..4,-0.5..2.5]);

> yex:=unapply(piecewise(t<1,rhs(dgl1_01),
    t<2,rhs(dgl1_02),
    t<3,rhs(dgl1_03),
    rhs(dgl1_04)),t):
  yex(0),yex(1),yex(2),yex(3),yex(4);
  evalf(%);

      2, 1 + e-1, -1 + e-2 + 2e-1, 1 - 2e-1 + 2e-2 + e-3, -1 + 2e-1 - 2e-2 + 2e-3 + e-4
      2., 1.367879441171442, -0.1289058344205027, 0.5846987524982047, -0.4170219085058787

> p100:=plot(yex(t),t=0..4,color=black,thickness=3):
  p100a:=pointplot([[0,2],[1,yex(1)],[2,yex(2)],[3,yex(3)]],color=black,
    symbol=solidcircle,symbolsize=16):
  display(p105,p100,p100a,view=[0..4,-0.5..2.5]);

```

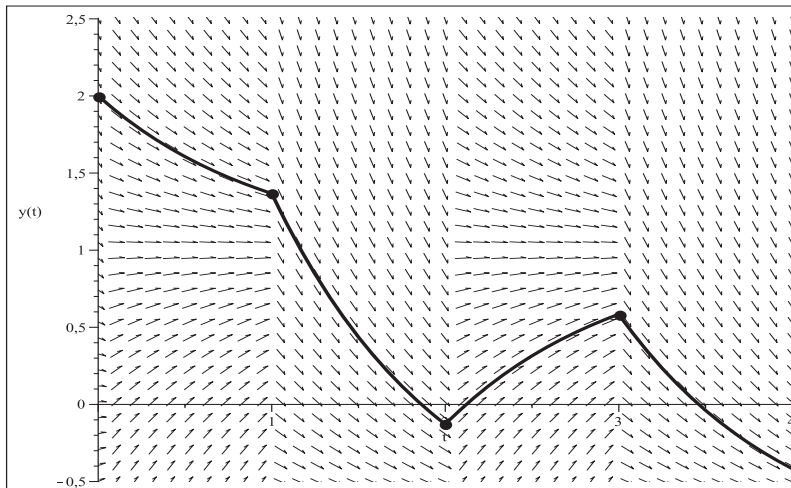


Abb. 2.8 Datei *rc_kreis_02.ps*,

Richtungsfeld und Lösung zum AWP $y' + y = \text{sign}(\sin(\pi t))$, $t \in [0, 4]$,
 mit AB $y(0) = 2$ und zu $y(1), y(2), y(3)$, intervallweise erzeugt,
 Kommandos `dsolve`, `dfieldplot`

Nun testen wir einige numerische Verfahren.

(a) Zur Integration in einem Bereich ohne Unstetigkeitsstellen der rechten Seite wird das Intervall $(0, 1)$ gewählt.

Mit der AB $y(0) = 2$ haben wir oben die Lösung $f(t) = 1 + e^{-t}$ erhalten. Daraus leiten wir die neue innere AB $y(0.5) = 1 + e^{-0.5}$ ab, sodass die numerischen Verfahren Näherungslösungen im Intervall $(0, 1)$ berechnen können. Von Interesse ist auch das Verhalten an den Intervallgrenzen, ob dort Näherungswerte berechnet werden. Denn es ist ein Unterschied, ob man auf $[0, 1]$ die gDGL (2.2) oder die gDGL $y' + y = 1$ löst.

Die Verfahren werden ohne spezielle Optionen aufgerufen. Die Rechenzeiten sind generell sehr klein.

```
> Digits:=16:
p111a:=dfieldplot(dg11,y,t=-0.1..1.1,y=0..2.5,color=gray,
    arrows=small,dirgrid=[25,26]):
p111b:=plot([[1,0],[1,1.37]],color=black):
p111c:=pointplot([0.5,1+exp(-0.5)],color=black,symbol=solidcircle,
    symbolsize=20):
init0:=y(0.5)=1+exp(-0.5):
```

(1) Das explizite ESV von Runge-Kutta-Fehlberg der Ordnung 4(5) (RKF45) ist das Standardverfahren.

Die entsprechenden Kommandos für den Aufruf und die Auswertung der Ergebnisse machen wir für die erste Methode etwas ausführlicher. Insbesondere zeigen wir die Näherungswerte an den Intervallgrenzen 0 und 1, um später Vergleiche bzgl. der Genauigkeit anzustellen.

```
> time0:=time(): # Startzeit
sol1:=dsolve({dg11,init0},y(t),type=numeric):
# Runge-Kutta-Fehlberg 4(5), Standard
cpu_time:=(time()-time0)*seconds; # Rechenzeit

> time0:=time():
sol1:=dsolve({dg11,init0},y(t),numeric,method=rkf45):
cpu_time:=(time()-time0)*seconds;

> sol1(0);
sol1(0.5); # =1+exp(-0.5)
sol1(1);
seq(sol1(k*0.1),k=0..10): # zur Kontrolle
[t = 0., y(t) = 1.999999924593291]
[t = 0.5, y(t) = 1.606530659712633]
[t = 1., y(t) = 1.367872248992532]

> p11:=odeplot(sol1,[t,y(t)],0..1,color=black,thickness=3):
display(p11,p111b,p111c,view=[0..1,0..2.5]);

> display(p111a,p111b,p111c,p11,view=[-0.1..1.1,0..2.5]);
```

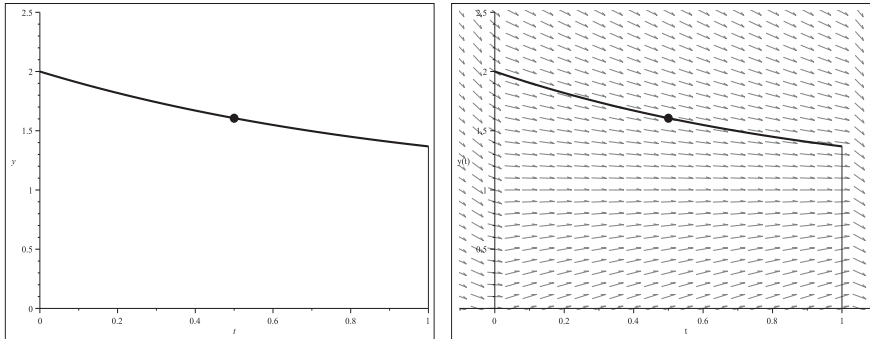


Abb. 2.9 Dateien *rc_kreis_03.ps*, *rc_kreis_04.ps*,
 Näherungslösung zum AWP $y' + y = \text{sign}(\sin(\pi t))$, $t \in [0, 1]$,
 mit AB $y(0.5) = 1 + e^{-0.5}$, rechts mit Richtungsfeld, Standardverf. RKF45,
 Kommandos `dsolve/numeric/rkf45`, `dfieldplot`

Bei den anderen Verfahren erkennt man in den Abbildungen keine qualitativen Unterschiede.

(2) Explizites ESV von Euler der Ordnung 1, PZV

```
> init0:=y(0.5)=1+exp(-0.5):
> time0:=time():
  sol2:=dsolve({dgl1,init0},y(t),numeric,method=classical):
                                     # method=classical[foreuler]
  cpu_time:=(time()-time0)*seconds;

> sol1(0);
  sol1(1);
  seq(sol1(k*0.1),k=0..10): # zur Kontrolle
                               [t = 0., y(t) = 1.998754926851450]
                               [t = 1., y(t) = 1.367418342479132]

> p12:=odeplot(sol2, [t,y(t)], 0..1, color=black, thickness=3):
  display(p12,p11b,p11c,view=[0..1,0..2.5]);

> display(p11a,p11b,p11c,p12,view=[-0.1..1.1,0..2.5]);
```

Nach demselben Muster werden noch weitere sechs Methoden in den Test einbezogen.

- (3) Explizites klassisches RKV 4. Ordnung (KRKV).
- (4) Explizites lineares MSV von Adams-Bashforth 4. Ordnung.
- (5) Prädiktor-Korrektor-Verfahren von Adams-Bashforth-Moulton 4. Ordnung.
- (6) Implizites Verfahren von Rosenbrock der Ordnung 3(4).
- (7) Explizites ESV von Burlisch-Stoer variabler Ordnung mit Extrapolation.
- (8) Explizites kontinuierliches RKV der Ordnung 7(8) (RKV78).

Beim letzten Verfahren gibt es zu den Berechnungen an der linken Intervallgrenze 0 ein kleines Problem.

```
> sol3:=dsolve({dgl1,init0},y(t),numeric,method=classical[rk4]):
  sol4:=dsolve({dgl1,init0},y(t),numeric,method=classical[adambash]):
  sol5:=dsolve({dgl1,init0},y(t),numeric,method=classical[abmoulton]):
  sol6:=dsolve({dgl1,init0},y(t),numeric,method=rosenbrock):
  sol7:=dsolve({dgl1,init0},y(t),numeric,method=gear[bstoer]):
> sol8:=dsolve({dgl1,init0},y(t),numeric,method=dverk78):
> sol8(0);
  limit(sol8(z),z=0,right);
  sol8(1);
  seq(sol8(k*0.1),k=1..10):
> pl8:=odeplot(sol8,[t,y(t)],0..1,color=black,thickness=3):
  display(pl8,pl11b,pl11c,view=[0..1,0..2.5]);
```

Error, (in sol8) cannot evaluate the solution past 0.14305115e-6, step size < hmin, problem may be singular or error tolerance may be too small

$$[t = 1.430511474609375 \cdot 10^{-7}, y(t) = 1.99999856948826]$$

$$[t = 1., y(t) = 1.367879441172197]$$

Warning, could not obtain numerical solution at all points, plot may be incomplete

Bzgl. der Genauigkeit der Verfahren wollen wir nur den Vergleich mit dem exakten Randwerten $y(0) = 2$ und $y(1) = 1 + e^{-1} = 1.36787944117144$ anstellen. In der folgenden Tabelle ist ihre Reihenfolge bei wachsender Genauigkeit notiert.

Auffällig ist, wie unterschiedlich die Platzierungen von KRKV und RKF45 an den beiden Rändern des Lösungsintervall sind. Dies ist ein weiterer Hinweis darauf, dass man beim Umgang mit Lösungsverfahren zu gDGl viel Erfahrung haben muss.

```
> printf(' Methode                yn(0)                yn(1)    \n');
printf(' -----                -\n');
printf(' PZV, EEV                %15.12f    PZV, EEV    %15.12f\n',rhs(sol2(0)[2]),rhs(sol2(1)[2]));
printf(' AM4                    %15.12f    RKF45      %15.12f\n',rhs(sol5(0)[2]),rhs(sol1(1)[2]));
printf(' KRKV                    %15.12f    ROSENBR34  %15.12f\n',rhs(sol3(0)[2]),rhs(sol6(1)[2]));
printf(' AB4                      %15.12f    AB4        %15.12f\n',rhs(sol4(0)[2]),rhs(sol14(1)[2]));
printf(' ROSENBR34                 %15.12f    GEAR       %15.12f\n',rhs(sol6(0)[2]),rhs(sol7(1)[2]));
printf(' GEAR                      %15.12f    AM4        %15.12f\n',rhs(sol7(0)[2]),rhs(sol15(1)[2]));
printf(' DVERK78 yn(1.4e-7)=%15.12f KRKV          %15.12f\n',
  rhs(limit(sol8(z),z=0,right)[2]),rhs(sol13(1)[2]));
printf(' RKF45                    %15.12f    DVERK78    %15.12f\n',rhs(sol1(0)[2]),rhs(sol8(1)[2]));
printf(' -----                -\n');
printf(' exakt                    %15.12f                %15.12f\n',f(0),f(1));
```

Methode	yn(0)	yn(1)
PZV, EEV	1.998754926851	PZV, EEV 1.367418342479
AM4	2.000833333341	RKF45 1.367872248993
KRKV	2.000833333331	ROSENBR34 1.367879596319
AB4	2.000833333229	AB4 1.367879441210
ROSENBR34	1.99998977055	GEAR 1.367879441167
GEAR	2.00000440103	AM4 1.367879441168
DVERK78 yn(1.4e-7)=	1.99999856949	KRKV 1.367879441172
RKF45	1.99999924593	DVERK78 1.367879441172
exakt	2.000000000000	1.367879441171

(b) Wir untersuchen nun dieselben numerischen Verfahren für die Lösung des AWP (2.2) im Intervall $[0, 4]$, wenn also über Unstetigkeitsstellen hinweg integriert wird.

Zum Vergleich haben wir die stückweise definierte exakte Lösung $yex(x)$ (siehe Abb. 2.8) und die Werte an der Intervallgrenze 4. Dabei ist jedoch Folgendes zu beachten.

Numerische Verfahren mit Schrittweitensteuerung und damit variabler Schrittweite integrieren i. Allg. über die Sprünge im Richtungsfeld hinweg. Es kann jedoch auch passieren, dass Verfahren an singulären Stellen, die z.B. in den AB schon enthalten sind, überhaupt nicht losrechnen bzw. über solche Stellen nicht hinweg rechnen können.

Mit der AB $y(0) = 2$ haben wir den exakten Vergleichswert

$$yex(4) = -1 + 2e^{-1} - 2e^{-2} + 2e^{-3} + e^{-4} = -0.417021908505880.$$

```
> p11a:=dfieldplot(dgl1,y,t=-2..4,y=-2..8,color=gray,
    arrows=small,dirgrid=[25,25]):
p11b:=plot([[4,0],[4,-0.417]],color=black):
p11c:=pointplot([0,2],color=black,symbol=solidcircle,
    symbolsize=20):
init1:=y(0)=2:
```

(1) Zunächst das explizite ESV von RKF45.

Die entsprechenden Kommandos für den Aufruf und die Auswertung der Ergebnisse machen wir nur für die erste Methode. Insbesondere zeigen wir die Näherungswerte an der Intervallgrenze 4, um später Vergleiche bzgl. der Genauigkeit anzustellen.

```
> time0:=time():
sol1:=dsolve({dgl1,init1},y(t),type=numeric): # Runge-Kutta-Fehlberg 4(5)
cpu_time:=(time()-time0)*seconds;

> sol1(0);
sol1(1);
sol1(4);
seq(sol1(k*0.5),k=0..8): # zur Kontrolle
    [t = 0., y(t) = 2.]
    [t = 1., y(t) = 1.367878448106152]
    [t = 4., y(t) = -0.4170210030622212]

> p11:=odeplot(sol1,[t,y(t)],-2..4,color=black,thickness=3):
display(p11,p11b,p11c,view=[-2..4,-2..8]);
> display(p11a,p11b,p11c,p11,view=[-2..4,-2..8]);
```

Die anderen Verfahren sind analog zum Teil (a) mit den Aufrufen

```
> sol2:=dsolve({dgl1,init1},y(t),type=numeric,method=classical):
sol3:=dsolve({dgl1,init1},y(t),type=numeric,method=classical[rk4]):
sol4:=dsolve({dgl1,init1},y(t),type=numeric,method=classical[adambash]):
sol5:=dsolve({dgl1,init1},y(t),type=numeric,method=classical[abmoulton]):
sol6:=dsolve({dgl1,init1},y(t),type=numeric,method=rosenbrock):
sol7:=dsolve({dgl1,init1},y(t),type=numeric,method=gear[bstoer]):
sol8:=dsolve({dgl1,init1},y(t),type=numeric,method=dverk78):
```

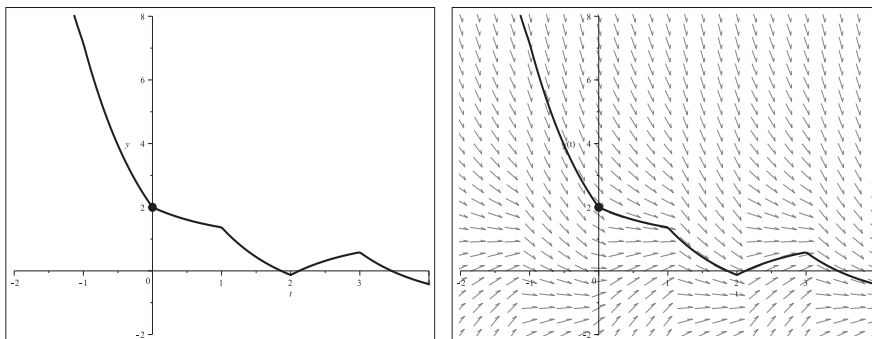



Abb. 2.10 Dateien *rc_kreis_05.ps*, *rc_kreis_06.ps*,
 Näherungslösung zum AWP $y' + y = \text{sign}(\sin(\pi t))$, $t \in [0, 4]$,
 mit AB $y(0) = 2$, rechts mit Richtungsfeld,
 Standardverfahren RKF45,
 Kommandos `dsolve/numeric/rkf45`, `dfieldplot`

Bei den ersten sechs Verfahren erkennt man in den Abbildungen keine qualitativen Unterschiede.

Aber beim ESV von Burlisch-Stoer wird nur bis zur Eins integriert und gezeichnet. Man erhält

```
> sol7(1);
  sol7(1.000001);
  sol7(1.1);
```

```
[t = 1.000000000000000, y(t) = 1.367879284969136]
[t = 1.000001, y(t) = 1.367877524494686]
```

Error, (in sol7) cannot evaluate the solution past 1.0000079,
 step taken with h = hmin but the requested error not achieved

Beim expliziten kontinuierlichen RKF78 gibt es zwei Probleme.

```
> # Loeser kann nicht bei AB (0,2) beginnen
  sol8(0);
  sol8(0.0000001);
```

```
[t = 0., y(t) = 2.]
```

Error, (in sol8) cannot evaluate the solution past 0., step
 size < hmin, problem may be singular or error tolerance may be too small

```
> # andere AB, etwas rechts von Null liegend
  sol8:=dsolve({dgl1,y(0.01)=1+exp(-0.01)},y(t),
               type=numeric,method=dverk78):
> # Loeser kommt nicht ueber die Stelle 1 hinweg
  sol8(0.01);
```

```
sol8(0.0100001);
sol8(1);
sol8(1.0000001);
```

```
[t = 0.01, y(t) = 1.990049833749168]
[t = 0.0100001, y(t) = 1.990049734744190]
[t = 1., y(t) = 1.367879441190621]
```

Error, (in sol8) cannot evaluate the solution past .99999991, step size < hmin, problem may be singular or error tolerance may be too small

Dazu die beiden Grafiken, wo aufgrund der fehlenden Berechnungen die entsprechenden Kurvenstücke natürlich fehlen.

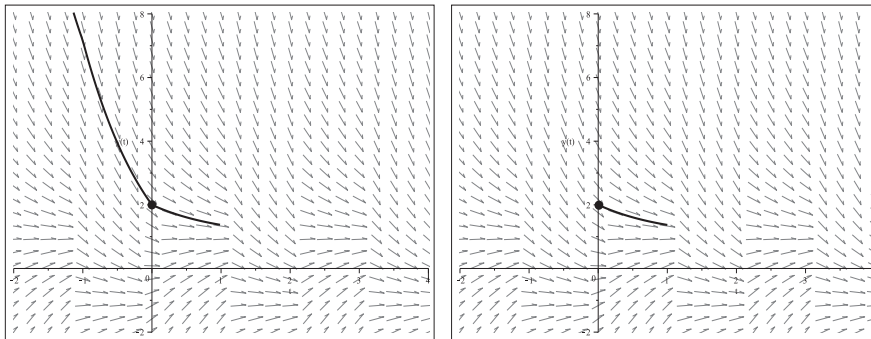


Abb. 2.11 Dateien *rc_kreis_07.ps*, *rc_kreis_08.ps*,
 Näherungslösung zum AWP $y' + y = \text{sign}(\sin(\pi t))$, $t \in [0, 4]$,
 mit AB $y(0) = 2$ bzw. $y(0.01) = 1 + e^{-0.01}$, mit Richtungsfeld,
 links: ESV von Burlisch-Stoer, rechts: kontinuierliches RKV78,
 Kommandos `dsolve/numeric/gear[bstoer], ./dverk78, dfieldplot`

Bzgl. der Genauigkeit der Verfahren wollen wir nur den Vergleich mit dem exakten Randwert `yex(4)` anstellen. In der folgenden Tabelle ist zu den ersten sechs Verfahren ihre Reihenfolge bei wachsender Genauigkeit notiert.

Methode	yn(4)
AB4	-0.415589362220
AM4	-0.415624404703
PZV, EEV	-0.415699836724
KRKV	-0.416566619496
RKF45	-0.417021003062
ROSENBROCK34	-0.417021985297
exakt	-0.417021908506

Informationen zur Abarbeitung von dsolve/numeric

Es gibt für dieses Kommando Möglichkeiten der Darstellung von Prozedurinhalt bzw. Prozedurerstellung.

1. Möglichkeit: Darstellung des Prozedurinhaltes

```
> # gDG1
deq3:={diff(y(t),t$4)-diff(y(t),t$3)=y(t)*t^2};
init3:={y(0)=3.56,D(y)(0)=12,(D@@2)(y)(0)=-4,(D@@3)(y)(0)=6.544};

> # Einstellen der Option
interface(verboseproc=1): # wie interface(verboseproc=0): # Kurzversion
interface(verboseproc=2): # wie interface(verboseproc=3): # mehr Infos
ans31:=dsolve(deq3 union init3,y(t),numeric,method=classical[rk3],
              output=procedurelist);

ans31 := proc(x_classical) option Copyright (c) 2000 by Waterloo Maple Inc. All rights reserved.; ... end proc

> ans32:=dsolve(deq3 union init3,y(t),numeric,method=classical[rk3],
               output=listprocedure);

ans32 := [t = proc(t) option Copyright (c) 2000 by Waterloo Maple Inc. All rights reserved.; ... end proc,
          y(t) = proc(t) ... end proc,  $\frac{d}{dt} y(t) = \text{proc}(t) \dots \text{end proc}$ ,  $\frac{d^2}{dt^2} y(t) = \text{proc}(t) \dots \text{end proc}$ ,
           $\frac{d^3}{dt^3} y(t) = \text{proc}(t) \dots \text{end proc}$ ]

> # Rueckstellen
interface(verboseproc=1):
```

2. Möglichkeit: Darstellung der Prozedurerstellung

```
> # Einstellen der Option
infolevel[dsolve]:=1: # wie infolevel[dsolve]:=1: # Kurzversion
infolevel[dsolve]:=2: # etwas mehr Infos
infolevel[dsolve]:=3: # wie infolevel[dsolve]:=4,5,... # mehr Infos

> ans33:=dsolve(deq3 union init3,y(t),numeric,method=classical[rk3],
               output=procedurelist);
> ans34:=dsolve(deq3 union init3,y(t),numeric,method=classical[rk3],
               output=listprocedure);

> # Rueckstellen
infolevel[dsolve]:=1:
```

Der erste Teil der Information in ans33 und ans34 ist der gleiche.

```
dsolve/numeric: entering dsolve/numeric
DEtools/convertsys: converted to first-order system Y'(x)=f(x,Y(x)) namely
(with Y' represented by YP)
```

$$[YP_1 = Y_2, YP_2 = Y_3, YP_3 = Y_4, YP_4 = Y_1 t^2 + Y_4]$$

DEtools/convertsys:

correspondence between Y[i] names and original functions:

$$\left[Y_1 = y(t), Y_2 = \frac{d}{dt} y(t), Y_3 = \frac{d^2}{dt^2} y(t), Y_4 = \frac{d^3}{dt^3} y(t) \right]$$

dsolve/numeric:

the procedure F(x,Y,YP) for computing $Y'(x)=f(x,Y(x))$ is:

```
proc (N, t, Y, YP)
option [Y[1] = y(t),
        Y[2] = diff(y(t),t),
        Y[3] = diff(diff(y(t),t),t),
        Y[4] = diff(diff(diff(y(t),t),t),t)];
YP[1] := Y[2]; YP[2] := Y[3]; YP[3] := Y[4]; YP[4] := Y[1]*t^2+Y[4];
0
end proc
dsolve/numeric: initial conditions: x0=0., y0=[3.56, 12., -4., 6.544]
```

ans33 := proc(x_classical) ... end proc

bzw.

```
ans34 := [ t = proc(t) ... end proc,
           d/dt y(t) = proc(t) ... end proc,
           d^2/dt^2 y(t) = proc(t) ... end proc,
           d^3/dt^3 y(t) = proc(t) ... end proc ]
```

3. Möglichkeit: Kombination beider

Darstellung der Prozedurerstellung und des Prozedurinhaltes

```
> # Einstellen der Option
interface(verboseproc=2):
infolevel[dsolve]:=3:

> ans35:=dsolve(deq3 union init3,y(t),numeric,method=classical[rk3]);

> # Rueckstellen
interface(verboseproc=1):
infolevel[dsolve]:=1:
```

2.2.3 Richtungsfeld und numerische Lösung

Typische Kommandos zur Berechnung des Richtungsfeldes der gDGL mit der numerischen Lösung des AWP sind `DEplot` und `phaseportrait`.

Ihre grobe Charakterisierung lautet folgendermaßen.

- Mit `DEplot` können Richtungsfeld und/oder numerische Lösungen von AWP erzeugt und letztere in unterschiedlicher Form bereitgestellt und ausgewertet werden.
- Mit `phaseportrait` werden Richtungsfeld/Phasenportraits geliefert und numerische Lösungen von AWP erzeugt und zurückgegeben.

Für die numerischen Ergebnisse werden zahlreiche Verfahren mit unterschiedlichen Eigenschaften bereitgestellt. Die Ergebnisse können auch weiterverarbeitet werden.

Man achte stets auch auf die Auswahl des Darstellungsbereiches und die zahlreichen Optionen.

DEtools[DEplot] - plot solutions to a system of DEs

Informationen zu den Calling Sequences und Parameters wurden schon angegeben.

Calling Sequences

```
DEplot(deqns, vars, trange, inits, xrange, yrange, options)
```

Description

- . Given a set or list of initial conditions (see below), and a system of first order differential equations or a single higher order differential equation, `DEplot` plots solution curves, by numerical methods. Note: This means that the initial conditions of the problem must be given in standard form, that is, the function values and all derivatives up to one less than the differential order of the differential equation at the same point.
- . A system of two first order differential equations produces a direction field plot, provided the system is determined to be autonomous. In addition, a single first order differential equation produces a direction field (as it can always be mapped to a system of two first order autonomous differential equations). A system is determined to be autonomous when all terms and factors, other than the differential, are free of the independent variable. For more information, see `DEtools[autonomous]`. For systems not meeting these criteria, no direction field is produced (only solution curves are possible in such instances). There can be only one independent variable.
- . The default method of integration is `method=rkf45`. Other methods can be specified in the optional equations. Note that because numerical methods are used to generate plots, the output is subject to the characteristics of the numerical method in use. In particular, unusual output may occur when dealing with asymptotes.
- . The direction field presented consists of either a grid of arrows or a set of randomly generated arrows. In either case, the arrows are tangential to solution curves. For each grid point, the arrow centered at (x,y) has slope dy/dx . This slope is computed using $(dy/dt)/(dx/dt)$, where these two derivatives are specified in the first argument to `DEplot`. The curved arrow types (`curves` and `comet`) require additional data

for the curvature of the direction field, which is computed by moving an epsilon in the direction of the slope dy/dx , and computing dy/dx , then moving an epsilon in the direction opposite the slope, and computing dy/dx . This data is then sufficient to draw a small portion of the direction field lines local to the point, which is then used to draw the curved arrows.

- . By default, the two dependent variables are plotted, unless otherwise specified in the scene option.
- . The `deqns` parameter can be given as a procedure, but must conform to the specification as given in `dsolve/numeric`, and the number option must be included before the initial conditions.

In this instance, `deqns` must be of the form:

```
proc(N,ivar,Y,YP)
  ...
  YP[1]:=f1(ivar,Y);
  YP[2]:=f2(ivar,Y);
  ...
end proc
```

where `N` represents the number of first order equations, `ivar` is the independent variable, `Y` is a vector of length `N`, and `YP` is a vector of derivatives which is updated by the procedure (for the equivalent first order system), also of length `N`.

For information on determining first order systems, see `DEtools[convertsys]`. `N` is given in this case by the fourth parameter number.

- . The `inits` parameter must be specified as:


```
[[x(t[0])=x[0], y(t[0])=y[0]], [x(t[1])=x[1], y(t[1])=y[1]],...]
[[y(t[0])=y[0]], [y(t[1])=y[1]],...]
[y(t[0])=y[0], y(t[1])=y[1],...]
```

where, in the above, sets can be used in place of lists, or

```
[[t[0],x[0],y[0]], [t[1],x[1],y[1]],...]
{[t[0],x[0],y[0]], [t[1],x[1],y[1]],...}
[[t[0],x[0]], [t[1],x[1]],...]
```

where the above is a list or set of lists, each sublist specifying one group of initial conditions.

- . The `xrange` and `yrange` parameters must be specified as follows.

```
x(t)=x[1]..x[2], y(t)=y[1]..y[2] or
x=x[1]..x[2], y=y[1]..y[2]
```

By default, integration along a solution curve stops one mesh point after the specified range is exceeded. This may be overridden by the `obsrange` option.

- . `DEplot` can also be implemented through the interactive `DEplot` assistant. For more information, see `DEplot[interactive]`.

Options

- . The optional equations, `eqns`, consist of the following, as well as a restricted set of plot and `dsolve[numeric]` options.

```
'animatecurves'
'animatefield'
'animate'
'arrows'
'color'
'dirfield'
'dirgrid'
'iterations'
'linecolor'
'number'
'numframes'
```

- 'numpoints'
 - 'numsteps'
 - 'obsrange'
 - 'scene'
 - 'size'
 - 'stepsize'
- . animatecurves = true or false (default = false)
 - . animatecurves creates an animation of the evolving solution curve from the specified initial value(s) with respect to time. The default number of animation frames is 25, which can be changed with the numframes option. dsolve[numeric] is used to compute the solution curves for all specified initial values at different times. Note that if numpoints is provided, and the number of intervals (numpoints-1), does not divide evenly by the number of frame intervals (numframes-1), then a warning is produced, and numpoints is modified to be $\text{ceil}((\text{numpoints}-1)/(\text{numframes}-1))*(\text{numframes}-1)+1$.
 - . animatefield = true or false (default = false)
 - . animatefield creates an animation of the direction field with respect to time (that is, the arrows describing the direction field move as a solution with respect to time). The default number of animation frames is 25, which can be changed with the numframes option. dsolve[numeric,rkf45] is used to compute the trajectory of each arrow.
 - . animate = true or false (default = false)
 - . animate is a shortcut that can be used to specify that both animatefield (if applicable) and animatecurves (if applicable) be used.
 - . arrows = arrowtype
 - . arrowtype consists of 'small', 'smalltwo', 'medium', 'mediumfill', 'large', 'curve', 'comet', 'line', or 'none'. The arrowtype indicates the type of arrow to be used in a direction field. The default, when a direction field is possible, is 'small'. Specifying an arrowtype of 'none' suppresses the calculation and display of any direction fields.
 - . color, colour = arrowcolor
arrowcolor can take many forms:
 - 1) a plot[color] name;
 - 2) COLOR('HUE',realcons);
 - 3) COLOR('RGB',realcons,realcons,realcons);
 - 4) the literal 'magnitude';
 - 5) magnitude[plot[color] name, plot[color] name];
 - 6) an expression in two variables;
 - 7) a procedure in two variables;
 - 8) a three-element list of expressions in two variables; or
 - 9) a three-element list of procedures in two variables.
 In case 4), the color of the arrows is determined by the magnitude of the field, where blue is used for the lowest magnitude, and red is used for the highest.
 Case 5) is identical to case 4), but the low and high magnitude colors are specified as the indices of magnitude (for example, magnitude is equivalent to magnitude[blue,red]).
 In cases 6) and 7), coordinates of the arrows in the plane are passed to the expression or procedure. The resulting values are then normalized on the range [0,1] and used as 'HUE' values.
 In cases 8) and 9), the resulting values are normalized on [0,1] and used as RGB values. Note that 6) and 8) must contain expressions of the plotted variables. This form of color handling is useful in differentiating features of the direction field.

The default arrow color is red.

- . `dirfield = [posint,posint]`, `posint`, or `[[float,float],[float,float],...]`
 If `[posint,posint]` is used, it specifies a grid on which to place the arrows. This is identical to the deprecated `dirgrid` option. If `posint` is used, it specifies the number of randomly positioned arrows to use for the direction field. If `[[float,float],[float,float],...]` is used, each sublist describes a point at which an arrow is to be drawn.
- . `dirgrid = [posint,posint]`
 The `dirgrid` option is maintained only for backward compatibility. It corresponds to the `dirfield = [posint, posint]` option.
- . `iterations = posint`
`iterations` provides a method of decreasing the step size while leaving the number of stored points constant. For example, setting `stepsize=0.05` and `iterations=5` computes internally with a step size of 0.01 but only stores every 5th point. This option is deprecated by `numpoints`, and is ignored by methods except the classical methods.
- . `linecolor = line_info`
`line_info` can take one of five forms:
 - 1) a `plot[color]` name;
 - 2) `COLOR('HUE',realcons)`;
 - 3) `COLOR('RGB',realcons,realcons,realcons)`;
 - 4) an expression of the independent variable; or
 - 5) a procedure of the independent variable.
 In cases 4) and 5), values of the independent variable are passed to the expression or procedure. The resulting values are then normalized on `[0,1]` and applied as 'HUE' values. This color feature is useful in displaying how the solution curve varies with the independent variable. In addition, forms 1) to 5) can be combined into a list, one element for each of the solution curves considered. Each list element is then applied to a corresponding solution curve.
 The default line color is yellow.
- . `number = posint`
`number` specifies the number of equations in the procedure-form first order system in `dproc` (see calling sequences above).
- . `numframes = posint`
`numframes` specifies the number of frames used for animations, using any of the `animate`, `animatefield`, or `animatecurves` options.
- . `numpoints = posint`
`numpoints` specifies the number of points used to plot the curves over the specified time range. The default number of points is 49. Note that for animations, `numpoints-1` must be an integer multiple of `numframes-1`.
- . `numsteps = posint`
`numsteps` allows specification of the `stepsize` parameter by a more convenient means. For `trange=a..b`, specifying `numsteps=n` computes with a step size of $(b-a)/n$. This option is independent of `numpoints`. Note that this option only applies to classical fixed step size methods, which are not the default.
- . `obsrange = boolean`
`obsrange` indicates whether the integrator should stop once the solution curve has passed outside the specified range. It also specifies that any arrows that pass completely outside the viewing area should be removed (for an arrow-based animation). This is useful in the plotting of functions with asymptotic behavior. The default is `true`.
- . `scene = [name,name]`

scene specifies the axes of the plot to be viewed.
 For example, scene=[x,y] indicates that the plot of x versus y (x horizontal) is to be plotted, with t implicit, while scene=[t,y] plots t versus y with t explicit. This option can also be used to change the order in which to plot the variables.
 There is no default ordering when vars is indicated as a set; if vars is given as a list, the given ordering is used.

- . size = float or magnitude
 When size=magnitude, the size of the arrows is proportional to the magnitude of the direction field. When size is specified as a positive floating point value, then this value is used to modify the size of all arrows in the direction field.
 The default value is size=1.0.
- . stepsize = realcons
 stepsize specifies the step size used by dsolve when computing the curves in the plot using the classical methods. For trange=a..b, the default value is |b-a|/48.
- . Note that the solution curves are plotted with a default line thickness of 3, but this can be changed with the thickness plot option.
- . There are many other optional equations that can be specified, for example, the thickness option. For more information, see plot[options] and the individual dsolve[numeric] help pages.

Wir kehren zum AWP (2.1) zurück.

```
> pts:={ [0,1], [0,1/2], [0,3/2] };
> p1:=DEplot(dgl=0,y(x),x=-2..2,pts,color=black,
             linecolor=[black,green,blue]): # dabei ist thickness=3
             display(p1);
> p2:=DEplot(dgl,y(x),x=-2..2,pts,color=black,
             linecolor=[black,green,blue],axes=frame,labels=['',''],thickness=1):
             display(p2);
> p3:=subs(THICKNESS(3)=THICKNESS(2),p1):
             # p3:=subs(thickness(3)=thickness(2),p1): # Grossschreibung notwendig
             display(p3);
```

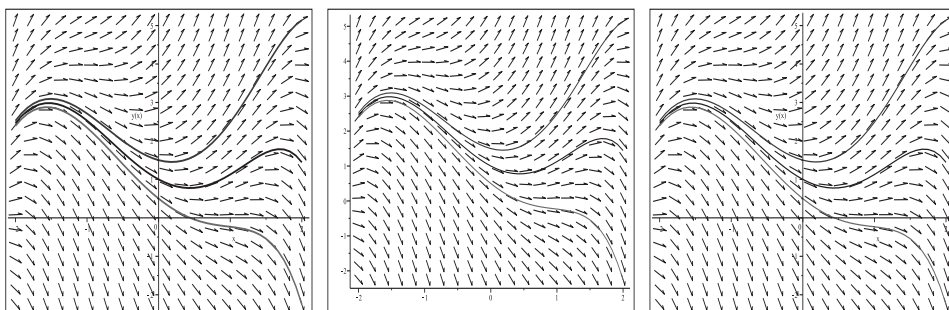


Abb. 2.12 Dateien *richt_field_14.16.ps* (jeweils zu p11, p12, p13),
 numerische Lösung zum AWP $y' - y + x^3 - 3x + 2 = 0$, $x \in [-2, 2]$,
 mit den AB $y(0) = 1, \frac{1}{2}, \frac{3}{2}$, Kommando DEplot

Für autonome SysgDGL 1. Ordnung mit zwei gDGL kann man ebenfalls Richtungsfeld und/oder numerische Lösungen erzeugen.

`dfieldplot` bzw. `DEplot`

Given either a system of two first order autonomous differential equations, or a single first order differential equation, `dfieldplot` produces a direction field plot. For non-autonomous systems, no direction field will be produced. There can be only one independent variable.

 A system of two first order differential equations produces a direction field plot, provided the system is determined to be autonomous. In addition, a single first order differential equation produces a direction field (as it can always be mapped to a system of two first order autonomous differential equations).

Das Testbeispiel (2.1) wandeln wir mittels der Substitution $\mathbf{y} = (y_1, y_2) = (x, y)$ um in das SysgDGL

$$\begin{aligned} y_1' &= 1, & y_1(0) &= 0, \\ y_2' &= y_2 - y_1^3 + 3y_1 - 2, & y_2(0) &= 1. \end{aligned} \quad (2.3)$$

Die Maple-Kommandos sind dann die folgenden.

```
> dgls:= [diff(y1(t),t)=1,diff(y2(t),t)=y2(t)-y1(t)^3+3*y1(t)-2);
  fcns:= [y1(t),y2(t)];
> p11:=dfieldplot(dgls,fcns,t=-2..2,y1=-2..2,y2=-2..5,color=black,
  arrows=slim,dirgrid=[30,30]): # dirfield=[30,30]
  display(p11);
> p12:=DEplot(dgls,fcns,t=-2..2,y1=-2..2,y2=-2..5,color=black,arrows=small):
  display(p12);
> ptss:={ [0,0,1], [0,0,1/2], [0,0,3/2] };
  p13:=DEplot(dgls,fcns,t=-2..2,y1=-2..2,y2=-2..5,ptss,
  color=black,arrows=small,linecolor=[black,green,blue]):
  display(p13);
```

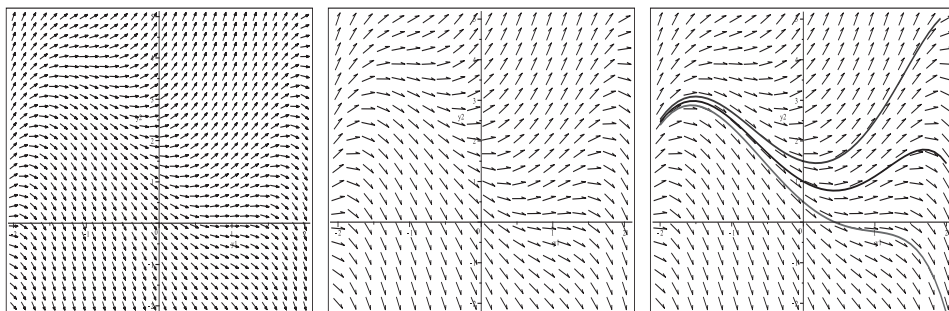


Abb. 2.13 Dateien *richt_field_17..19.ps* (jeweils zu `p11`, `p12`, `p13`), Richtungsfeld bzw. numerische Lösung zum AWP $y_1' = 1$, $y_2' = y_2 - y_1^3 + 3y_1 - 2$, $t \in [-2, 2]$, mit den AB $(y_1, y_2)(0) = (0, 1)$, $(y_1, y_2)(0) = (0, \frac{1}{2})$, $(y_1, y_2)(0) = (0, \frac{3}{2})$, Kommandos `dfieldplot`, `DEplot`

DEtools[phaseportrait] - plot solutions to a system of DEs

Calling Sequences

```
phaseportrait(deqns,vars,trange,inits,options)
```

Parameters

```
deqns  - list or set of first order ordinary differential equations,
        or a single differential equation of any order
vars    - dependent variable, or list or set of dependent variables
trange  - range of the independent variable
inits   - initial conditions for solution curves
options - (optional) equations of the form keyword=value
```

Description

- . Given a list (or set) of initial conditions (see below), and a system of first order differential equations or a single higher order differential equation, `phaseportrait` plots solution curves, by numerical methods. Note: This means that the initial conditions of the problem must be given in standard form, that is, the function values and all derivatives up to one less than the differential order of the differential equation at the same point.
- . A system of two first order differential equations also produces a direction field plot, provided the system is determined to be autonomous. In addition, a single first order differential equation also produces a direction field (as it can always be mapped to a system of two first order autonomous differential equations). For systems not meeting these criteria, no direction field is produced (only solution curves are possible in such instances). There can be ONLY one independent variable.
- . All of the properties and options available in `phaseportrait` are also found in `DEplot`. For more information, see `?DEplot`.
- . `inits` should be specified as
`[[x(t0)=x0, y(t0)=y0], [x(t1)=x1, y(t1)=y1], ()..()]`
 where the above is a list (or set) of lists, each sublist specifying one group of initial conditions.

Das Kommando `phaseportrait` dient i. Allg. der Darstellung des Richtungsfelds der gDGL mit numerischen Lösungen von AWP. AB müssen gegeben sein.

Es gibt eine Vielzahl der Optionen, die sich aus den Kommandos `dsolve/numeric` und `DEplot` ableiten. Insbesondere findet passend zum Kommando `phaseportrait` die Option `scene` für unterschiedliche Phasenkurven bei gDGL und SysgDGL Verwendung.

Die vertikale Ausdehnung der Richtungsfeldes richtet sich bei gDGL mit AB nach dem y -Intervall der numerischen Lösungen. Mit der `view`-Option kann sie nicht "verkleinert" werden. Sind die Intervalle in `view` zu groß, entsteht um das Richtungsfeld eine "leerer Rand".

```
> # wenn keine AB, dann Fehler
phaseportrait(dgl=0,y(x),x=-2..2);
```

Error, (in DEtools/phaseportrait) expecting at least 4 arguments but got 3

```

> phaseportrait(dgl=0,y(x),x=-2..2,[[y(0)=1],[y(0)=1/2],[y(0)=3/2]],
  linecolor=[black,green,blue]); # Farbe red fuer Richtungsfeld
> pts:=[[0,1],[0,1/2],[0,3/2]]:
> pl1:=phaseportrait(dgl=0,y(x),x=-2..2,pts,
  color=black,linecolor=[black,green,blue],dirgrid=[30,30]):
  pl2:=pointplot(pts,color=[black,green,blue],
  symbol=solidcircle,symbolsize=16):
  display(pl1,pl2);

> pl3:=phaseportrait(dgl=0,y(x),x=-2..2,pts,
  method=rkf45, # RKF-Verfahren, default
  scene=[x,y(x)], # Szene-Auswahl, default
  # color=-x-(x^2+4*y),
  # color=[red,blue],
  color=[gray,black], # flach->gray, steil->black
  linecolor=[black,green,blue],dirgrid=[40,40]):
  display(pl2,pl3);

```

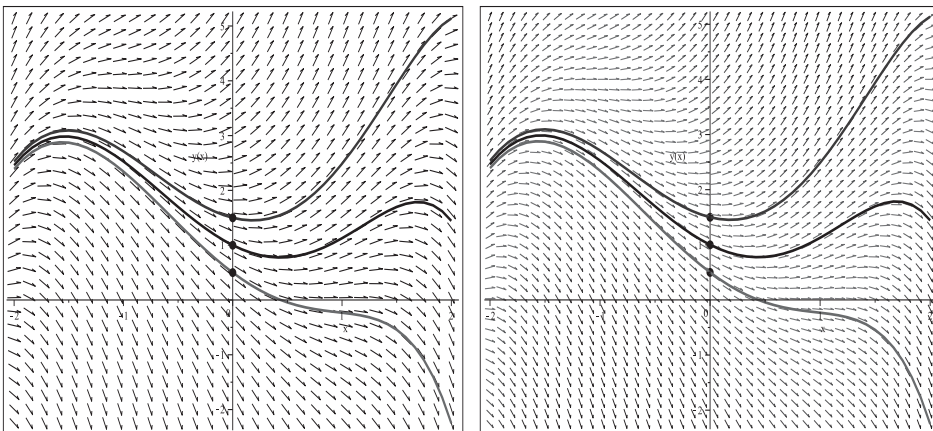


Abb. 2.14 Dateien *richt_field_08.ps* (zu $p11, p12$), *richt_field_09.ps* (zu $p12, p13$), Richtungsfeld zur gDGL $y' - y + x^3 - 3x + 2 = 0$ in $[-2, 2] \times [-2, 5]$, mit numerischen Lösungen bei den AB $y(0) = \frac{1}{2}, 1, \frac{3}{2}$, Kommando `phaseportrait`

Für das SysgDGL (2.3) zum Testbeispiel (2.1) notieren wir folgende Maple-Kommandos.

```

> dgls:=[diff(y1(t),t)=1,diff(y2(t),t)=y2(t)-y1(t)^3+3*y1(t)-2];
  fcns:=[y1(t),y2(t)];
> pl1:=phaseportrait(dgls,fcns,t=-2..2,[[0,0,1]],color=black,
  # [[y1(0)=0,y2(0)=1]]
  linecolor=black,arrows=small,dirgrid=[30,30]):
  display(pl1);
> pl2:=phaseportrait(dgls,fcns,t=-2..2,[[y1(0)=0,y2(0)=1]],color=black,
  linecolor=black,arrows=small,dirgrid=[30,30],view=[-2..2,-1..4]):
  display(pl2);

```

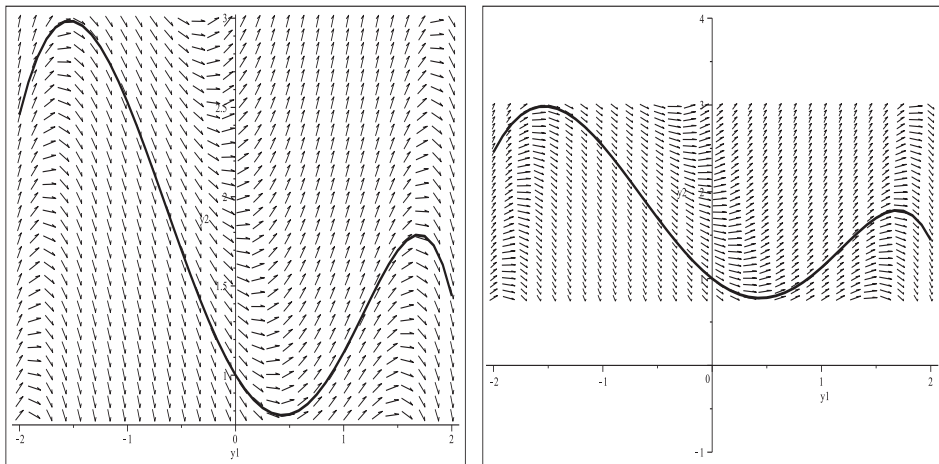


Abb. 2.15 Dateien *richt_field_20.ps* (zu p11), *richt_field_21.ps* (zu p12),
Richtungsfeld und numerische Lösung zum AWP
 $y_1' = 1$, $y_2' = y_2 - y_1^3 + 3y_1 - 2$, $t \in [-2, 2]$, mit der AB $(y_1, y_2)(0) = (0, 1)$,
Kommando `phaseportrait`

SysgDGl mit drei gDGl (SysgDGl (2.3) um eine gDGl erweitert)

```
> pts3:={[0,0,1,1],[0,0,1/2,1],[0,0,3/2,1]}:
dgl3:=[diff(y1(t),t)=1,
      diff(y2(t),t)=y2(t)-y1(t)^3+3*y1(t)-2,
      diff(y3(t),t)=y3(t)-y2(t)+y1(t)]:
fcns3:=[y1(t),y2(t),y3(t)]:
> # Fehler
phaseportrait(dgl3,fcns3,t=-2..2,[[y1(0)=0,y2(0)=1,y3(0)=1]],color=black,
linecolor=black,dirgrid=[30,30]);
```

Error, (in DEtools/phaseportrait)
More than two dependent variables - please indicate the desired scene.

```
> # System von 3 gDGl, kein Richtungsfeld, nur Kurven mittels scene
pl31:=phaseportrait(dgl3,fcns3,t=-2..2,[[y1(0)=0,y2(0)=1,y3(0)=1]],
linecolor=black,
scene=[y1(t),y2(t)]:
# scene=[y1(t),y3(t)]:
# scene=[y2(t),y3(t)]:
pl31a:=pointplot([0,1],color=black,symbol=solidcircle,symbolsize=25):
display(pl31,pl31a);
# analog fuer andere Szenen
... display(pl32,pl32a);
... display(pl33,pl33a);
```

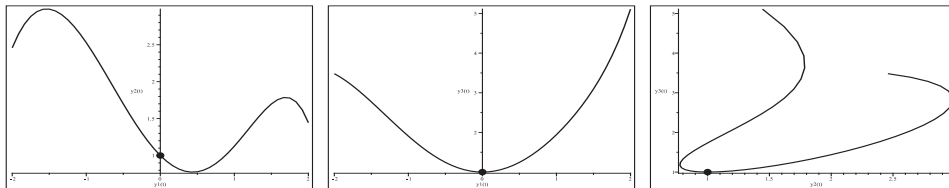


Abb. 2.16 Dateien *richt_field_22..24.ps* (jeweils zu p13k, p13ka, k=1, 2, 3), numerische Lösung zum AWP $y'_1 = 1$, $y'_2 = y_2 - y_1^3 + 3y_1 - 2$, $y'_3 = y_3 - y_2 + y_1$, $t \in [-2, 2]$, mit der AB $(y_1, y_2, y_3)(0) = (0, 1, 1)$, Kommando `phaseportrait` mit `scene=[y1(t), y2(t)], [y1(t), y3(t)], [y2(t), y3(t)]`

Zu weiteren gDGl zeigen wir ausgewählte Eigenschaften und Ergebnisse. Dabei werden sowohl die gDGl als auch AWP mit verschiedenen AB betrachtet und wie vorher in geeigneter Weise die Kommandos

`dsolve`, `fieldplot`, `dfieldplot`, `DEplot`, `odeplot`, `phaseportrait` einbezogen.

Beispiel 2.4

Für das erste AWP werden die Maple-Kommandos ausführlicher notiert.

Damit wird eine sichere Vorgehensweise gezeigt. Aber nicht immer lassen sich Lösungen und Richtungsfelder so einfach wie in Teil (1) behandeln.

(1) $y' = \sin(y)$, $y(0) = 1$ (vergl. Abschnitt 1.8.1 Nr. 69)

$$y(x) = \arctan\left(\frac{2e^x \sin(1)(1 - \cos(1))}{\sin(1)^2 + (e^x)^2 (2 - 2\cos(1) - \sin(1)^2)}, \frac{\sin(1)^2 - (e^x)^2 (2 - 2\cos(1) - \sin(1)^2)}{\sin(1)^2 + (e^x)^2 (2 - 2\cos(1) - \sin(1)^2)}\right)$$

Das Richtungsfeld ist 2π -periodisch bezüglich y . Es gibt die konstanten Trajektorien $y(x) = k\pi$, k ganzzahlig, wobei für ungerade k die stationären Lösungen (Gleichgewichtslagen) anziehend und damit stabil sind.

```
> dgl:=diff(y(x),x)-sin(y(x));
dsolve({dgl=0},y(x));
dsolve(dgl=0,y(x));
f:=rhs(%);
# Kontrolle der Richtigkeit der Loesung y(x) durch Einsetzen in die DGL
subs(y(x)=f,dgl); simplify(%);

> # AWP
dsolve({dgl=0,y(0)=1},y(x));
f1:=rhs(%);
# Kontrolle der Richtigkeit der Loesung y(x) durch Einsetzen
simplify(subs(y(x)=f1,dgl));
# Kontrolle der Richtigkeit der AB y(0)=1 durch Einsetzen
simplify(subs(x=0,f1));
```

```

> # Andere AB
f2:=rhs(dsolve({dgl=0,y(-1)=4},y(x)));
f3:=rhs(dsolve({dgl=0,y(0)=3/2},y(x)));
f4:=rhs(dsolve({dgl=0,y(0)=1/2},y(x)));

> # Richtungsfeld [1,y'] der Loesung
v:=[1,sin(y)];
fieldplot(v,x=-2..2,y=-2..5,color=black,thickness=1);
> # Richtungsfeld [1,y'] mit Loesungen von AWP
v:=[1,sin(y)];
pl1:=fieldplot(v,x=-2..2,y=-2..5,color=black):
pl2:=plot([f1,f2,f3,f4],x=-2..2,color=[black,green,blue,magenta],
          thickness=2):
pl3:=pointplot([[0,1],[-1,4],[0,3/2],[0,1/2]],
               color=[black,green,blue,magenta],symbol=solidcircle,symbolsize=16):
display(pl1,pl2,pl3);

> # Richtungsfeld der Loesung
dfieldplot({dgl=0},[y],x=-2..2,y=-2..5,color=black);
> # Richtungsfeld mit sehr guten numerischen Loesungen
pts:[[0,1],[-1,4],[0,3/2],[0,1/2]];
pl1:=DEplot({dgl=0},[y(x)],x=-2..2,y=-2..5,pts,color=black,
            linecolor=[black,green,blue,magenta],dirgrid=[21,15]):
# dabei ist thickness=3
pl1a:=plot([0,Pi],-2..2,color=black,thickness=2):
pl1b:=pointplot([[0,1],[-1,4],[0,3/2],[0,1/2]],color=black,
                symbol=solidcircle,symbolsize=16):
display(pl1,pl1a,pl1b);

> pl2:=subs(THICKNESS(3)=THICKNESS(1),pl1): pl2;
# nicht pl2:=subs(thickness(3)=thickness(1),pl1): # Grossschreibung

> datei:='D:/temp/dar_loes_01.ps':
interface(plotdevice=ps,plotoutput=datei,
          plotoptions='color,portrait,width=640,height=480');
display(pl1,pl1a,pl1b);
interface(plotdevice=default);

```

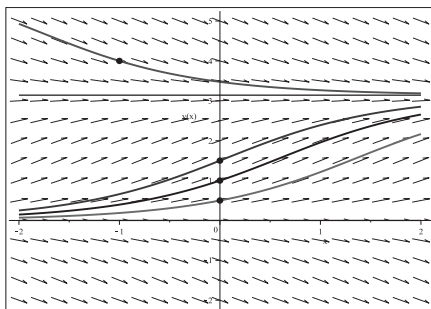


Abb. 2.17

Datei *dar_loes_01.ps*,
 Richtungsfeld zur gDGL des AWP (1),
 Näherungslösungen mit den AB
 $y(0) = 1, \frac{1}{2}, \frac{3}{2}, y(-1) = 4$,
 stationäre Lösungen $y(x) = 0, \pi$,
 Kommando `DEplot`

(2) $y' = 3y - 5x$, $y(1) = 2$ (vergl. Abschnitt 1.8.1 Nr. 78)

$$y(x) = \frac{5}{9} + \frac{5}{3}x - \frac{2}{9}e^{3(x-1)}$$

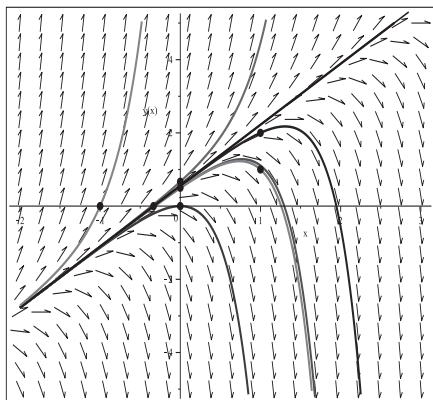


Abb. 2.18

Datei *dar_Loes_02.ps*,

Richtungsfeld zur gDGL des AWP (2),

Näherungslösungen mit den AB

$y(1) = 2$, $y(0) = 0$, $\frac{1}{2}$, $\frac{2}{3}$, $y(-\frac{1}{3}) = 0$, $y(-1) = 0$,

Kommando `DEplot`

(3) $y' = -y - 2x^3 + x^2$, $y(0) = 2$

$$y(x) = 14 - 14x + 7x^2 - 2x^3 - 12e^{-x}$$

> # Richtungsfeld mit sehr guten numerischen Loesungen

```
pts:=[[0,2],[0,0],[0,1],[0,3],[0,4],[0,10],[0,15],[0,-1],[0,-3]];
```

```
> p11:=DEplot({dgl=0},[y(x)],x=-2..3,y=-7..17,pts,color=black,
```

```
linecolor=[green,black,brown,yellow,magenta,blue,cyan,pink,gray],
```

```
stepsize=0.01): # Schrittweite beachten
```

```
p11a:=plot(14-14*x+7*x^2-2*x^3,x=-2..3,color=black,thickness=2,
```

```
linestyle=dash): # Grenzverlauf
```

```
p11b:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
```

```
display(p11,p11a,p11b);
```

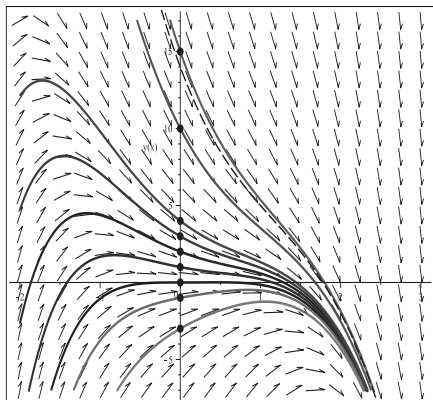


Abb. 2.19

Datei *dar_Loes_03.ps*,

Richtungsfeld zur gDGL des AWP (3),

Näherungslösungen mit den AB

$y(0) = 2, 0, 1, 3, 4, 10, 15, -1, -3$,

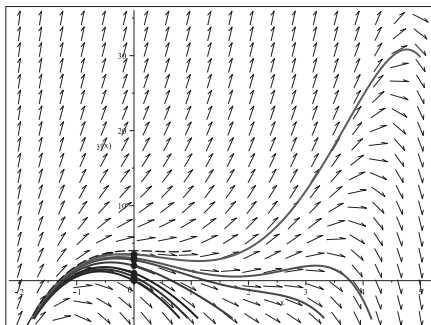
Grenzverlauf

$y(x) = 14 - 14x + 7x^2 - 2x^3$, $x \gg 0$,

Kommando `DEplot`

$$(4) \quad y' = y - x^3 + 4x^2 - 2x - 4, \quad y(0) = 1$$

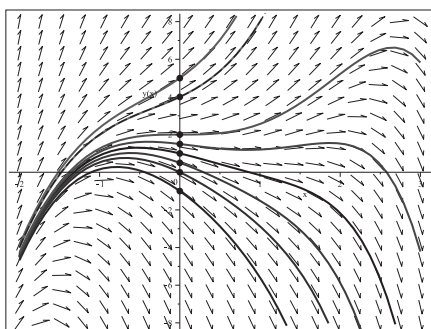
$$y(x) = 4 - x^2 + x^3 - 3e^x$$

**Abb. 2.20**

Datei *dar_Loes_04.ps*,
 Richtungsfeld zur gDGl des AWP (4),
 Näherungslösungen mit den AB
 $y(0) = 1, y(0) = 0, \frac{1}{2}, 2, \frac{11}{4}, 3, \frac{7}{2}$,
 Grenzverlauf $y(x) = 4 - x^2 + x^3, x \ll 0$,
 Kommando `DEplot`

$$(5) \quad y' = y - x^3 + 2x^2 - 2, \quad y(0) = 1$$

$$y(x) = 4 + 2x + x^2 + x^3 - 3e^x$$

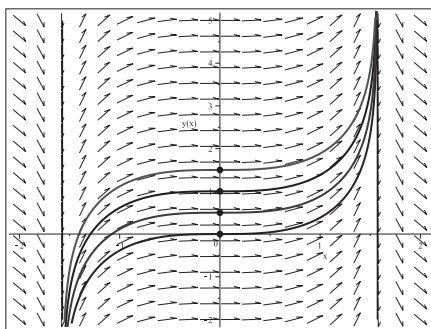
**Abb. 2.21**

Datei *dar_Loes_05.ps*,
 Richtungsfeld zur gDGl des AWP (5),
 Näherungslösungen mit den AB
 $y(0) = 1, y(0) = 1, -1, 0, \frac{1}{2}, \frac{3}{2}, 2, 4, 5$,
 Grenzverlauf $y(x) = 4 + 2x + x^2 + x^3, x \ll 0$,
 Kommando `DEplot`

$$(6) \quad y' = \sin(x) \tan(x), \quad y(0) = 1$$

$$y(x) = -\sin(x) + \ln(\sec(x) + \tan(x)) + 1 = -\sin(x) + \ln(1/\cos(x) + \tan(x)) + 1$$

Es wird in `DEplot` nicht über die Polstellen $\pm \frac{\pi}{2}$ von $\tan(x)$ hinweg integriert.

**Abb. 2.22**

Datei *dar_Loes_06.ps*,
 Richtungsfeld zur gDGl des AWP (6),
 Näherungslösungen mit den AB
 $y(0) = 1, 0, \frac{1}{2}, \frac{3}{2}$,
 Kommando `DEplot`

(7) $y' = e^y \sin^2(x) \cos(x)$, $y(0) = 1$ (vergl. Abschnitt 1.8.1 Nr. 79)

$$y(x) = 1 + \ln(12) - \ln(e \sin(3x) - 3e \sin(x) + 12)$$

Das Richtungsfeld ist 2π -periodisch bzgl. x . An den Stellen $x = k\frac{\pi}{2}$, k ganzzahlig, sind die Anstiege 0. Aber bei $y \gg 0$ können wegen e^y die Linienelemente fast vertikal werden. Es wird also auch Lösungen von AWP geben, die sehr steil verlaufen und Polstellen (Asymptoten) besitzen. Wir verweisen deshalb auf einige Besonderheiten der Lösung. Mit der AB $y(0) = y_0$ ist die Lösung

$$y(x) = y_0 + \ln(12) - \ln(e^{y_0}(\sin(3x) - 3\sin(x)) + 12).$$

Sie ist 2π -periodisch, hat aber für gewisse AB auch Polstellen, über die dann nicht hinweg integriert werden kann. Dazu untersucht man die Nullstellen der Argumentfunktion

$$g(x) = e^{y_0}(\sin(3x) - 3\sin(x)) + 12.$$

Für die AB $y(0) = \ln(3)$ erhält man die Grenzsituation mit dem Punkt $\frac{\pi}{2}$ und dazu $g(\frac{\pi}{2}) = 0$, der zugleich ein lokales Minimum darstellt. Für Anfangswerte $y(0) > \ln(3)$ ist $g(x) \leq 0$ in einem beschränkten Intervall $[\frac{\pi}{2} - \delta, \frac{\pi}{2} + \delta]$, sodass dort der reelle Logarithmus nicht auswertbar ist. Für die AB $y(0) = \frac{3}{2}$ ist das Intervall ungefähr $[1.06, 2.07]$.

```
> # Funktionen 2Pi-periodisch
q11:=plot([1,exp(1)*(sin(3*x)-3*sin(x))+12,
          exp(1)^ln(3)*(sin(3*x)-3*sin(x))+12,
          exp(1)^(1.5)*(sin(3*x)-3*sin(x))+12],x=-6..3,numpoints=500,
          color=[black,black,magenta,brown],
          thickness=[1,2,2,2],view=[-6..3,-6..30]);
display(q11);
> q12:=plot([ln(exp(1)*(sin(3*x)-3*sin(x))+12),
          ln(exp(1)^(ln(3))*(sin(3*x)-3*sin(x))+12),
          ln(exp(1)^(1.5)*(sin(3*x)-3*sin(x))+12)],x=-6..3,numpoints=500,
          color=[black,magenta,brown],thickness=2,view=[-6..3,-4..4]);:
display(q12);
```

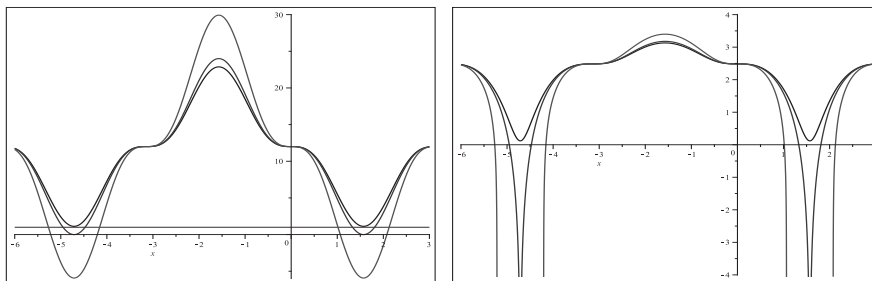


Abb. 2.23 Dateien *dar_Loes_07a.ps*, *dar_Loes_07b.ps*,
Argumentfunktion $g(x)$ bzw. rechts $\ln(g(x))$ zur Lösung des AWP (7)
mit den AB $y(0) = 1, \ln(3), \frac{3}{2}$

DEplot kann zu den AB $y(0) = \ln(3), \frac{3}{2}$ nur Näherungslösungen bis zu den Polstellen erzeugen. Bei grober Schrittweite erfolgt sogar eine Warnung.

Warning, plot may be incomplete, the following error(s) were issued:
cannot evaluate the solution further right of 1.5704030 resp. 1.0649585,
probably a singularity

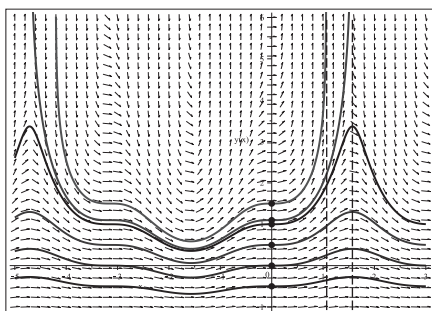


Abb. 2.24

Datei *dar_Loes_07.ps*,
Richtungsfeld zur gDGL des AWP (7),
Näherungslösungen mit den AB
 $y(0) = 1, -\frac{1}{2}, 0, \frac{1}{2}, \ln(3), \frac{3}{2}$,
Kommando DEplot

(8) $y' = y - |x|, y(0) = 1$

$$y(x) = \begin{cases} -x - 1 + 2e^x, & \text{falls } x \leq 0, \\ x + 1, & \text{falls } x > 0 \end{cases}$$

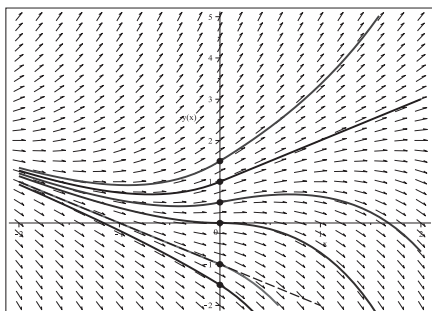


Abb. 2.25

Datei *dar_Loes_08.ps*,
Richtungsfeld zur gDGL des AWP (8),
Näherungslösungen mit den AB
 $y(0) = 1, -\frac{3}{2}, -1, 0, \frac{1}{2}, \frac{3}{2}$,
Grenzverlauf $y(x) = -1 - x, x \ll 0$,
Kommando DEplot

(9) $y' = y/2 + e^{x/2}, y(0) = 1$

$$y(x) = 2e^{-x/2} - e^{x/2} = -2 \sinh(x/2) + e^{-x/2}$$

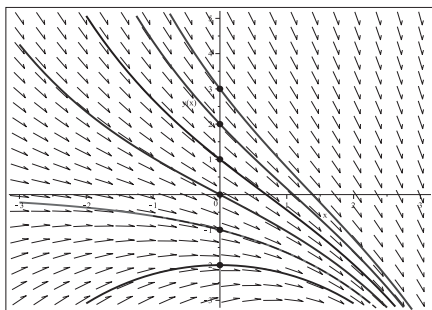


Abb. 2.26

Datei *dar_Loes_09.ps*,
Richtungsfeld zur gDGL des AWP (9),
Näherungslösungen mit den AB
 $y(0) = 1, -2, -1, 0, 2, 3$,
Grenzverlauf $y(x) = -e^{x/2}, x \gg 0$,
Kommando DEplot

$$(10) \quad y' = -y - x + 5, \quad y(0) = 1$$

$$y(x) = 6 - x - 5e^{-x}$$

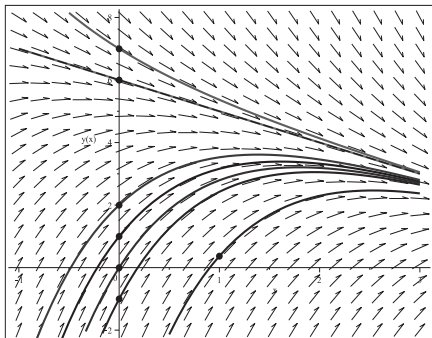


Abb. 2.27

Datei *dar_Joes_10.ps*,

Richtungsfeld zur gDGI des AWP (10),

Näherungslösungen mit den AB

$$y(0) = 1, -1, 0, 2, 6, 7, \quad y(1) = e^{-1},$$

Grenzverlauf $y(x) = 6 - x$, $x \gg 0$,

Kommando `DEplot`

$$(11) \quad y' = -2xy + 2x^3, \quad y(0) = 1 \quad (\text{vergl. Abschnitt 1.8.1 Nr. 31})$$

$$y(x) = -1 + x^2 + 2e^{-x^2}$$

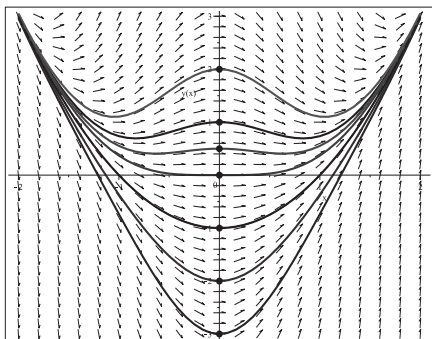


Abb. 2.28

Datei *dar_Joes_11.ps*,

Richtungsfeld zur gDGI des AWP (11),

Näherungslösungen mit den AB

$$y(0) = 1, -3, -2, -1, 0, \frac{1}{2}, 2,$$

Grenzverlauf $y(x) = -1 + x^2$, $|x| \gg 0$,

Kommando `DEplot`

$$(12) \quad y' = -y(x-1), \quad y(0) = 1, 0$$

$$y(x) = e^{-x(x-2)/2} \quad \text{bzw.} \quad y(x) = 0$$

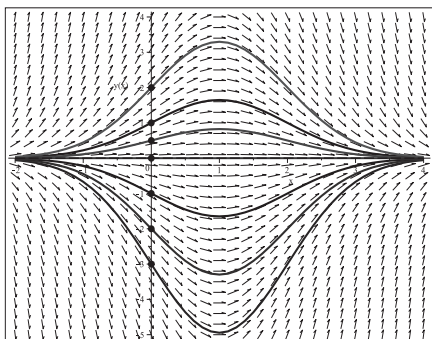


Abb. 2.29

Datei *dar_Joes_12.ps*,

Richtungsfeld zur gDGI des AWP (12),

Näherungslösungen mit den AB

$$y(0) = 1, -3, -2, -1, 0, \frac{1}{2}, 2,$$

Grenzverlauf $y(x) = 0$, $|x| \gg 0$,

Kommando `DEplot`

$$(13) \quad y' = y - \sin(x) - 2, \quad y(0) = 1$$

$$y(x) = 2 + (\sin(x) + \cos(x))/2 - 3e^x/2$$

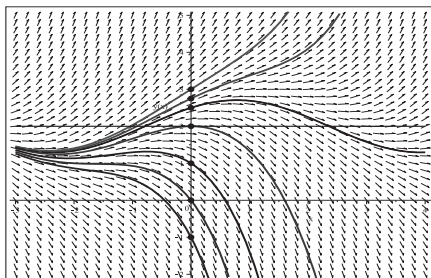


Abb. 2.30

Datei *dar_Loes.13.ps*,

Richtungsfeld zur gDGL des AWP (13),

Näherungslösungen mit den AB

$$y(0) = 1, -1, 0, 2, \frac{5}{2}, \frac{11}{4}, 3,$$

Grenzverlauf

$$y(x) = 2 + \frac{1}{2}(\sin(x) + \cos(x)), \quad x \ll 0,$$

Kommando `DEplot`

$$(14) \quad y' = y - x^4 \sin(x), \quad y(0) = 1$$

$$y(x) = (x^4/2 + 2x^3 + 3x^2 - 3) \cos(x) + (x^4/2 - 3x^2 - 6x - 3) \sin(x) + 4e^x$$

```
> dgl:=diff(y(x),x)-y(x)+x^4*sin(x);
f4:=rhs(dsolve({dgl=0,y(0)=-3},y(x)));
# f4=(x^4/2+2x^3+3x^2-3)cos(x)+(x^4/2-3x^2-6x-3)sin(x), ohne Glied e^x
pts:=[[0,1],[0,-5],[0,-4],[0,-3],[0,-2],[0,-1],[0,0]];

> p11:=DEplot({dgl=0},[y(x)],x=-4..6,y=-400..50,pts,color=black,
  stepsize=0.01,linecolor=[black,green,red,blue,black,magenta,brown],
  dirgrid=[41,37]):
p11a:=plot(f4,x=-4..6,color=black,linestyle=dash,thickness=2):
p11b:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p11a,p11b);

> p12:=DEplot({dgl=0},[y(x)],x=-4..6,y=-15..5,pts,color=black,
  stepsize=0.01,linecolor=[black,green,red,blue,black,magenta,brown],
  dirgrid=[41,37]):
display(p12,p11a,p11b);
```

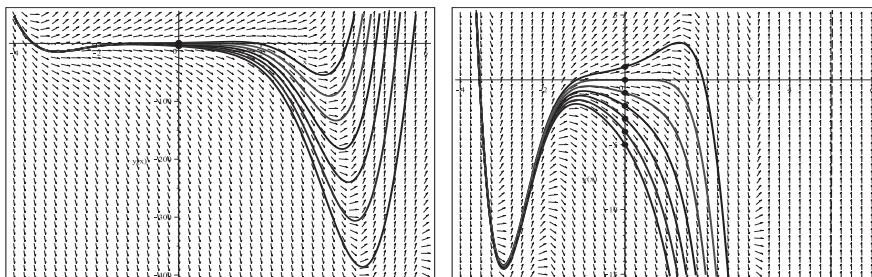


Abb. 2.31 Dateien *dar_Loes.14.ps*, *dar_Loes.14a.ps*, Richtungsfeld zur gDGL (14), Näherungslösungen mit den AB $y(0) = 1, -5, -4, -3, -2, -1, 0$, Grenzverlauf $y(x) = (\frac{x^4}{2} + 2x^3 + 3x^2 - 3) \cos(x) + (\frac{x^4}{2} - 3x^2 - 6x - 3) \sin(x)$, $x \ll 0$, Kommando `DEplot`, `method=rkf45`

$$(15) \quad y' = 2xy^2, \quad y(0) = 1$$

$$y(x) = 1/(1 - x^2), \quad x \in (-1, 1)$$

Es gilt $y(x) = 0$ zur AB $y(0) = 0$. Für AB $y(0) = y_0 < 0$ sind die Lösungskurven überall definiert. Es wird in `DEplot` bei der AB $y(0) = y_0 > 0$ nicht über die Polstellen $\pm\sqrt{1/y_0}$ der Lösung $y(x) = 1/(1/y_0 - x^2)$ hinweg integriert. Genauso gibt es Polstellen mit den AB $y(x_0) = y_0 < 0$ bei $x_0 \neq 0$ und y_0 hinreichend klein.

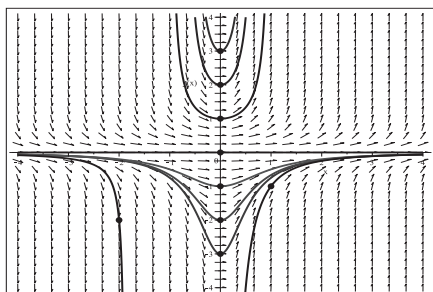


Abb. 2.32

Datei `dar_Joes_15.ps`,

Richtungsfeld zur gDGI des AWP (15),

Näherungslösungen mit den AB

$$y(0) = -3(1)3, \quad y(1) = -1, \quad y(-2) = -2,$$

Asymptoten für $y(0) = y_0 > 0$ bei $\pm\sqrt{1/y_0}$,

Kommando `DEplot`

$$(16) \quad y' = y/(2x) + 1, \quad y(1) = 1$$

$$y(x) = 2x - \sqrt{x}, \quad x \in [0, \infty)$$

Die Untersuchungen für diese gDGI erfolgen etwas ausführlicher.

Man erkennt, dass $x = 0$ eine kritische Stelle ist, wo die gDGI nicht definiert ist und die Richtungselemente vertikal verlaufen. Das hat unterschiedliche Auswirkungen.

Mit der AB $y(0) = y_0 \neq 0$ findet `dsolve` keine Lösung, jedoch wird bei $y(0) = 0$ das allgemeine parameterabhängige Ergebnis $y(x, c) = 2x + c\sqrt{x}$ geliefert. Damit zeigt sich die Mehrdeutigkeit der Lösung. So passt die Lösung $2x - \sqrt{x}$ sowohl zur AB $y(1) = 1$ als auch zur AB $y(0) = 0$, zur letzten AB gibt es auch die einfache Lösung $2x$.

Zur AB $y(-1) = y_0$ ist die Lösung $y(x) = 2x + (2 + y_0)\sqrt{-x}$, $x < 0$.

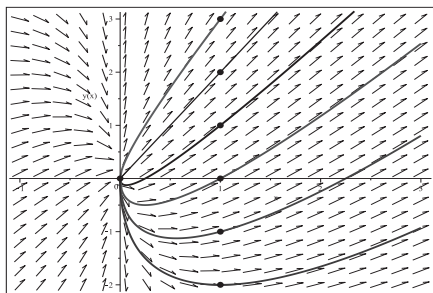


Abb. 2.33

Datei `dar_Joes_16a.ps`,

Richtungsfeld zur gDGI des AWP (16),

Lösungen mit den AB

$$y(1) = 1, \quad -2, \quad -1, \quad 0, \quad 2, \quad 3,$$

Kommandos `dsolve`, `dfieldplot`
`plot`, `pointplot`

Mit dem Kommando `DEplot` erhält man ganz verschiedene Ergebnisse.

Dabei beschränken wir uns auf die numerischen Verfahren `method=classical[rk4]` und `rkf45` sowie eine entsprechende Wahl von Integrationsbereich und Parametern.

Die Singularität an der Stelle 0 bewirkt z.B. Folgendes. Ein Näherungsverfahren kann mit der AB $y(0) = y_0 \neq 0$ nicht starten, aber bei $y(0) = 0$ liefert es eine Näherung zu $y(x) = 2x$. Manchmal wird über Stelle $x = 0$ hinweg integriert (abhängig noch von der Schrittweite) und trotz der Warnung

Warning, plot may be incomplete, the following errors(s) were issued:
division by zero.

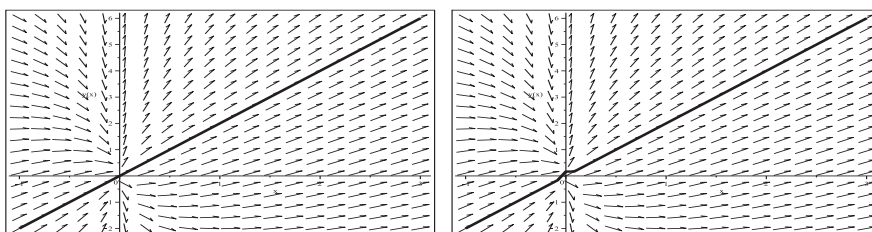


Abb. 2.34 Dateien *dar_Joes_16b.ps*, *dar_Joes_16c.ps*, Richtungsfeld zur gDGl (16), Näherungslösungen mit den AB $y(0) = 0$ und $y(1) = 2$ (rechts) zu $y = 2x$, Kommando `DEplot, method=classical[rk4]`

In Maple rechnen wir nun das AWP mit den AB $y(1) = 2$ sowie $y(-1) = 4.5e8$ (die Lösung ist $y(x) = 2x + 450000002\sqrt{-x}$) bei einer Schrittweite `stepsize=0.01`.

Maple 14

```
> p11:=DEplot({dgl=0}, [y(x)], x=-1..3, [[1,2]], color=black,
#p11:=DEplot({dgl=0}, [y(x)], x=-1..3, y=0..4.5e8, [[1,2]], # analog
color=black, method=classical[rk4], stepsize=0.01,
linecolor=black):
p11:=subs(THICKNESS(3)=THICKNESS(5), p11);
> # aehnlich
> p12:=DEplot({dgl=0}, [y(x)], x=-1..3, y=0..4.5e8, [[-1,450000000]], ...
> p13:=DEplot({dgl=0}, [y(x)], x=-1..3, [[-1,450000000]], ...
```

Es wird nicht über die singuläre Stelle $x = 0$ hinweg integriert.

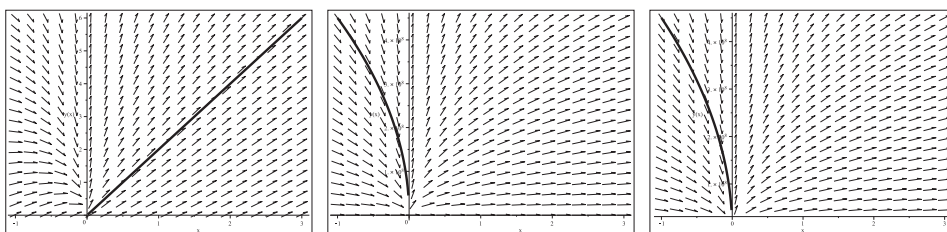


Abb. 2.35 Dateien *dar_Joes_16d.ps*, *dar_Joes_16e.ps*, *dar_Joes_16f.ps*, Richtungsfeld zur gDGl (16), Näherungslösungen p11 mit der AB $y(1) = 2$ sowie p12/p13 zu $y(-1) = 4.5e8$ mit/ohne y -Bereich, Maple 14, Kommando `DEplot, method=classical[rk4], stepsize=0.01`

Anders in Maple 12 mit denselben Befehlen.

Von der AB $y(1) = 2$ wird die numerische Lösung auch nach links fortgesetzt und springt für $x < 0$ auf einen Lösungsweig zur AB $y(-1) = 4.5e8$.

Von der AB $y(-1) = 4.5e8$ wird die numerische Lösung auch nach rechts fortgesetzt und springt für $x > 0$ auf einen extrem hohen Lösungsweig zur AB $y(1) \approx 5.5e32$, der bei der Bereichswahl $y=0..4.5e8$ (mittlere Abb.) gar nicht angezeigt wird.

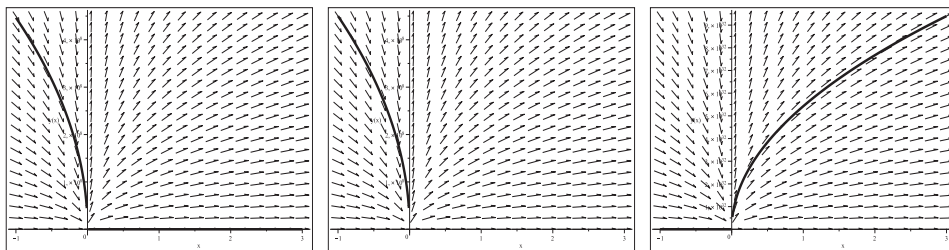


Abb. 2.36 Dateien *dar_Joes_16d1.ps*, *dar_Joes_16e1.ps*, *dar_Joes_16f1.ps*,
Richtungsfeld zur gDGL (16),
Näherungslösungen mit den AB $y(1) = 2$ bzw. $y(-1) = 4.5e8$,
Maple 12, Kommando `DEplot, method=classical[rk4], stepsize=0.01`

Wir ergänzen noch Ergebnisse einiger Testvarianten in Maple 14, die viele Kombinationen zwischen Auswahl von Integrationsbereich, numerischen Verfahren, Schrittweite, AB u.a. zulassen. Dazu sind noch die Maßstäbe in den Abbildungen zu beachten. Wenn der y -Bereich in Zehnerpotenzen geht, dann werden moderate Funktionsverläufe fast auf der x -Achse liegen.

Die numerischen Lösungen der Verfahren `classical[rk4]` und `rkf45` mit den AB $y(0) = 0$ bzw. $y(1) = 2$ sind Approximationen der Lösung $y(x) = 2x$.

(a) Test, wie extrem der “numerische Sprung“ bei $x = 0$ wird.

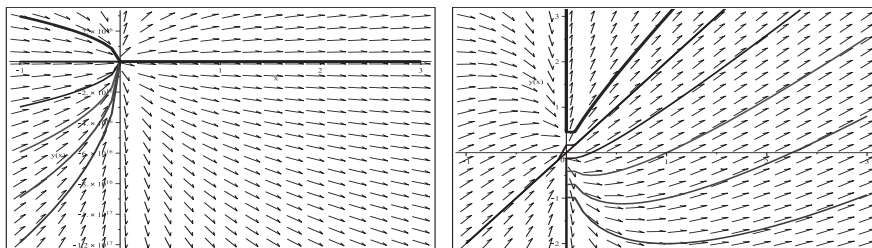


Abb. 2.37 Dateien *dar_Joes_16g1..2.ps*, Richtungsfeld zur gDGL (16),
Näherungslösungen mit den AB $y(1) = 1, -2, -1, 0, 2, 3$ und $y(0) = 0$,
rechts Zoom mittels Bereich $y=-2..3$ an der Singularität $x = 0$,
Kommando `DEplot, method=classical[rk4]`


```

> # AB
pts:=[[1,1],[1,-2],[1,-1],[1,0],[1,2],[1,3],[0,0]];
> p11a:=DEplot({dgl=0},[y(x)],x=-1..3,pts,color=black,
method=classical[rk4],
linecolor=[black,green,red,blue,black,magenta,black]):
p11a;
> p11b:=DEplot({dgl=0},[y(x)],x=-1..3,y=-2..3,pts,color=black,
method=classical[rk4],
linecolor=[black,green,red,blue,black,magenta,black]):
p11b;

```

(b) Test zur Veränderung des Integrationsbereichs $x \in [x_u, 3]$, $x_u = -0.6(-0.1) - 1.4$

```

> xu:=-0.6: # -0.6,-0.7,...,-1,...,-1.4
p11:=DEplot({dgl=0},[y(x)],x=xu..3,pts,color=black,
method=classical[rk4],
linecolor=[black,green,red,blue,black,magenta,black]):
p11;

```

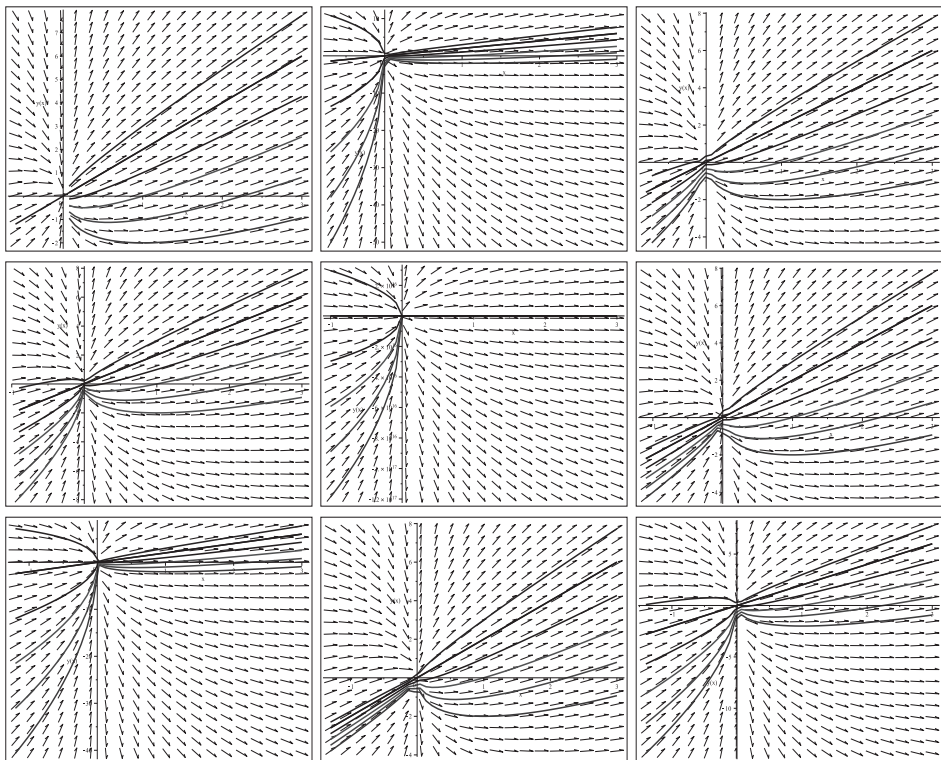


Abb. 2.38 Dateien *dar_Joes_16h1..9.ps*, Richtungsfeld zur gDGl (16), Näherungslösungen mit den AB $y(1) = 1, -2, -1, 0, 2, 3$ und $y(0) = 0$, Komm. `DEplot,x=xu..3,method=classical[rk4],xu=-0.6(-0.1)-1.4`

(c) Veränderung der Schrittweite

```
> p11:=DEplot({dgl=0}, [y(x)], x=-1..3, pts, color=black,
  method=classical[rk4], stepsize=0.01,
  linecolor=[black,green,red,blue,black,magenta,black]):
p11;
```

Mit einer anderen Schrittweite wird nach dem “Sprung“ über die singuläre Stelle $x = 0$ i. Allg. ein anderer Lösungszweig verfolgt, deshalb auch der Unterschied zur mittleren Grafik in der Abb. 2.38 (da ist $x=-1..3$).

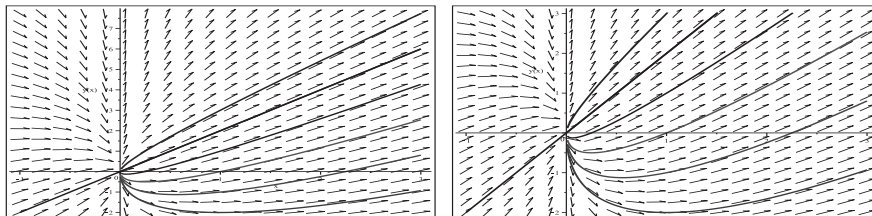
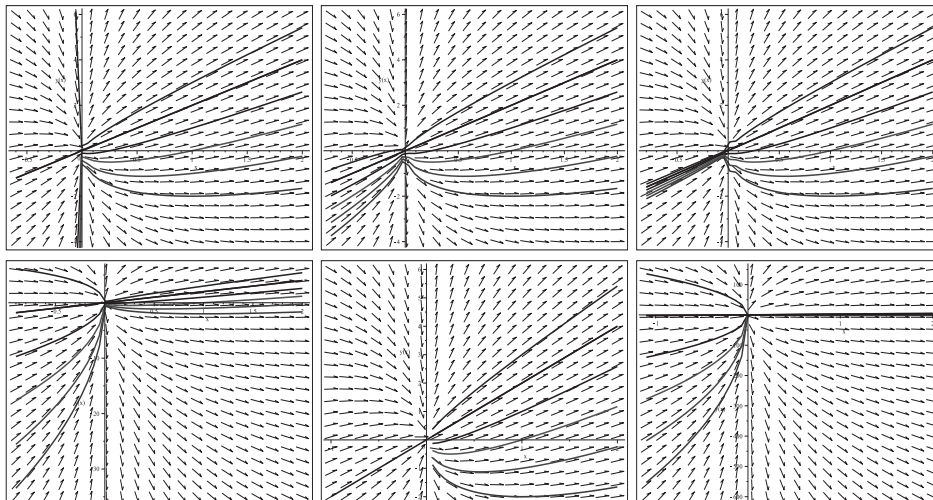


Abb. 2.39 Dateien *dar_Loes_16i1..2.ps*, Richtungsfeld zur gDGl (16),
Näherungslösungen mit den AB $y(1) = 1, -2, -1, 0, 2, 3$ und $y(0) = 0$,
Kommando `DEplot, method=classical[rk4], stepsize=0.01`

(d) Veränderung des Integrationsbereichs $x \in [x_u, 2]$, $x_u = -0.6(-0.1) - 1.4$

Man beachte den Einfluss der Änderung der oberen Grenze 2 des x -Bereichs und vergleiche die Grafiken mit der Abbildung 2.38, wo $x \in [x_u, 3]$ ist.

```
> xu:=-0.6: # -0.6,-0.7,...,-1,...,-1.4
p11:=DEplot({dgl=0}, [y(x)], x=xu..2, pts, color=black, method=classical[rk4],
  linecolor=[black,green,red,blue,black,magenta,black]):
p11;
```



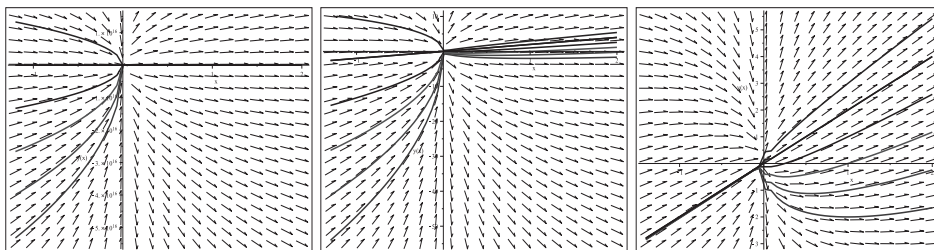


Abb. 2.40 Dateien *dar_Joes_16j1..9.ps*, Richtungsfeld zur gDGL (16),
 Näherungslösungen mit den AB $y(1) = 1, -2, -1, 0, 2, 3$ und $y(0) = 0$,
 Kommando `DEplot, x=xu..2,method=classical[rk4]`,
`xu=-0.6(-0.1)-1.4`

(e) Veränderung der numerischen Methode

Die Standardmethode in `DEplot` ist `rkf45`. Die Wahl des x -Bereichs hat dabei keinen großen Einfluss. Die Schrittweite sollte entsprechend klein sein. Die Methode integriert über die Stelle $x = 0$ hinweg und erzeugt auf der anderen Seite “vernünftige“ Lösungszweige.

```
> # aehnlich
#p11:=DEplot({dgl=0}, [y(x)], x=-1.5(0.1)-0.5..3, pts, color=black,
p11:=DEplot({dgl=0}, [y(x)], x=-1..3, pts, color=black, stepsize=0.01,
  linecolor=[black,green,red,blue,black,magenta,black]):
p11;
> # Zoom
p11:=DEplot({dgl=0}, [y(x)], x=-1..3, y=-2..3, pts, color=black, stepsize=0.01,
  linecolor=[black,green,red,blue,black,magenta,black]):
p11a:=plot(2*x, x=-1..3, color=black, thickness=1):
  # einfachste Loesung bei y(0)=0 bzw. bei y(1)=2
p11b:=pointplot(pts, color=black, symbol=solidcircle, symbolsize=16):
display(p11,p11a,p11b);
```

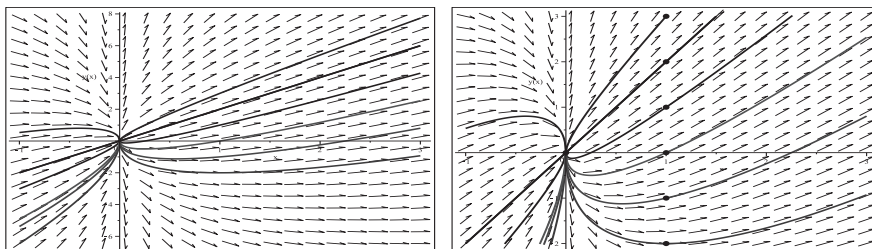


Abb. 2.41 Dateien *dar_Joes_16k1..2.ps*, Richtungsfeld zur gDGL (16),
 Näherungslösungen mit den AB $y(1) = 1, -2, -1, 0, 2, 3$ und $y(0) = 0$,
 Kommando `DEplot, method=rkf45,stepsize=0.01`

Bzgl. der Maple-Versionen ist mit abweichenden Ergebnissen zu rechnen.

$$(17) \quad y' = y/x + 1, \quad y(1) = 1$$

$$y(x) = x(1 + \ln(x)), \quad x \in (0, \infty)$$

Man sieht, dass $x = 0$ eine kritische Stelle ist.

Wegen $\lim_{x \rightarrow 0} x \ln(x) = 0$ kann man $y(0) = 0$ definieren und damit die Lösung auf das Intervall $[0, \infty)$ fortsetzen.

Der Vergleich zur gDGL (16) zeigt eine Reihe von Ähnlichkeiten.

Mit der AB $y(0) = y_0 \neq 0$ findet `dsolve` keine Lösung, jedoch wird bei $y(0) = 0$ das allgemeine parameterabhängige Ergebnis $y(x, c) = x(c + \ln(x))$ geliefert. Damit zeigt sich die Mehrdeutigkeit der Lösung, wenn man den Punkt $y(0) = 0$ bei den Trajektorien hinzunimmt.

Zur AB $y(-1) = y_0$ ist die Lösung $y(x) = x(\ln(-x) - y_0)$, $x < 0$.

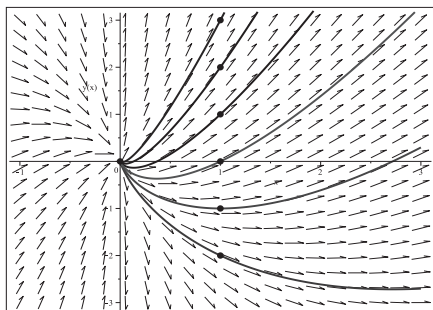


Abb. 2.42

Datei `dar_Loes_17a.ps`,
 Richtungsfeld zur gDGL des AWP (17),
 Lösungen mit den AB
 $y(1) = 1, -2, -1, 0, 2, 3$,
 Kommandos `dsolve`, `dfieldplot`
`plot`, `pointplot`

Mit dem Kommando `DEplot` erhält man ganz verschiedene Ergebnisse.

Dabei beschränken wir uns auf die numerischen Verfahren `method=classical[rk4]` und `rkf45` sowie eine entsprechende Wahl von Integrationsbereich und Parameter.

Die Singularität an der Stelle 0 bewirkt z.B. Folgendes. Ein Näherungsverfahren kann mit der AB $y(0) = y_0$ nicht starten und liefert die Fehlermeldung

```
Error, (in dsolve/numeric/make_proc) ode system is singular
at the initial point.
```

Einige weitere Ergebnisse sind:

(a) `DEplot, method=classical[rk4]`

```
AB [1,0] -> y=x ln(x), x>0
AB [1,y0] -> y=x(ln(x)+y0), x>0
             y=... zumeist stark abfallender negativer Zweig fuer x<0
AB [-1,0] -> y=xln(-x), x<0
AB [-1,y0] -> y=x(ln(-x)-y0), x<0
             y=... zumeist stark zunehmender positiver Zweig fuer x>0
```

Intervallgroesse der Integration ->

- Wachstum der Loesungszweige

- Integration ueber x=0 (falls 0 im Integrationsintervall)

Bei Angabe der Schrittweite `stepsize` wird generell nicht ueber 0 hinweg integriert.

```

> pts1:=[[1,1],[1,-2],[1,-1],[1,0],[1,2],[1,3]];

> p11:=DEplot({dgl=0},[y(x)],x=-2..3,pts1,color=black, # x=-2..2, x=-2..1
method=classical[rk4],
linecolor=[black,green,red,blue,black,magenta]):
p11;

```

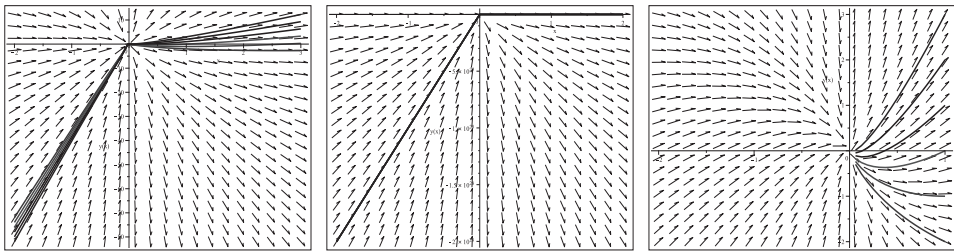


Abb. 2.43 Dateien *dar_Joes_17b1..3.ps*, Richtungsfeld zur gDGL (17),
 Näherungslösungen mit den AB $y(1) = 1, -2, -1, 0, 2, 3$,
 Kommando `DEplot, x=-2..3,2,1,method=classical[rk4]`

```

> pts1m:=[[-1,1],[-1,-2],[-1,-1],[-1,0],[-1,2],[-1,3]];

> p11:=DEplot({dgl=0},[y(x)],x=-2..3,pts1m,color=black, # x=-2..2, x=-2..1
method=classical[rk4],
linecolor=[black,green,red,blue,black,magenta]):
p11;

```

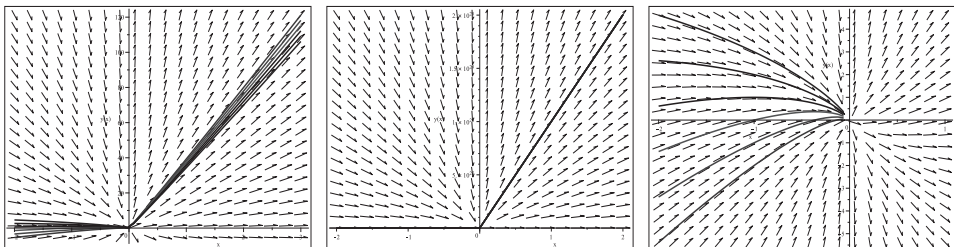


Abb. 2.44 Dateien *dar_Joes_17b4..6.ps*, Richtungsfeld zur gDGL (17),
 Näherungslösungen mit den AB $y(-1) = 1, -2, -1, 0, 2, 3$,
 Kommando `DEplot, x=-2..3,2,1,method=classical[rk4]`

```

> pts1:=[[1,1],[1,-2],[1,-1],[1,0],[1,2],[1,3]];

> p11:=DEplot({dgl=0},[y(x)],x=-2..3,pts1,color=black,
# aehnlich x=-2..2, x=-2..1
#p11:=DEplot({dgl=0},[y(x)],x=-2..3,pts1m,color=black,
method=classical[rk4],stepsize=0.01,
linecolor=[black,green,red,blue,black,magenta]):
p11;

```

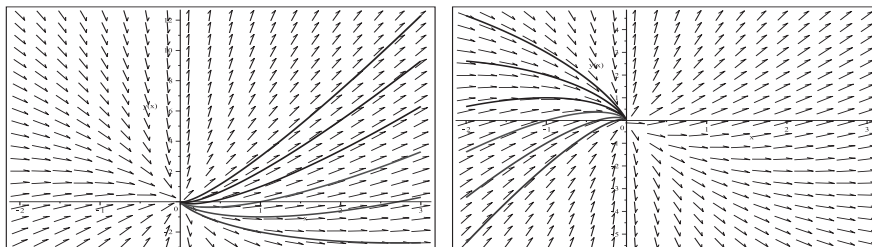


Abb. 2.45 Dateien *dar_Loes_17b7..8.ps*, Richtungsfeld zur gDGI (17),
 Näherungslösungen mit den AB $y(\pm 1) = 1, -2, -1, 0, 2, 3$,
 Kommando `DEplot, x=-2..3,method=rk4`, ähnlich `x=-2..2,x=-2..1`

(b) `DEplot, method=rkf45`

AB $[1, y_0]$ $\rightarrow y = x(\ln(x) + y_0), x > 0$
 $y = \dots$ zumeist moderat abfallender negativer Zweig oder
 zunehmender positiver Zweig fuer $x < 0$

AB $[-1, y_0]$ $\rightarrow y = x(\ln(-x) - y_0), x < 0$
 $y = \dots$ zumeist moderat zunehmender positiver Zweig oder
 abfallender negativer Zweig fuer $x > 0$

Es wird immer ueber 0 hinweg integriert, egal ob ohne/mit Schrittweite
`stepsize`. Das Integrationsintervall ist ohne groesseren Einfluss
 auf das Wachstumsverhalten.

```
> p11:=DEplot({dgl=0}, [y(x)], x=-2..3, [[1,1]], color=black,
# [[1,-1]], [[-1,1]], [[-1,-1]]
method=rkf45, # stepsize=0.01,
linecolor=black):
p11;
```

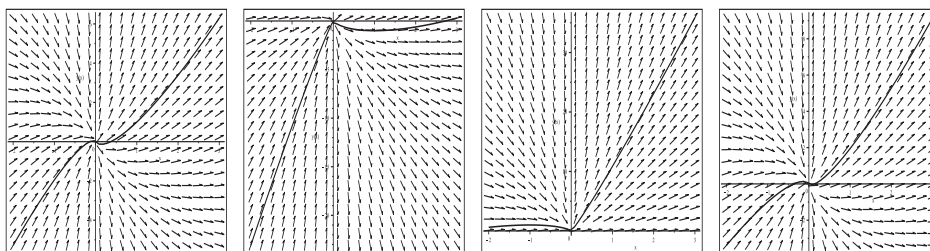


Abb. 2.46 Dateien *dar_Loes_17c1..4.ps*, Richtungsfeld zur gDGI (17),
 Näherungslösungen mit den AB $y(\pm 1) = \pm 1$,
 Kommando `DEplot, x=-2..3,method=rkf45`, ähnlich `x=-2..2,x=-2..1`

(c) Die Lösungen zur AB $y(1) = y_0$ sind $y(x) = x(\ln(x) + y_0)$.

Die gleichen Lösungen erhält man mit den AB $y(0.05) = \tilde{y}_0 = 0.05(\ln(0.05) + y_0)$.

Mit dem Verfahren `rkf45` werden beide AB gerechnet und insbesondere die Fortsetzung
 über die Stelle $x = 0$ nach links beobachtet.

```

> pts1;
pts1a:=[[0.05,0.05*(ln(0.05)+1)],[0.05,0.05*(ln(0.05)-2)],
        [0.05,0.05*(ln(0.05)-1)],[0.05,0.05*(ln(0.05)+0)],
        [0.05,0.05*(ln(0.05)+2)],[0.05,0.05*(ln(0.05)+3)]];
> p11a:=DEplot({dgl=0},[y(x)],x=-2..3,pts1,color=black,
               method=rkf45,
               linecolor=[black,green,red,blue,black,magenta]):
display(p11a,view=[-2..3,-25..15]);
> p11b:=DEplot({dgl=0},[y(x)],x=-2..3,pts1a,color=black,dirgrid=[20,33],
               method=rkf45,
               linecolor=[black,green,red,blue,black,magenta]):
display(p11b,view=[-2..3,-25..15]);
> # Zoom zu p11b
p11c:=DEplot({dgl=0},[y(x)],x=-0.1..0.1,y=-0.25..0.05,pts1a,color=black,
               method=rkf45,stepsize=0.001,
               linecolor=[black,green,red,blue,black,magenta]):
p11c;

```

Die Integration über $x = 0$ hinweg liefert links teilweise andere Lösungszweige.

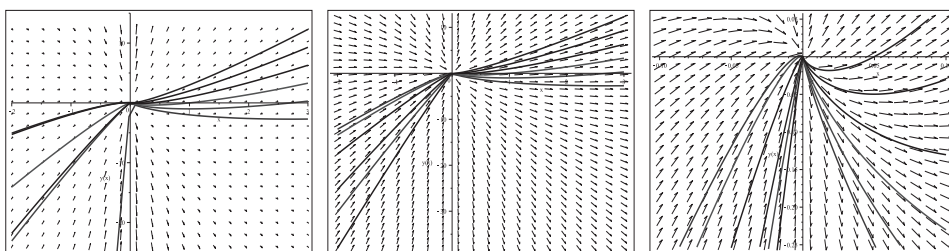


Abb. 2.47 Dateien *dar_Loes_17c5..7.ps*, Richtungsfeld zur gDGl (17),
 links: Näherungslösungen mit den AB $y(1) = 1, -2, -1, 0, 2, 3$,
 Kommando `DEplot, x=-2..3,method=rkf45`,
 Mitte und rechts: Näherungslösungen mit den AB
 $y(0.05) = 0.05(\ln(0.05) + y_0)$, $y_0 = 1, -2, -1, 0, 2, 3$,
 Kommando `DEplot, x=-2..3,method=rkf45`,
 bzw. $x=-0.1..0.1, y=-0.25..0.05$, (Zoom)

(d) Interessant ist, was die numerischen Verfahren machen, wenn sie sich mit ihrer AB $y(x_0) = 0$, $x_0 > 0$, der Bedingung $y(0) = 0$ annähern. Die Folge der AB sei $[[\exp(-1), 0], [\exp(-2), 0], [\exp(-5), 0], [\exp(-10), 0], [\exp(-14), 0], [\exp(-50), 0]]$. Die Lösungen dazu sind entsprechend $y(x) = x(\ln(x) + z)$, $z = 1, 2, 5, 10, 14, 50$. Diese werden zunächst mit den sehr genauen numerischen Lösungen des Verfahrens `rkf45` verglichen. Man bemerkt, dass AB $y(x_0) = 0$ mit $x_0 \leq e^{-14}$ für das Verfahren problematisch werden. Deshalb werden `rkf45` und andere in `DEplot` enthaltene Methoden in Bezug auf das Verhalten bzgl. der AB untersucht.

```

> pts2:=[[1,1],[exp(-1),0],[exp(-2),0],[exp(-5),0],
        [exp(-10),0],[exp(-14),0],[exp(-50),0]];
> p11:=DEplot({dgl=0},[y(x)],x=0..1,y=-0.5..2,pts2,color=black,
             linecolor=[black,blue,green,red,cyan,brown,magenta],
             method=rkf45,stepsize=0.0001):
p11;
# method=classical[rk4],stepsize=0.0001
# method=classical[adambash]
# method=classical[foreuler]
# method=rkf45
# method=ck45
# method=dverk78
# method=gear
# method=rosenbrock

```

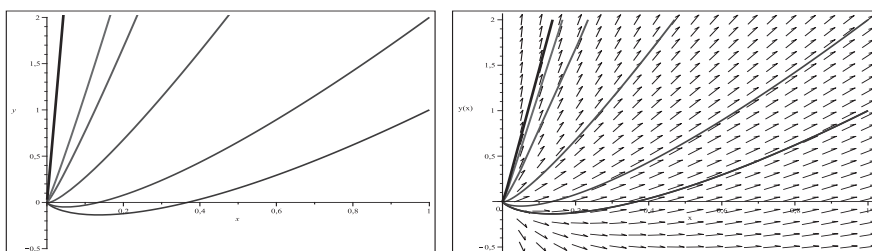


Abb. 2.48 Dateien *dar_Loes_17d1..2.ps*, Richtungsfeld zur gDGL (17),
 links: Lösungen mit den AB $y(x_0) = 0$ mit $x_0 = e^{-z}$, $z = 1, 2, 5, 10, 14, 50$,
 Kommandos `dsolve`, `plot`,
 rechts: Näherungslösungen mit den AB $y(x_0) = 0$,
 Kommando `DEplot`, `x=0..1,method=rkf45,stepsize=1e-4`

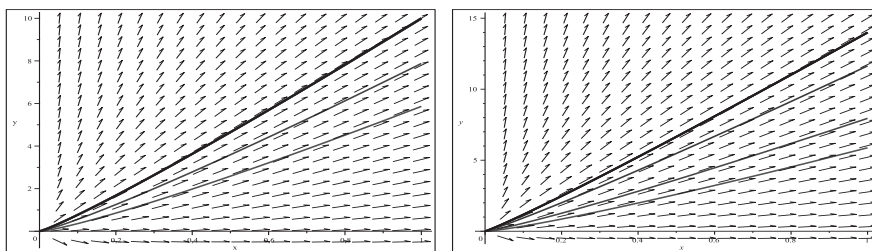


Abb. 2.49 Dateien *dar_Loes_17d3..4.ps*, Richtungsfeld zur gDGL (17),
 links: Näherungslösungen mit den AB $y(e^{-10}) = 0$,
 Kommando `DEplot`, `x=0..1,method = 8 Varianten`,
 exakt \sim `rosenb` \sim `gear` \sim `dverk78` \sim `ck45` \sim `rkf45` \sim `rk4` $>$ `adamb` $>$ `fore`,
 rechts: Näherungslösungen mit den AB $y(e^{-14}) = 0$,
 Kommando `DEplot`, `x=0..1,method = 8 Varianten`,
 exakt \sim `rosenb` \sim `gear` \sim `dverk78` \sim `ck45` \sim `rkf45` $>$ `rk4` $>$ `adamb` $>$ `fore`

Bei der AB $y(e^{-15}) = 0$ beginnen die Probleme. Einige Verfahren liefern falsche Ergebnisse oder versagen. Dazu gehören `gear` beginnend mit der AB $y(e^{-15}) = 0$, `rosenbrock` bei $y(e^{-50}) = 0$ oder `dverk78` bei $y(e^{-70}) = 0$.

All das ist noch abhängig von der Maple-Version, der `Digits`-Einstellung und/oder weiteren Verfahrensparametern.

$$(18) \quad y' = -3y/x + 2/x^3, \quad y(1) = 1$$

$$y(x) = (2x - 1)/x^3, \quad x \in (0, \infty)$$

Man sieht, dass $x = 0$ eine kritische Stelle ist, nämlich eine Polstelle.

Die allgemeine parameterabhängige Lösung ist $y(x, c) = (2x + c)/x^3$. Man hat damit keinen Grenzwert für $x \rightarrow 0$ und eine andere Situation als bei den gDGL (16) und (17). Überraschend ist, dass `dsolve` mit der AB $y(0) = 0, 1, y_0, \dots$ die Lösung $y(x) = 2/x^2$ angibt, die ja im Punkt 0 gar nicht definiert ist. Zur AB $y(x_0) = y_0$ ist die Lösung $y(x) = (2x + x_0(x_0^2 y_0 - 2))/x^3$, sodass bei $x_0 = 1$ sich $y(x) = (2x + y_0 - 2)/x^3$, $x > 0$, sowie bei $x_0 = -1$ sich $y(x) = (2x - y_0 + 2)/x^3$, $x < 0$, ergeben.

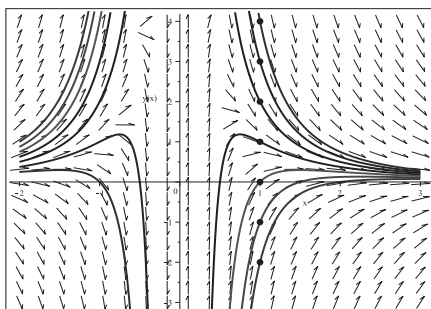


Abb. 2.50

Datei `dar_Loes_18a.ps`,
 Richtungsfeld zur gDGL des AWP (18),
 Lösungen mit den AB
 $y(1) = 1, -2, -1, 0, 2, 3, 4$,
 mit Lösungszweigen für $x < 0$,
 Kommandos `dsolve`, `dfieldplot`
`plot`, `pointplot`

Bei den numerischen Verfahren im Kommando `DEplot` beschränken wir uns auf die Methoden `method=classical[rk4]` und `rkf45` sowie eine entsprechende Wahl von Integrationsbereich und Parameter.

Die Polstelle an der Stelle 0 bewirkt z.B. Folgendes. Ein Näherungsverfahren kann mit der AB $y(0) = y_0$ nicht starten und liefert die Fehlermeldung

```
Error, (in dsolve/numeric/make_proc) ode system is singular
at the initial point.
```

Wenn dieses nahe der Polstelle rechnet, entstehen extrem große Werte. Deshalb ist die Angabe eines y -Bereichs sinnvoll.

Einige weitere Ergebnisse folgen nun.

(a) `DEplot`, `method=classical[rk4]`

```
AB [1,0] -> y=(2x-2)/x^3, x>0
```

```
AB [1,y0] -> y=(2x+y0-2)/x^3, x>0, manchmal Zweig fuer x<0
```

AB $[-1,0]$ $\rightarrow y=(2x+2)/x^3, x<0$
 AB $[-1,y0]$ $\rightarrow y=(2x-y0+2)/x^3, x<0$, manchmal Zweig fuer $x>0$
 Ohne Angabe des y -Bereichs in DEplot \rightarrow
 - grosse y -Werte, damit sehr grosser y -Bereich, Massstab beachten
 - Integration ueber $x=0$ (falls 0 im Integrationsintervall)
 - Abhaengigkeit von Maple Version und Digits-Einstellung
 Bei Angabe der Schrittweite stepsize wird generell nicht ueber 0 hinweg integriert.
 Das Integrationsintervall ohne Angabe des y -Bereichs hat Einfluss auf y -Werte.

```

> pts1:=[[1,1],[1,-2],[1,-1],[1,0],[1,2],[1,3],[1,4]];
> p11:=DEplot({dgl=0},[y(x)],x=-2..3,pts1,color=black, # x=-2..2, x=-2..1
method=classical[rk4],
linecolor=[black,green,red,blue,black,magenta,cyan]):
p11;
  
```

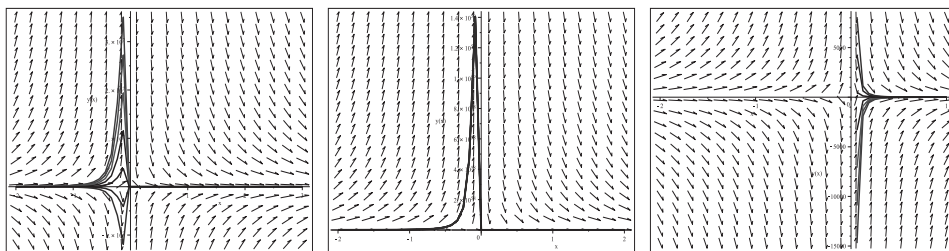


Abb. 2.51 Dateien *dar_Joes_18b1..3.ps*, Richtungsfeld zur gDGl (18),
 Näherungslösungen mit den AB $y(1) = 1, -2, -1, 0, 2, 3, 4$,
 Kommando DEplot, $x=-2..3, 2, 1, method=classical[rk4]$

```

> pts1m:=[[[-1,1],[[-1,-2],[[-1,-1],[[-1,0],[[-1,2],[[-1,3],[[-1,4]]];
> p11:=DEplot({dgl=0},[y(x)],x=-2..3,pts1m,color=black, # x=-2..2, x=-2..1
method=classical[rk4],
linecolor=[black,green,red,blue,black,magenta,cyan]):
p11;
  
```

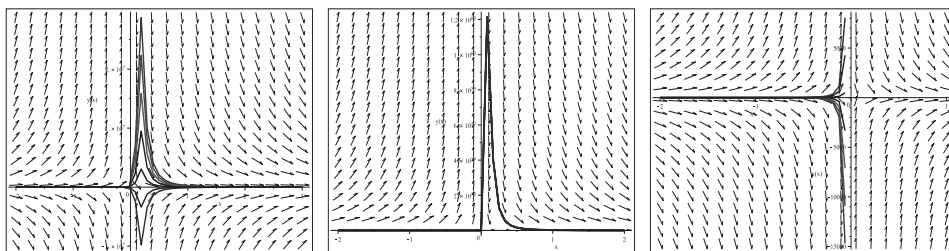


Abb. 2.52 Dateien *dar_Joes_18b4..6.ps*, Richtungsfeld zur gDGl (18),
 Näherungslösungen mit den AB $y(-1) = 1, -2, -1, 0, 2, 3, 4$,
 Kommando DEplot, $x=-2..3, 2, 1, method=classical[rk4]$

```

> pts1:=[[1,1],[1,-2],[1,-1],[1,0],[1,2],[1,3]];
> p11:=DEplot({dgl=0},[y(x)],x=-2..3,y=-3..4,pts1,color=black,
             # aehnlich x=-2..2, x=-2..1
#p11:=DEplot({dgl=0},[y(x)],x=-2..3,y=-3..4,pts1m,color=black,
             method=classical[rk4],stepsize=0.01,
             linecolor=[black,green,red,blue,black,magenta,cyan]):
p11;

```

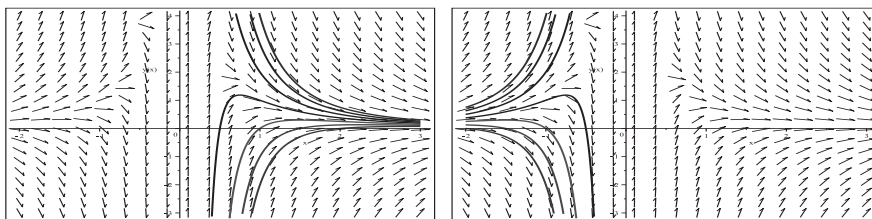


Abb. 2.53 Dateien *dar_Loes_18b7..8.ps*, Richtungsfeld zur gDGL (18),
 Näherungslösungen mit den AB $y(\pm 1) = 1, -2, -1, 0, 2, 3, 4$,
 Kommando `DEplot, x=-2..3,y=-3..4,method=classical[rk4]`,
 ähnlich bei $x=-2..2, x=-2..1$

(b) `DEplot, method=rkf45`

```

AB [1,0]  -> y=(2x-2)/x^3, x>0
AB [1,y0] -> y=(2x+y0-2)/x^3, x>0, kein Zweig fuer x<0
AB [-1,0] -> y=(2x+2)/x^3, x<0
AB [-1,y0] -> y=(2x-y0+2)/x^3, x<0, kein Zweig fuer x>0

```

Es wird nicht ueber 0 hinweg integriert, egal ob ohne/mit Schrittweite
`stepsize`, ohne/mit Angabe des y -Bereichs.

Ev. gibt es grosse y -Werte, wenn kein y -Bereich steht.

Das Integrationsintervall hat Einfluss auf das Verhalten.

```

> p11:=DEplot({dgl=0},[y(x)],x=-2..3,y=-2..2,[[1,1]],color=black,
             # [[1,-1],[[-1,1],[[-1,-1]]
             method=rkf45, # stepsize=0.01,
             linecolor=black):
p11;

```

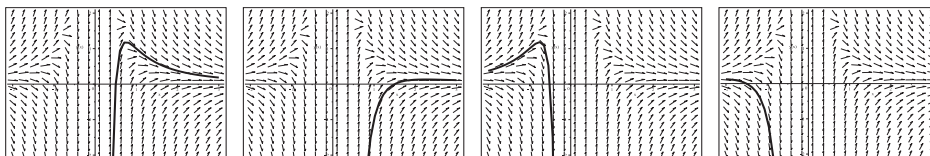


Abb. 2.54 Dateien *dar_Loes_18c1..4.ps*, Richtungsfeld zur gDGL (18),
 Näherungslösungen mit den AB $y(\pm 1) = \pm 1$,
 Kommando `DEplot, x=-2..3,y=-2..2,method=rkf45`,
 ähnlich bei $x=-2..2, x=-2..1$

```
> p11:=DEplot({dgl=0}, [y(x)], x=-2..3, [[1,1]], color=black,
               # [[1,-1]], [[-1,1]], [[-1,-1]]
               method=rkf45, stepsize=0.01, linecolor=black):
p11;
```

Bei den AB $y(\pm 1) = 1$ geht der y -Bereich bis -10^6 , bei $y(\pm 1) = -1$ bis $-3 \cdot 10^6$.

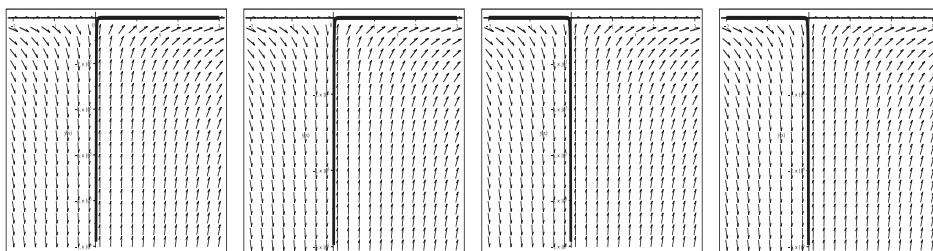


Abb. 2.55 Dateien *dar_Joes_18c5..8.ps*, Richtungsfeld zur gDGL (18), Näherungslösungen mit den AB $y(\pm 1) = \pm 1$, Kommando `DEplot`, `x=-2..3`, `method=rkf45`, `stepsize=0.01`, ähnlich bei `x=-2..2`, `x=-2..1`

$$(19) \quad y' = y/x - y^2, \quad y(1) = 1$$

$$y(x) = 2x/(1+x^2), \quad x \in (-\infty, \infty)$$

Man sieht, dass $x = 0$ eine kritische Stelle ist, was die gDGL betrifft.

Betrachtet man die allgemeine parameterabhängige Lösung $y(x, c) = 2x/(x^2 + 2c)$, so stellt man fest, dass es eine einfache Lösung $y(x) = 2/x$ gibt mit einer Polstelle bei 0 (das ist ein Grenzfall), aber auch welche mit zwei Polstellen symmetrisch zum Koordinatenursprung (da ist $c < 0$) und solche wie oben, die überall definiert und beliebig glatt sind (bei $c > 0$). Das ist abhängig von der AB. Des Weiteren gibt es die Nulllösung, wie man sofort erkennt.

Definiert man am kritischen Punkt $y(0) = 0$, so verschwindet das Problem nicht ganz, denn es bleibt ja noch die Mehrdeutigkeit des AWP.

Zur AB $y(0) = 0$ gibt es z.B. die Lösungen $y(x) = 0$ und $y(x) = 2x/(x^2 + 1)$.

Überraschend ist, dass `dsolve` mit der AB $y(0) = 0$ die vier Lösungen

$$0, \quad \frac{2x}{x^2 + c}, \quad \frac{2x(x^2 \pm \sqrt{-d})}{x^4 + d}$$

erzeugt.

Die beiden letzten korrespondieren mit der zweiten bei der Substitution $-d = c > 0$.

Bei den AB $y(0) = y_0, 1, \dots$ liefert `dsolve` keine Lösung.

Zur AB $y(x_0) = y_0$ ist die Lösung

$$y(x) = \frac{2x y_0}{y_0(x^2 - x_0^2) + 2x_0}$$

Wenn die AB nicht auf der Grenzkurve $2/x$ liegt, dann gilt für die Lösung $y(x)$, dass sie bzw. der innere Lösungszweig durch den Koordinatenursprung verläuft.

Für $x_0 > 0$, $y_0 > 2/x_0$ sind die Polstellen $\pm\sqrt{x_0^2 - 2x_0/y_0}$, sodass für die positive Polstelle $0 < \sqrt{x_0^2 - 2x_0/y_0} < x_0$ gilt.

Die Polstellen im Fall der Lösung $2x/(x^2 - c)$, $c > 0$, liegen bei $\pm\sqrt{c}$.

Die Nulllösung allein erhält man mit der AB $y(x_0) = 0$, $x_0 \neq 0$.

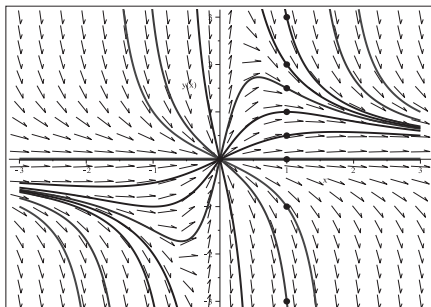


Abb. 2.56

Datei *dar_Loes_19a.ps*,

Richtungsfeld zur gDGL des AWP (19),

Lösungen mit den AB

$y(1) = 1, -3, -1, 0, \frac{1}{2}, \frac{3}{2}, 2, 3$,

mit Lösungszweigen zwischen den Polstellen,

Kommandos `dsolve`, `dfieldplot`

`plot`, `pointplot`

Bei den numerischen Verfahren im Kommando `DEplot` beschränken wir uns auf die Methoden `method=classical[rk4]` und `rkf45` sowie eine entsprechende Wahl von Integrationsbereich und Parameter.

Die Polstelle an der Stelle Null bewirkt z.B. Folgendes. Ein Näherungsverfahren kann mit der AB $y(0) = y_0$ nicht starten und liefert die Fehlermeldung

```
Error, (in dsolve/numeric/make_proc) ode system is singular
at the initial point.
```

Wenn Verfahren nahe der Polstellen rechnen, entstehen extrem große Werte. Deshalb ist die Angabe eines y -Bereichs sinnvoll.

Einige weitere Ergebnisse sind wie folgt.

(a) `DEplot`, `method=classical[rk4]`

```
AB [1,0] -> y=0, x>0
```

```
AB [1,y0] -> y=2xy0/(2+y0(x^2-1)), y0>2,x>sqrt((y0-2)/y0)=rechte Polstelle
           2/x, y0=2,Polstelle bei 0
           2xy0/(2+y0(x^2-1)), y0<2,x>0
```

```
AB [-1,0] -> y=0, x<0
```

```
AB [-1,y0] -> y=2xy0/(2+y0(x^2-1)), y0<-2,x<-sqrt((y0-2)/y0)=linke Polstelle
           2/x, y0=-2,Polstelle bei 0
           2xy0/(2+y0(x^2-1)), y0>-2,x<0
```

I. Allg. wird ueber Polstellen bzw. ueber die 0 nicht hinweg integriert.

Abhaengigkeit von Maple Version und Einstellung von `Digits`, y -Bereich.

Bei Angabe der Schrittweite `stepsize` wird naeher an die Polstellen bzw. an die 0 heran integriert.

Das Integrationsintervall ohne Angabe des y -Bereichs hat Einfluss auf y -Werte.

Ohne stepsize Loesungsweig abhaengig von x-Bereich

- Loesung in $x > 0$, weiter weg von der Null
- hinweg ueber Null, aber einige Zweige sind nicht da oder anders, vielfaeltige Ergebnisse

Mit stepsize nur ein Loesungsweig in $x > 0$, auch nahe an der Null.

```
> pts1:=[[1,1],[1,-3],[1,-1],[1,0],[1,1/2],[1,3/2],[1,2],[1,3]];
> p11:=DEplot({dgl=0},[y(x)],x=-3..3,y=-5..5,pts1,color=black,
               # x=-2..3, x=-1..3, x=-1.6..1.5
               method=classical[rk4],
               linecolor=[black,green,red,blue,black,magenta,black,cyan]):
p11;
```

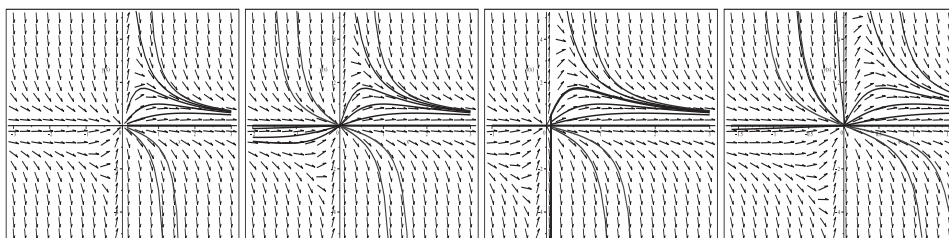


Abb. 2.57 Dateien *dar_Loes_19b1..4.ps*, Richtungsfeld zur gDGl (19),
 Näherungslösungen mit den AB $y(1) = 1, -3, -1, 0, \frac{1}{2}, \frac{3}{2}, 2, 3$,
 Kommando DEplot, $x=-3, -2, -1..3, x=-1.6..1.5, method=rk4$

```
> pts1m:=[[[-1,-1],[[-1,2],[[-1,1],[[-1,0],[[-1,-1/2],[[-1,-3/2],[[-1,-2],[[-1,-3]]];
> p11:=DEplot({dgl=0},[y(x)],x=-3..3,pts1m,color=black,
               # x=-2..3, x=-1..3, x=-1.6..1.5
               method=classical[rk4],
               linecolor=[black,green,red,blue,black,magenta,black,cyan]):
p11;
```

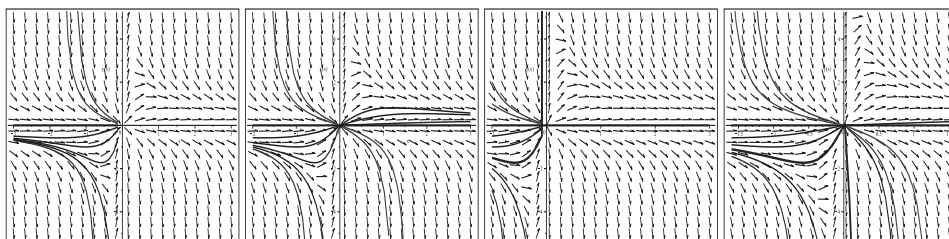


Abb. 2.58 Dateien *dar_Loes_18b5..8.ps*, Richtungsfeld zur gDGl (19),
 Näherungslösungen mit den AB $y(-1) = -1, 2, 1, 0, -\frac{1}{2}, -\frac{3}{2}, -2, -3$,
 Kommando DEplot, $x=-3, -2, -1..3, x=-1.6..1.5, method=rk4$

Man beachte also den deutlichen Einfluss der Größe des x -Bereichs auf die Näherungslösungen und die möglichen Lösungswege.

```

> pts1:=[[1,1],[1,-3],[1,-1],[1,0],[1,1/2],[1,3/2],[1,2],[1,3]];
> p11:=DEplot({dgl=0},{y(x)},x=-3..3,y=-5..5,pts1,color=black,
             # aehnlich x=-2..3, x=-1..3
#p11:=DEplot({dgl=0},{y(x)},x=-3..3,y=-5..5,pts1m,color=black,
             method=classical[rk4],stepsize=0.01,
             linecolor=[black,green,red,blue,black,magenta,black,cyan]):
p11;

```

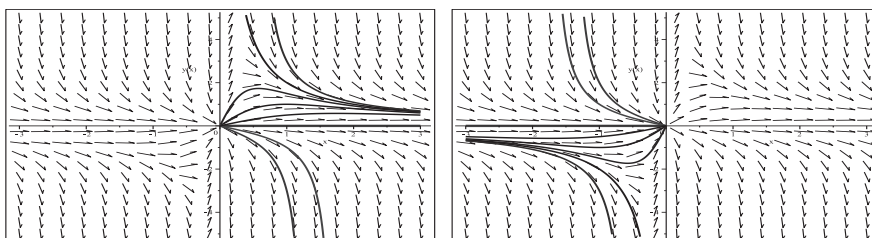


Abb. 2.59 Dateien *dar_Loes_19b91..92.ps*, Richtungsfeld zur gDGL (19),
Näherungslösungen mit den AB

$y(1) = 1, -3, -1, 0, \frac{1}{2}, \frac{3}{2}, 2, 3$ bzw. $y(-1) = -1, 2, 1, 0, -\frac{1}{2}, -\frac{3}{2}, -2, -3$,
Kommando DEplot, $x=-3..3, y=-5..5, method=classical[rk4]$,
ähnlich bei $x=-2..3, x=-1..3$

(b) DEplot, method=rkf45

```

AB [1,0] -> y=0, -infty<x<infty
AB [1,y0] -> y=2xy0/(2+y0(x^2-1)), y0>2,x>sqrt((y0-2)/y0)=rechte Polstelle
              2/x, y0=2,Polstelle bei 0
              2xy0/(2+y0(x^2-1)), y0<2,-infty<x<infty

AB [-1,0] -> y=0, -infty<x<infty
AB [-1,y0] -> y=2xy0/(2+y0(x^2-1)), y0<-2,x<-sqrt((y0-2)/y0)=linke Polstelle
              2/x, y0=-2,Polstelle bei 0
              2xy0/(2+y0(x^2-1)), y0>-2,-infty<x<infty

```

Generell wird ueber Polstellen bei $|y_0|>2$ (2 Polstellen) nicht integriert.
Es wird ueber die kritische Stelle 0 bei $|y_0|<2$ hinweg integriert.
Bei den AB (1,2) bzw. (-1,-2)

```

ohne y-Bereich oder sehr gross in DEplot ->
- Integration ueber x=0 (falls 0 im Integrationsintervall)
- unabhaengig von stepsize
mit y-Bereich in DEplot ->
- keine Integration ueber x=0 (falls 0 im Integrationsintervall)
- unabhaengig von stepsize

```

Abhaengigkeit von Maple Version und Einstellung von Digits und y-Bereich.
Bei Angabe der Schrittweite stepsize wird naeher an die Polstellen bzw.
an die 0 heran integriert.

Das Integrationsintervall ohne Angabe des y-Bereichs hat Einfluss auf y-Werte.
Integration ueber $x=0$ bei der Loesung $2/x$, wenn kein y-Bereich angegeben oder
sehr gross ist.

```
> p11:=DEplot({dgl=0}, [y(x)], x=-3..3, y=-5..5, color=black,
pts1, # pts1m,
method=rkf45, # stepsize=0.01,
linecolor=[black, green, red, blue, black, magenta, black, cyan]):
p11;
```

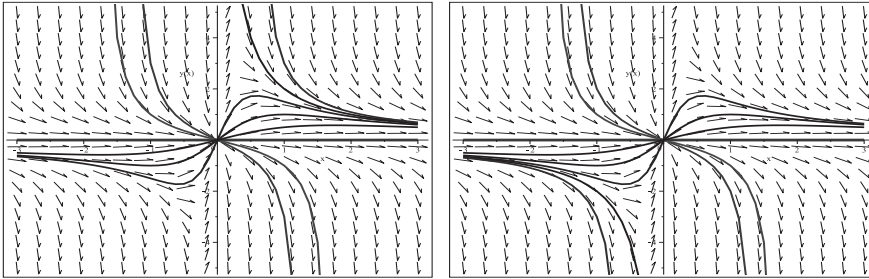


Abb. 2.60 Dateien *dar_Loes_19c1..2.ps*, Richtungsfeld zur gDGl (19), Näherungslösungen zu den AB $y(\pm 1) = y_0$ mit `pts1/pts1m`, Kommando `DEplot`, $x=-3..3, y=-5..5, method=rkf45$, ähnlich bei $x=-2..3, x=-1..3$

```
> p11:=DEplot({dgl=0}, [y(x)], x=-3..3, y=-200..200, color=black,
# oder ohne y-Bereich
[[1,2]], # [[-1,-2]],
method=rkf45, stepsize=0.01,
linecolor=black):
p11;
```

Man beachte die Integration über die Stelle $x = 0$ hinweg.

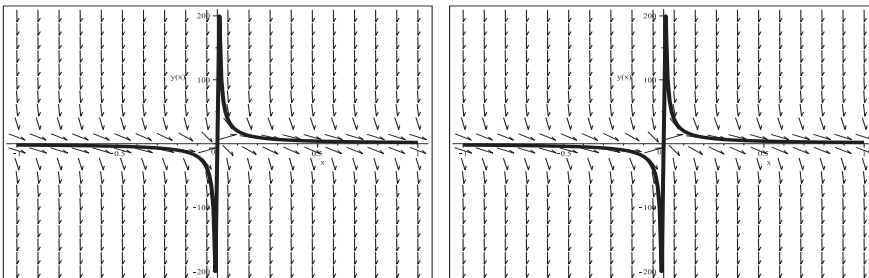


Abb. 2.61 Dateien *dar_Loes_19c3..4.ps*, Richtungsfeld zur gDGl (19), Näherungslösungen mit den AB $y(\pm 1) = \pm 2$, Kommando `DEplot`, $x=-1..1, y=-200..200$, `method=rkf45, stepsize=0.01`, ähnlich bei anderen x -Breichen

(20) Riccati-DGL $y' = x - y^2$, $y(0) = 1$

$$y(x) = \frac{c \operatorname{AiryAi}(1, x) + \operatorname{AiryBi}(1, x)}{c \operatorname{AiryAi}(x) + \operatorname{AiryBi}(x)}, \quad c = \frac{-2\pi 3^{5/6} + 3^{5/3} \Gamma^2(2/3)}{2\pi 3^{1/3} + 3^{7/6} \Gamma^2(2/3)}, \quad -1 < x < \infty$$

Die allgemeine parameterabhängige Lösung ist

$$y(x, c) = \frac{c \operatorname{AiryAi}(1, x) + \operatorname{AiryBi}(1, x)}{c \operatorname{AiryAi}(x) + \operatorname{AiryBi}(x)}.$$

Zu jeder AB gibt es einen Lösungsweig, aber viele Funktionszweige mit entsprechenden Polstellen. Der am weitesten rechts liegende Zweig tendiert gegen eine stabile Gleichgewichtslage. Nach links hin wiederholen sich die steilen Zweige periodisch.

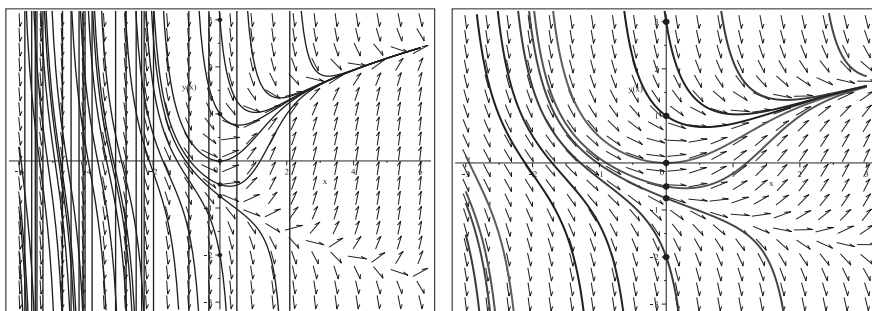


Abb. 2.62 Dateien *dar_Joes_20a1..2.ps*, Richtungsfeld zur gDGL (20), $x \in [-6, 6]$ bzw. $[-3, 3]$, Lösungen mit den AB $y(0) = 1, -2, -\frac{3}{4}, -\frac{1}{2}, 0, 3$, mit Lösungszweigen zwischen den Polstellen, Kommandos `dsolve`, `dfieldplot`, `plot`, `pointplot`

Bei den numerischen Verfahren im Kommando `DEplot` beschränken wir uns auf die Methoden `method=classical[rk4]` und `rkf45` sowie eine entsprechende Wahl des endlichen Integrationsbereichs und der Parameter.

Wenn diese nahe der Polstellen rechnen, entstehen extrem große Werte. Deshalb ist die Angabe eines y -Bereichs angebracht.

Gerade, wenn man mehrere Näherungskurven berechnet, gibt es bei einem großen x -Bereich viele Polstellen. So kann es passieren, dass die Methode `method=classical[rk4]` keine Kurven berechnet, während `rkf45` nicht ganz so empfindlich reagiert. Die Wahl des Integrationsintervalls hat Einfluss auf die y -Werte und Anzeige. Und dann ist da noch die Abhängigkeit vom Parameter `stepsize`, der Maple-Version und `Digits`-Einstellung.

Wir notieren einige weitere Ergebnisse.

(a) `DEplot, method=classical[rk4]`

```
> pts:=[[0,1],[0,-2],[0,-3/4],[0,-1/2],[0,0],[0,3]];
> p11:=DEplot({dgl=0},{y(x)},x=-3..3, #y=-3..3, # keine Kurven
```

```

x=-0.3..0.5, #y=-3..3, # y-Bereich sinnvoll
x=-0.25..0.4, #y=-3..3,
x=-3..3,y=-3..3,

pts,color=black,
method=classical[rk4], # stepsize=0.01,
linecolor=[black,green,red,blue,magenta,cyan]):
p11;

```

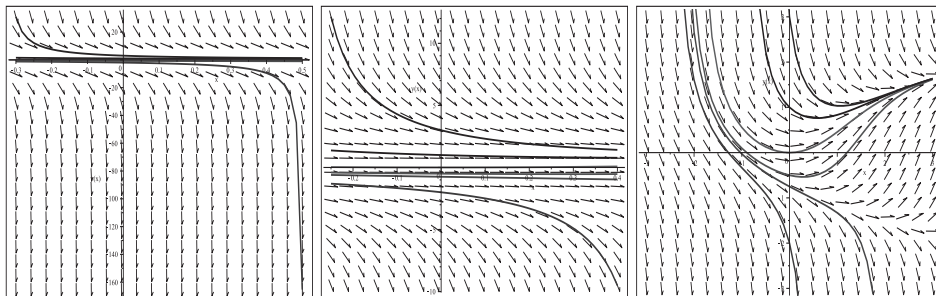


Abb. 2.63 Dateien *dar_Loes_20b1..3.ps*, Richtungsfeld zur gDGl (20), Näherungslösungen mit den AB $y(0) = 1, -2, -\frac{3}{4}, -\frac{1}{2}, 0, 3$, Kommando `DEplot, method=rk4`, 3 Bereiche wie oben

(b) `DEplot, method=rkf45`

```

> p11:=DEplot({dgl=0},[y(x)],x=-3..3, #y=-3..3, # y-Bereich sinnvoll
x=-0.9..1, #y=-3..3,
x=-3..3,y=-3..3,

pts,color=black,
method=rkf45,stepsize=0.01,
linecolor=[black,green,red,blue,magenta,cyan]):
p11;

```

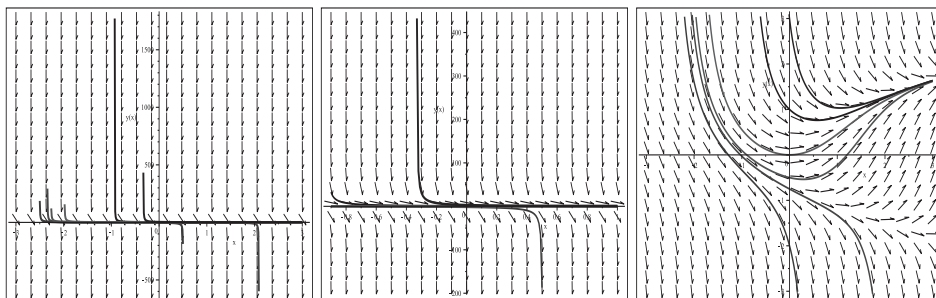


Abb. 2.64 Dateien *dar_Loes_20c1..3.ps*, Richtungsfeld zur gDGl (20), Näherungslösungen mit den AB $y(0) = 1, -2, -\frac{3}{4}, -\frac{1}{2}, 0, 3$, Kommando `DEplot,method=rkf45,stepsize=0.01`, 3 Bereiche wie oben

$$(21) \quad y' = \text{sign}(\sin(x)) - y, \quad y(1) = -1$$

$$y(x) = \text{sign}(\sin(x)) - 2e^{1-x} = 1 - 2e^{1-x}, \quad 0 < x < \pi$$

Obwohl die rechte Seite der gDGL offensichtlich Unstetigkeitsstellen bei $x = k\pi$ hat, kann man eine allgemeine parameterabhängige Lösung notieren.

$$y(x, c) = \text{sign}(\sin(x)) + ce^{-x} = \begin{cases} 1 + ce^{-x} & \text{für } x \in (2k\pi, (2k+1)\pi), \quad k \text{ ganzz.}, \\ -1 + ce^{-x} & \text{für } x \in ((2k-1)\pi, 2k\pi), \\ ce^{-x} & \text{für } x = k\pi. \end{cases}$$

An den erwähnten Unstetigkeitsstellen macht die Funktion einen Sprung der Höhe 2. Dort findet man auch die sogenannten Knicke im Richtungsfeld der gDGL.

Normalerweise sind die exakten und auch die Näherungslösungen nur in den offenen Intervallen $(k\pi, (k+1)\pi)$, k ganzzahlig, zu suchen. Darin ist die gDGL linear und inhomogen mit konstanten Koeffizienten, also von einfacher Gestalt. In den Intervallen sind auch die AB vorzugeben.

Mit der AB $y(x_0) = y_0$ erhält man die Lösung

$$y(x) = \text{sign}(\sin(x)) - e^{x_0-x} (y_0 - \text{sign}(\sin(x_0))).$$

Formal kann man diese Funktion überall berechnen.

Für $x \gg 0$ tendiert sie gegen die 2π -periodische Signalfunktion $\text{sign}(\sin(x))$, für $x \ll 0$ i. Allg. gegen $\pm\infty$, je nach dem Vorzeichen von $y_0 - \text{sign}(\sin(x_0))$, mit kleinen Sprüngen "unterwegs".

AWP mit speziellen AB und Lösungen sind

$$\begin{aligned} y(0) = 1, & \quad y(x) = \text{sign}(\sin(x)) + e^{-x}, \\ y(0) = 0, & \quad y(x) = \text{sign}(\sin(x)), \\ y(0) = -1, & \quad y(x) = \text{sign}(\sin(x)) - e^{-x}, \\ y(1) = 1, & \quad y(x) = \text{sign}(\sin(x)), \\ y(1) = 0, & \quad y(x) = \text{sign}(\sin(x)) - e^{1-x}, \\ y(1) = -1, & \quad y(x) = \text{sign}(\sin(x)) - 2e^{1-x}. \end{aligned}$$

Wir zeichnen das Richtungsfeld der gDGL (21) und zu einigen AB die Lösungen.

```
> pts:=[[0,-1],[0,0],[0,1],[0,3/2],[0,2]]:
> pts1:=[[1,-1],[1,0],[1,1],[1,3/2],[1,2]]:
> f1:=rhs(dsolve({dgl=0,y(0)=-1},y(x))); ...
> f1a:=rhs(dsolve({dgl=0,y(1)=-1},y(x))); ...
> p11:=dfieldplot({dgl=0},[y],x=-4..13,y=-2..3,color=black,dirgrid=[30,30]):
> p12:=plot([f1,f2,f3,f4,f5],x=-4..13,-2..3,color=black,thickness=2):
```

```

p13:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12,p13);
> p12:=plot([f1a,f2a,f3a,f4a,f5a],x=-4..13,-2..3,color=black,thickness=2,
           discontinuity=true):
p131:=pointplot(pts1,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12,p131);

```

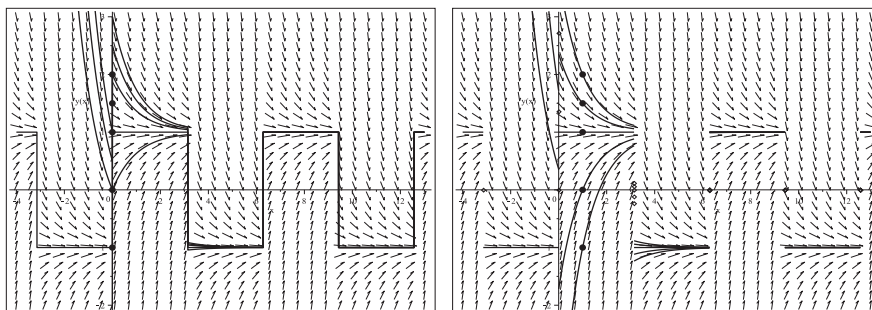


Abb. 2.65 Dateien *dar_Loes_21a1..2.ps*, Richtungsfeld zur gDGL (21), Lösungen mit den AB $y(0) = -1, 0, 1, \frac{3}{2}, 2$ bzw. $y(1) = -1, 0, 1, \frac{3}{2}, 2$, mit/ohne Vertikalen an den Sprungstellen (Option `discont=false/true`), Kommandos `dsolve`, `dfieldplot`, `plot`, `pointplot`

Bei den numerischen Verfahren im Kommando `DEplot` beschränken wir uns auf die Methoden `method=classical[rk4]` und `rkf45` sowie eine entsprechende Wahl von Integrationsbereich und Parameter. Beide Verfahren liefern ähnliche Ergebnisse, letzteres ist etwas genauer. Beide sowie weitere Methoden niedriger Ordnung integrieren über die Sprungstellen hinweg und können auch dort ihre AB haben. Es entstehen stetige Kurven mit Knicken an den Sprungstellen. Für $x \gg 0$ tendiert die Näherungslösung gegen eine 2π -periodische Funktion ähnlich einer Sägezahnkurve.

Genauere Verfahren wie `gear`, `dverk78`, ... integrieren nicht über die Sprungstellen hinweg bzw. erzeugen keine Näherungslösung mit AB an solchen.

Eine gute numerische Näherungslösung ist nur zwischen zwei benachbarten Sprungstellen zu erwarten, dort sollte auch die AB liegen.

Startet ein Näherungsverfahren an einer Sprungstelle, dann vollzieht es den Sprung der dazugehörigen exakten Lösung nicht nach, sondern geht stetig nach rechts und links. Um zumindest im ersten Abschnitt eine gute Näherung zu haben, muss die AB $y(x_0) = y_0$ so gewählt werden, dass y_0 der Wert gleich nach dem Sprung ist.

Die Angabe eines y -Bereichs ist sinnvoll, wenn $x \ll 0$ und damit große y -Werte entstehen können. Der Parameter `stepsize` hat Einfluss auf die Genauigkeit. Dazu ist stets die Abhängigkeit von Integrationsintervall, Maple-Version und/oder `Digits`-Einstellung zu beachten.

Einige weitere Ergebnisse sind die folgenden.

(a) `DEplot, method=classical[rk4]`, ähnlich `method=rkf45`

```
> # Kurven mit Knicken, Angabe von y-Bereich sinnvoll
p11:=DEplot({dgl=0},[y(x)],x=-4..13,y=-3..3,,color=black,
# p11:=DEplot({dgl=0},[y(x)],x=-1..4,y=-2..3,color=black,
pts, # pts1
method=classical[rk4], # method=rkf45,
stepsize=0.01,
linecolor=[black,green,red,blue,magenta]):
display(p11,p13); # display(p11,p131);
```

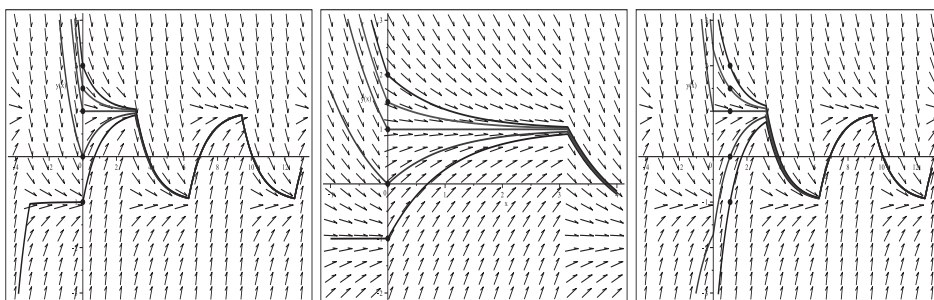


Abb. 2.66 Dateien *dar_Joes_21b1..3.ps*, Richtungsfeld zur gDGL (21), Näherungslösungen mit AB $y(0) = -1, 0, 1, \frac{3}{2}, 2$ bzw. $y(1) = -1, 0, 1, \frac{3}{2}, 2$, Kommando `DEplot, method=classical[rk4]`, Bereiche wie oben

(b) `DEplot, method=rkf45`

Wir vergleichen nun die exakte Lösung und eine Näherung bei verschiedenen AB.

- Die AB $y(1) = -1$ ist im Inneren von $(0, \pi)$.
Die exakte Lösung ist $y(x) = \text{sign}(\sin(x)) - 2e^{1-x}$.
Die Näherungslösung mit dieser AB ist sehr genau im Intervall $(0, \pi)$, außerhalb nicht.
- Die AB $y(0) = -1$ liegt an der Sprungstelle $x = 0$.
Die exakte Lösung ist $y(x) = \text{sign}(\sin(x)) - e^{-x}$ und macht bei 0 einen Sprung von -2 auf 0. Die Näherungslösung mit dieser AB läuft stetig in beide Richtungen und kann mit der exakten Lösung nicht übereinstimmen.
- Die AB für die Näherungslösung $y(0) = 0$ liegt an der Sprungstelle $x = 0$.
Die exakte Lösung ist $y(x) = \text{sign}(\sin(x)) - e^{-x}$ und macht bei 0 einen Sprung von -2 auf 0. Damit kann die Näherungslösung mit der AB $y(0) = 0$ die exakte Lösung mit der AB $y(0) = -1$ zumindest im Intervall $(0, \pi)$ gut approximieren, außerhalb nicht.

```

> # AB im Inneren
# Uebereinstimmung im Intervall (0,Pi)
p11:=DEplot({dgl=0}, [y(x)], x=0..13, y=-4.5..1.1, # y-Bereich sinnvoll
            [[y(1)=-1]], color=black,
            method=rkf45, stepsize=0.01,
            linecolor=black, dirgrid=[30,30]):
p12:=plot(f1a, x=0..13, y=-4.5..1.1, color=blue, thickness=3, linestyle=dash):
p13:=pointplot([[1, -1]], color=black, symbol=solidcircle, symbolsize=16):
display(p11, p12, p13);

> # AB an der Sprungstelle
# keine Uebereinstimmung im Intervall (0,Pi)
p11:=DEplot({dgl=0}, [y(x)], x=-4..13, y=-4..1.1, # y-Bereich sinnvoll
            [[y(0)=-1]], color=black,
            method=rkf45, stepsize=0.01,
            linecolor=black, dirgrid=[30,30]):
p12:=plot(f1, x=-4..13, y=-4..1.1, color=blue, thickness=3, linestyle=dash):
p13:=pointplot([[0, -1]], color=black, symbol=solidcircle, symbolsize=16):
display(p11, p12, p13);

> # AB  $y(x_0)=y_0$  an der Sprungstelle mit anderem  $y_0$ -Wert
# Uebereinstimmung im Intervall (0,Pi)
p11:=DEplot({dgl=0}, [y(x)], x=-4..13, # y=-4..1.1,
            [[y(0)=0]], color=black,
            method=rkf45, stepsize=0.01,
            linecolor=black, dirgrid=[30,120]):
p12:=plot(f1, x=-4..13, y=-4..1.1, color=blue, thickness=3, linestyle=dash):
p13:=pointplot([[0, 0]], color=black, symbol=solidcircle, symbolsize=16):
display(p11, p12, p13, view=[-4..13, -3..3]);

```

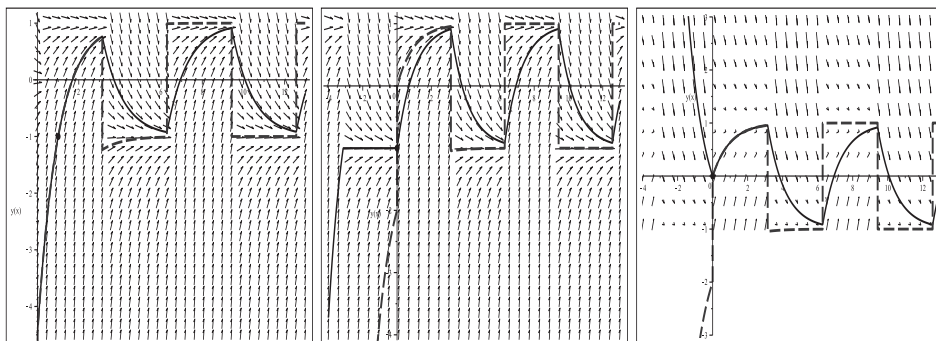


Abb. 2.67 Dateien *dar_Loes_21d1..3.ps*, Richtungsfeld zur gDGl (21), Näherungslösungen mit den AB $y(1) = -1$, $y(0) = -1$, $y(0) = 0$ im Vergleich mit den exakten Lösungen $f1a$, $f1$, $f1$ (gestrichelt), Kommando `DEplot`, `method=rkf45`, `stepsize=0.01`, Bereiche wie oben

$$(22) \quad y' = -x/y, \quad y(0) = 1$$

$$y(x) = \sqrt{1-x^2}, \quad x \in [-1, 1]$$

Man sieht, dass bei $y = 0$ das Richtungsfeld vertikale Anstiege hat und damit das Lösungsintervall beschränkt ist. Das Richtungsfeld bildet konzentrische Kreise.

Die allgemeine parameterabhängige Lösung besteht aus den beiden Zweigen

$$y_1(x) = \sqrt{c-x^2}, \quad y_2(x) = -\sqrt{c-x^2}, \quad x \in [-\sqrt{c}, \sqrt{c}], \quad c > 0.$$

Bei impliziter Lösung ergibt sich $y(x)^2 + x^2 - c = 0$. Man findet in Maple keine expliziten Lösungen mit dem Kommando `dsolve({dgl=0,y(..)=y0},y(x))`.

Aber man erhält z.B.

```
dsolve({dgl=0,y(0)=0},y(x));           -> y(x)=-Ix, y(x)=Ix
dsolve({dgl=0,y(0)=0},y(x),implicit);  -> y(x)^2+x^2=0
dsolve({dgl=0,y(1)=0},y(x));           -> y(x)=+-sqrt(1-x^2)
dsolve({dgl=0,y(1)=0},y(x),implicit);  -> y(x)^2+x^2-1=0
dsolve({dgl=0,y(x0)=y0},y(x),implicit); -> y(x)^2+x^2-y0^2-x0^2=0
```

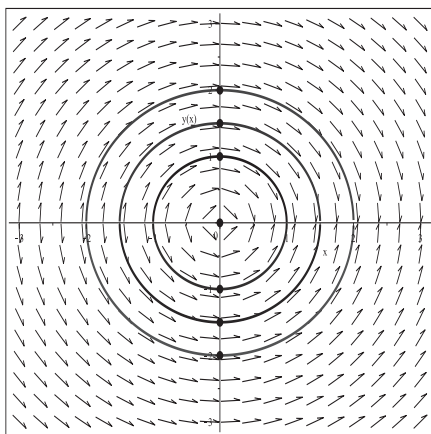


Abb. 2.68

Datei `dar_Loes.22a.ps`,
 Richtungsfeld zur gDGL des AWP (22),
 Lösungen mit den AB
 $y(0) = 1, \frac{3}{2}, 2, -2, -\frac{3}{2}, -1$,
 mit Lösungskurven
 in $[-\sqrt{x_0^2 + y_0^2}, \sqrt{x_0^2 + y_0^2}] = [-|y_0|, |y_0|]$,
 Kommandos `dsolve`, `dfieldplot`
`plot`, `pointplot`

Bei den numerischen Verfahren im Kommando `DEplot` bemerkt man einen Unterschied zwischen den Methoden niedriger und hoher Genauigkeit. Eine gute numerische Lösung zur AB $y(x_0) = y_0$ liegt im Intervall $[-\sqrt{x_0^2 + y_0^2}, \sqrt{x_0^2 + y_0^2}]$. Die Verfahren niedriger Genauigkeit integrieren über die Intervallgrenzen hinweg, die Werte außerhalb sind jedoch nicht zu gebrauchen und manchmal betragsmäßig groß. Deshalb ist die Angabe eines y -Bereichs oft sinnvoll. Bei den Verfahren hoher Ordnung passiert dies nicht.

Dann ist da noch die Abhängigkeit vom Integrationsbereich, vom Parameter `stepsize`, von der Maple-Version und `Digits`-Einstellung.

Es folgen einige weitere Ergebnisse.

(a) `DEplot, method=classical[rk4]` und weitere Methoden niedriger Ordnung

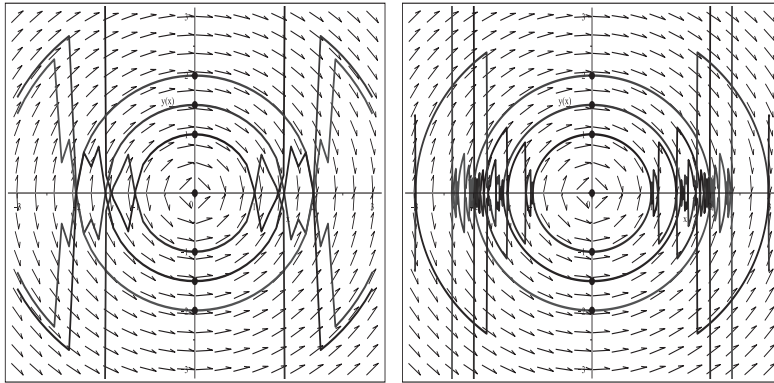


Abb. 2.69 Dateien *dar_Joes_22b1..2.ps*, Richtungsfeld zur gDGI (22), Näherungslösungen mit den AB $y(0) = 1, \frac{3}{2}, 2, -2, -\frac{3}{2}, -1$, ohne/mit Option `stepsize=0.01`, Kommando `DEplot, method=classical[rk4]`

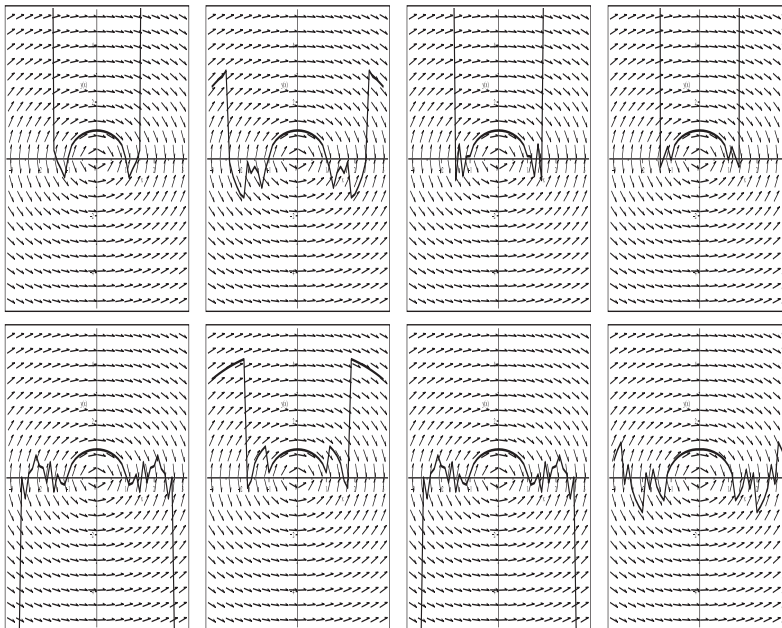


Abb. 2.70 Dateien *dar_Joes_22c31..8.ps*, Richtungsfeld zur gDGI (22), Näherungslösungen mit der AB $y(0) = 1$, Methoden v.l.o.n.r.u. `rk4, abmoulton, adambash, rk3, rk2, impoly, heunform, foreuler`, Kommando `DEplot, method=classical[...]` ohne Option `stepsize`

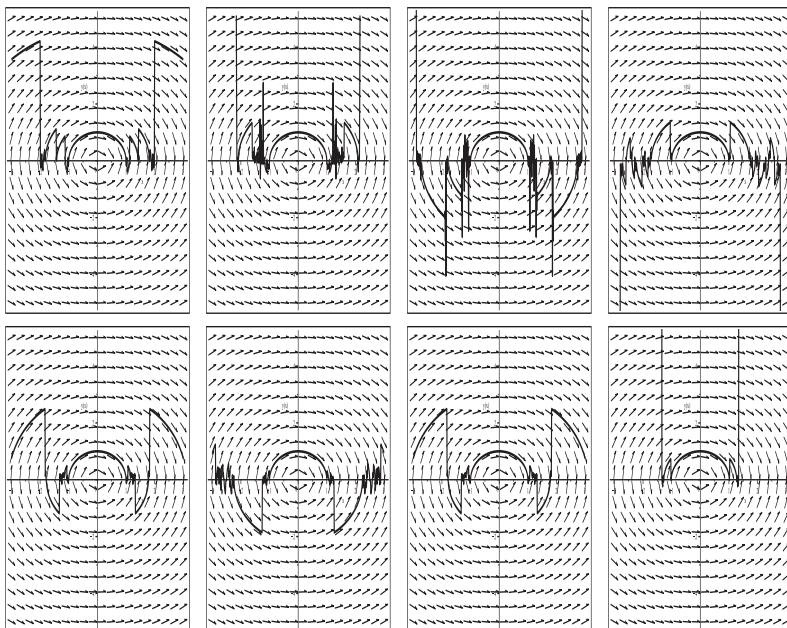


Abb. 2.71 Dateien *dar_Joes_22d31..8.ps*, Richtungsfeld zur gDGl (22), Näherungslösungen mit der AB $y(0) = 1$, Methoden v.l.o.n.r.u. `rk4, abmoulton, adambash, rk3, rk2, impoly, heunform, foreuler`, Kommando `DEplot, method=classical[.], stepsize=0.01`

(b) `DEplot, method=rkf45`

Ähnlich verhalten sich die Methoden `ck45, gear, rosenbrock, dverk78`.

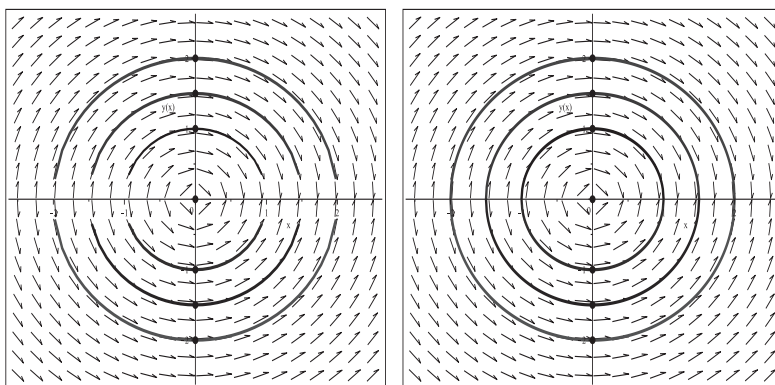


Abb. 2.72 Dateien *dar_Joes_22e1..2.ps*, Richtungsfeld zur gDGl (22), Näherungslösungen mit den AB $y(0) = 1, \frac{3}{2}, 2, -2, -\frac{3}{2}, -1$, Kommando `DEplot, method=rkf45` ohne/mit Option `stepsize=0.01`

$$(23) \quad \cos(x)y''' - y'' + \pi y' - y = -x, \quad y(0) = -1, \quad y'(0) = 2, \quad y''(0) = 1$$

Für eine gDGL höherer Ordnung kann kein Richtungsfeld erzeugt werden.

Man sieht, dass bei $x = \pm \frac{\pi}{2}$ der Faktor vor der dritten Ableitung verschwindet und die gDGL eine Ordnung verliert. Dies hat Einfluss auf das Lösungsintervall.

Es gibt keine allgemeine Lösung, auch keine für ein AWP.

Eine hinreichend genaue numerische Lösung im Intervall $I_x = [-\frac{\pi}{2}, \frac{\pi}{2}]$ erhält man mittels `dsolve/numeric`.

```
> dgl:=cos(x)*diff(y(x),x$3)-diff(y(x),x$2)+Pi*diff(y(x),x)-y(x)+x:
> dsol23:=dsolve({dgl,y(0)=-1,D(y)(0)=2,(D@@2)(y)(0)=1},y(x),
    range=-Pi/2..Pi/2,
    numeric,stepsize=0.001,      # rkf45
    output=procedurelist):      # Standard
dsol23(0);
dsol23(-Pi/2);
dsol23(Pi/2);
P1:=[rhs(dsol23(-Pi/2)[1]),rhs(dsol23(-Pi/2)[2])];
P2:=[rhs(dsol23(Pi/2)[1]),rhs(dsol23(Pi/2)[2])];
```

$$\begin{aligned} & \left[x = 0., \quad y(x) = -1., \quad \frac{d}{dx}y(x) = 2., \quad \frac{d^2}{dx^2}y(x) = 1. \right] \\ & \left[x = -1.570796326794896, \quad y(x) = -1.448226387148232, \quad \frac{d}{dx}y(x) = -0.5277636339586919, \right. \\ & \quad \left. \frac{d^2}{dx^2}y(x) = -1.785047378439673 \right] \\ & \left[x = 1.570796326794896, \quad y(x) = -1.946708252258528, \quad \frac{d}{dx}y(x) = -1.678130743590536, \right. \\ & \quad \left. \frac{d^2}{dx^2}y(x) = -3292.047246380032 \right] \\ & P1 = [-1.570796326794896, -1.448226387148232] \\ & P2 = [1.570796326794896, -1.946708252258528] \end{aligned}$$

Numerische Lösungen zu einigen AB erzeugen wir mit dem Kommando `phaseportrait`.

```
> # sinnvolles Intervall [-Pi/2,Pi/2], kein Richtungsfeld
pl1:=phaseportrait(dgl,y(x),x=-2..2,      # x=-Pi/2..Pi/2,
    [[y(0)=-1,D(y)(0)=2,(D@@2)(y)(0)=1],
    [y(0)=0,D(y)(0)=2,(D@@2)(y)(0)=1],
    [y(0)=1,D(y)(0)=2,(D@@2)(y)(0)=1],
    [y(0)=2,D(y)(0)=2,(D@@2)(y)(0)=1]],
    y=-2..4,
    method=gear,      # rkf45,dverk,rosenbrock,classical[rk4],...
    linecolor=[black,green,red,blue],
    stepsize=0.01):
pl2:=plot({[[-Pi/2,4],[-Pi/2,-2]],[[Pi/2,4],[Pi/2,-2]]},
    color=black,linestyle=dot):
display(pl1,pl2);
```

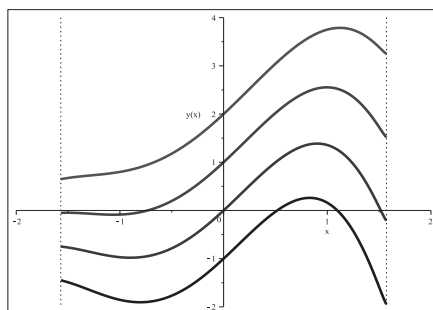


Abb. 2.73

Datei *dar_Loes_23a.ps*,
 Näherungslösungen zum AWP (23) mit den
 AB $y(0) = -1, 0, 1, 2$, $y'(0) = 2$, $y''(0) = 1$,
 Lösungskurven in $[-\frac{\pi}{2}, \frac{\pi}{2}]$,
 Kommando `phaseportrait, method=gear`

Bei den numerischen Verfahren im Kommando `DEplot` bemerkt man einige Unterschiede zwischen den Methoden.

Zunächst sollten die Verfahren eigentlich nur im Intervall $I_x = [-\frac{\pi}{2}, \frac{\pi}{2}]$ angewendet werden. Die rechte Intervallgrenze $\frac{\pi}{2}$ ist dabei "schärfer" als die linke.

Die Verfahren hoher Genauigkeit wie `gear`, `dverk78` rechnen nur im Bereich I_x .

Andere Verfahren guter Genauigkeit wie `rkf45`, `ck45`, `rosenbrock` liefern Lösungen im Bereich $(-\infty, \frac{\pi}{2}]$.

Die Verfahren wie `classical` [`rk4`, `abmoulton`, `adambash`, `rk3`, `rk2`, `heunform`, `impoly`, `foreuler`] erzeugen Kurven im Bereich $(-\infty, \infty)$, wobei an der Stelle $\frac{\pi}{2}$ meist ein Knick auftritt und dann rechts davon der Kurvenverlauf falsch wird. Links von $-\frac{\pi}{2}$ ist die Fortsetzung der Kurve oft akzeptabel.

Die Werte außerhalb von I_x sind manchmal betragsmäßig groß. Deshalb ist die Angabe eines y -Bereichs oft sinnvoll.

Dann ist da noch die Abhängigkeit vom Integrationsbereich, vom Parameter `stepsize`, der Maple-Version und `Digits`-Einstellung.

Tests können folgende Anweisungen enthalten.

```

> Digits:=16:
# Test zu Methode, Bereich, Schrittweite, AB
p1:=DEplot({dgl=0}, [y(x)], x=-3..3, y=-6..6, # y-Bereich sinnvoll
  [[y(0)=-1, D(y)(0)=2, (D@@2)(y)(0)=1]], color=black,
  #method=classical[rk2],
  # rk4, abmoulton, adambash, rk3, rk2, impoly, heunform, foreuler
#stepsize=0.01,
method=rkf45,
  # rkf45, ck45, gear, rosenbrock, dverk78, ...
  linecolor=black, axes=normal):
p2:=plot({[[-Pi/2, 0], [-Pi/2, P1[2]]], [[Pi/2, 0], [Pi/2, P2[2]]]},
  color=black):
p3:=pointplot([P1, P2], color=black, symbol=solidcircle, symbolsize=16):
p1:=subs(THICKNESS(3)=THICKNESS(4), p1):
display(p1, p2, p3);

```

Es folgen einige weitere Ergebnisse.

(a) `DEplot, method=classical[rk4]` und weitere Methoden niedriger Ordnung

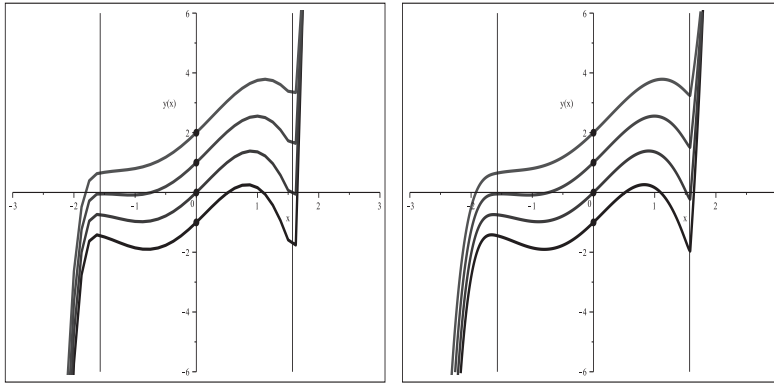


Abb. 2.74 Dateien *dar_Loes_23b1..2.ps*, `gDGI (23)`,
 Näherungslösungen mit den AB $y(0) = -1, 0, 1, 2$, $y'(0) = 2$, $y''(0) = 1$,
 ohne/mit Option `stepsize=0.01`,
 Kommando `DEplot, method=classical[rk4]`

An den Intervallgrenzen $\pm \frac{\pi}{2}$ sind zum Vergleich auch die sehr guten Näherungswerte als Punkte *P1* und *P2* eingezeichnet.

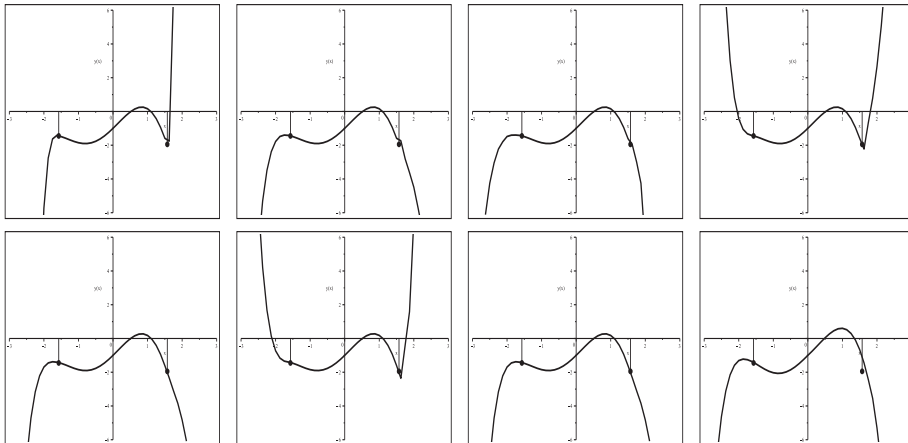


Abb. 2.75 Dateien *dar_Loes_23c31..8.ps*, `gDGI (23)`,
 Näherungslösungen mit der AB $y(0) = -1$, $y'(0) = 2$, $y''(0) = 1$,
 Methoden v.l.o.n.r.u.
`rk4, abmoulton, adambash, rk3, rk2, impoly, heunform, foreuler`,
 Kommando `DEplot, method=classical[...]` ohne Option `stepsize`

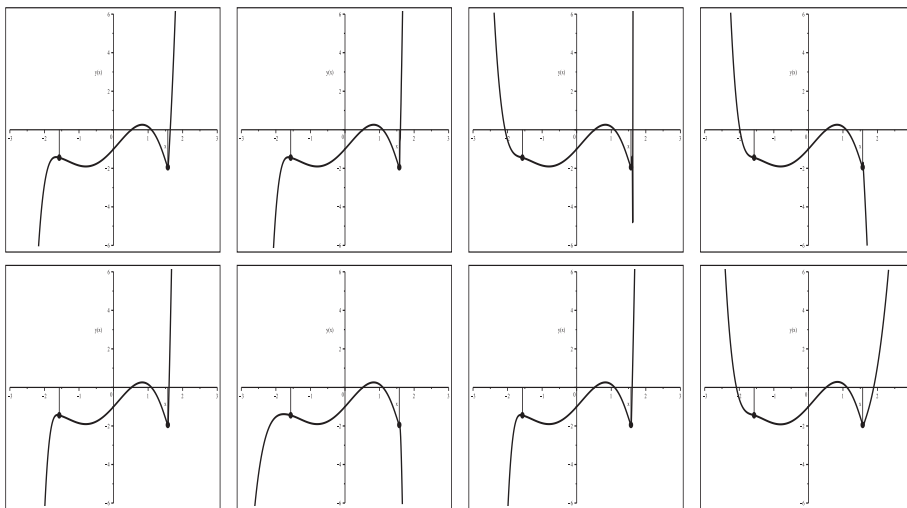


Abb. 2.76 Dateien *dar_Joes_23d31..8.ps*, gDGl (23),
 Näherungslösungen mit der AB $y(0) = -1$, $y'(0) = 2$, $y''(0) = 1$,
 Methoden v.l.o.n.r.u.
`rk4, abmoulton, adambash, rk3, rk2, impoly, heunform, foreuler,`
 Kommando `DEplot, method=classical[...], stepsize=0.01`

(b) `DEplot, method=rkf45`

Ähnlich verhalten sich auch die Methoden `ck45` und `rosenbrock`.

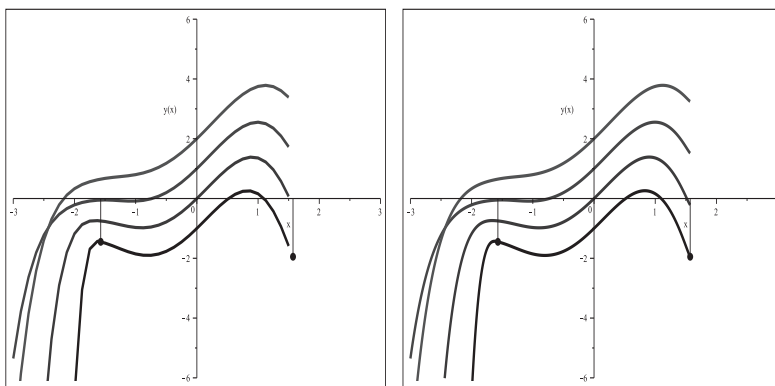


Abb. 2.77 Dateien *dar_Joes_23e1..2.ps*, gDGl (23),
 Näherungslösungen mit den AB $y(0) = -1, 0, 1, 2$, $y'(0) = 2$, $y''(0) = 1$,
 Kommando `DEplot, method=rkf45` ohne/mit Option `stepsize=0.01`

2.3 Definitionsbereiche von Lösungen

Bei der Lösung von gDGL oder AWP kontrolliert man zunächst, ob und wo die gDGL definiert ist. Dann betrachtet man das Richtungsfeld, die analytische Lösung und ihren Definitionsbereich. Schließlich findet man noch durch numerische Verfahren Näherungslösungen in ausgewählten Bereichen.

In vielen Fällen sind geeignete Methoden, Parameter, Optionen, Bereiche u.ä. zu wählen, um verständliche Darstellungen zu erzeugen. Die Standardversionen erweisen sich in manchen Fällen zu "schwach" für gute grafische Ergebnisse.

So sind z.B. folgende Situationen zu beachten.

- Das Richtungsfeld existiert überall, aber in der gDGL $y' = f(x, y)$ enthält die rechte Seite Terme der Form y^m , $m > 1$, oder e^y . Damit geht die Lösung schnell ins Unendliche. Es gibt also Polstellen (Bsp. 2.4 (7), (15), (20)).
- Das Richtungsfeld existiert an einer Stelle nicht und dort auch nicht die Lösung. Es gibt also Polstellen (Bsp. 2.4 (6), (18)).
- Das Richtungsfeld existiert an einer Stelle nicht, aber die Lösung von AWP gibt es überall. Dabei kommt es zur Mehrdeutigkeit der Lösung (Bsp. 2.4 (19)).
- Das Richtungsfeld existiert an einer Stelle nicht. Aber es gibt Lösungen von AWP, sogar eindeutige bei impliziter Option (Bsp. 2.4 (22)).
- Das Richtungsfeld existiert und macht Sprünge und die formale Lösung von AWP ebenfalls (Bsp. 2.4 (21)).

An ausgewählten Beispielen sollen Definitionsbereiche von Lösungen von gDGL erläutert werden. Auf alle möglichen Fälle wird jedoch nicht eingegangen.

Beispiel 2.5

Für das erste AWP werden die Maple-Kommandos ausführlicher notiert.

Damit wird eine sichere Vorgehensweise gezeigt. Aber nicht immer lassen sich Lösungen und Richtungsfelder so einfach wie in Teil (1) behandeln.

(1) $y' = e^y \sin(x)$ (vergl. Abschnitt 1.8.1 Nr. 72)

$$y(x) = \ln(\cos(x) + c)$$

Die rechte Seite der gDGL ist überall definiert, glatt und 2π -periodisch bzgl. x .

Das Richtungsfeld zeigt für $y < 0$ flache, für $y \ll 0$ fast horizontale Linienelemente.

Bei $y \gg 0$ bemerkt man den Wechsel zwischen Linienelementen mit absolut kleinen und extrem großen Steigungen. Das erzeugt den Verdacht auf Polstellen in den Lösungen.

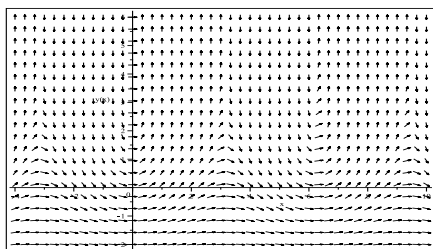


Abb. 2.78

Datei *dber_loes_01a.ps*,

Richtungsfeld zur gDGL (1),

Kommando `dfieldplot`

Somit sind AWP mit beliebigen AB lösbar. Aber man muss trotzdem auf den Definitionsbereich und das Verhalten der Lösung achten.

Wir rechnen zwei Gruppen von AWP mittels DEplot und verwenden dazu die numerischen Verfahren `method=classical[rk4]` und `rkf45`.

Neben der Methodenauswahl werden die Angabe der Bereiche und die Schrittweitenwahl `stepsize` getestet. Wenn es also Probleme mit der Darstellung gibt, versucht man sich irgendwie einzuschränken, nach dem Prinzip: manchmal ist weniger mehr.

```
> pts1:=[[0,-2],[0,-1],[0,-ln(2)],[0,-1/2],[0,0],[0,1]]: # AB
p11:=DEplot(dgl,y(x),x=-4..10,pts1,method=classical[rk4],
            color=black,linecolor=black):
p121:=pointplot(pts1,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p121);
> pts2:=[[Pi,-2],[Pi,-1],[Pi,0],[Pi,1],[Pi,4],[Pi,6]]: # AB
p11:=DEplot(dgl,y(x),x=-4..10,pts2,method=classical[rk4],
            color=black,linecolor=black,dirgrid=[29,20]):
p122:=pointplot(pts2,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p122);
```

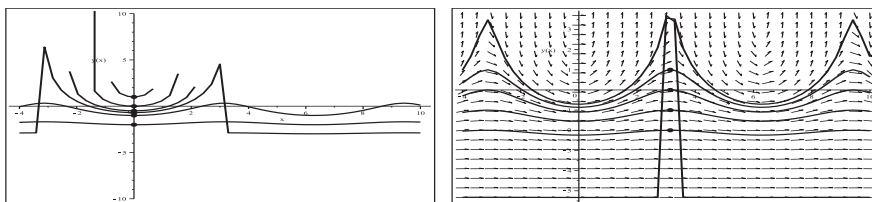


Abb. 2.79 Dateien *dber_Loes_01b01..02.ps*, nur rechts Richtungsfeld zur gDGL (1), Näherungslösungen zu den AB $y(0) = -2, -1, -\ln(2), -\frac{1}{2}, 0, 1$ (pts1) bzw. $y(\pi) = -2, -1, 0, 1, 4, 6$ (pts2), Kommando `DEplot, method=classical[rk4]`

Das Kommando mit dem KRKV erzeugt in einem Fall nicht einmal ein Richtungsfeld. Einige Kurven sind ungenau und unvollständig, Sprünge in den Verläufen nicht plausibel. Besser sind die Ergebnisse mit dem Kommando `DEplot, method=rkf45`.

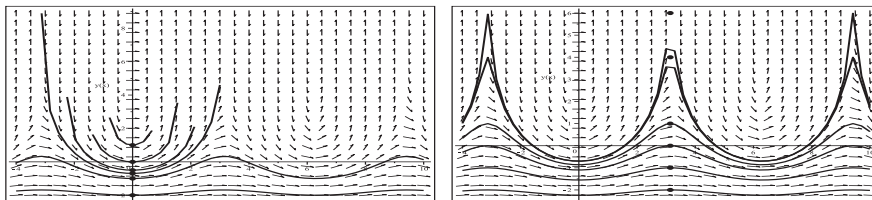


Abb. 2.80 Dateien *dber_Loes_01b03..04.ps*, Richtungsfeld zur gDGL (1), Näherungslösungen zu den AB wie in Abb. 2.79, Kommando `DEplot, method=rkf45`

Wir bleiben bei `method=rkf45` und ändern den x - und y -Darstellungsbereich.
Im nächsten Schritt erhöhen wir die Genauigkeit mittels `stepsize=0.0005`.

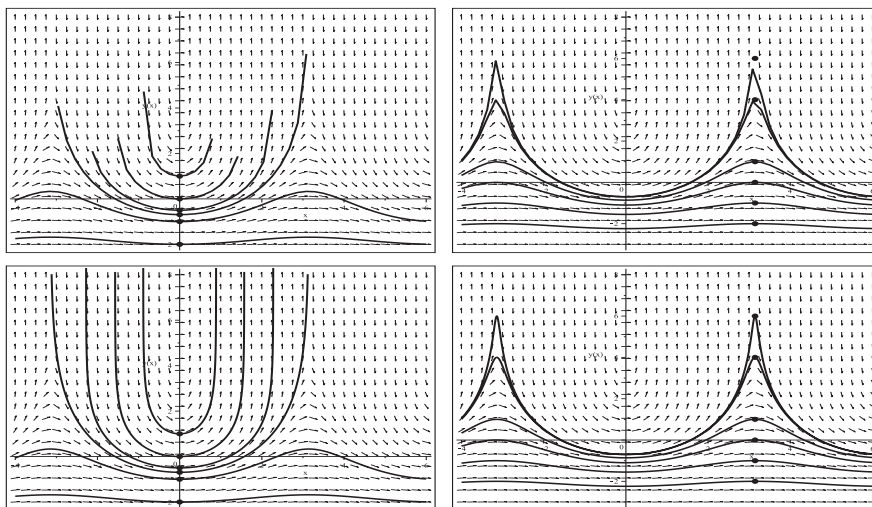


Abb. 2.81 Dateien `dbcr_loes_01b05..08.ps`, Richtungsfeld zur gDGl (1),
Näherungslösungen zu den AB wie in Abb. 2.79,
Kommando `DEplot, rkf45` oben ohne/unten mit `stepsize=0.0005`

Zum Abschluss verkleinern wir den x -Bereich auf eine Periodenlänge und nehmen auch noch einen passenden y -Bereich dazu.

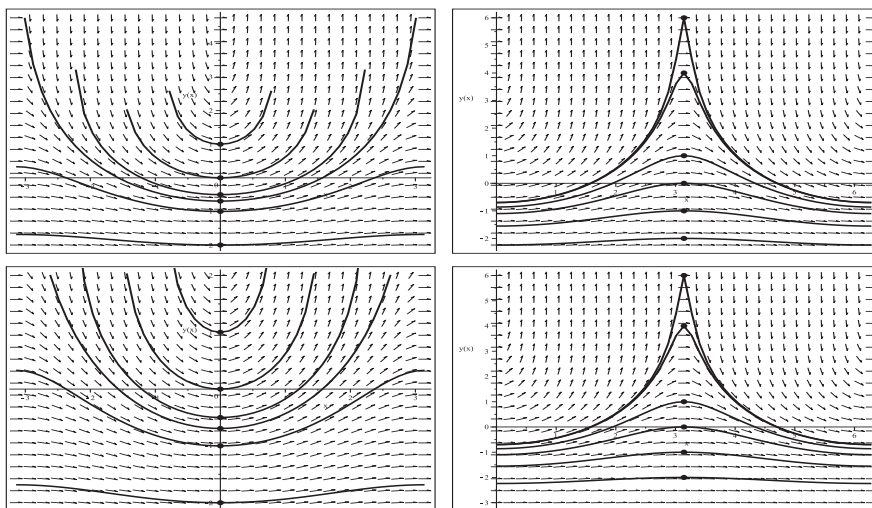


Abb. 2.82 Dateien `dbcr_loes_01b09..12.ps`, Richtungsfeld zur gDGl (1),
Näherungslösungen zu den AB wie in Abb. 2.79,
Kommando `DEplot, method=rkf45` ohne `stepsize`

Mit den beiden Gruppen von AB $y(x_0) = y_0$ bei $x_0 = 0$ bzw. $x_0 = \pi$, was in der linken bzw. rechten Spalte der obigen Abbildungen berücksichtigt wird, unterscheidet man die zwei Fälle der analytischen Lösung des AWP.

In der allgemeinen Lösung wird das von x abhängige Argument des natürlichen Logarithmus zu untersuchen sein.

Zur AB $y(x_0) = y_0$ ist die Lösung

$$y = y_0 - \ln(1 + e^{y_0} (\cos(x) - \cos(x_0))).$$

Damit erkennt man die Fallunterscheidung bzgl. der Anfangsstellen x_0 .

– $x_0 = (2k + 1)\pi$, k ganzzahlig

$$y = y_0 - \ln(1 + e^{y_0} (\cos(x) + 1)), \quad \cos(x) + 1 \geq 0, \quad g(x) = 1 + e^{y_0} (\cos(x) + 1) \geq 1$$

Es gilt

$$y' = \frac{\sin(x) e^{y_0}}{1 + e^{y_0} (\cos(x) + 1)}, \quad y'(x_0) = 0, \quad |y'(x)| \gg 0 \text{ für } x \approx x_0 \text{ und } y_0 \gg 0.$$

Die AB befindet sich dann also an einem ‘runden Pik’, was zu Problemen bei der numerischen Lösung führen kann.

– $x_0 \neq (2k + 1)\pi$, k ganzzahlig

$$y = y_0 - \ln(1 + e^{y_0} (\cos(x) - \cos(x_0))), \quad g(x) = 1 + e^{y_0} (\cos(x) - \cos(x_0)) > 0$$

Aus der Bedingung $g(x) = 0$ bildet man die Funktion

$$g_1(x_0, y_0) = \arccos(\cos(x_0) - e^{-y_0})$$

und untersucht, für welche Argumente x_0, y_0 die Funktion einen reellen Wert $z = g(x_0, y_0)$ erzeugt. Dann ist das Lösungsintervall $(-z, z)$, das man mit der Periode 2π fortsetzen kann. Bei $\pm z$ hat man Polstellen der Lösung. Im Intervall $[z, 2\pi - z]$ gibt es keine Lösung. Der Grenzfall ist $z = \pi$.

Ist $z = g(x_0, y_0)$ komplex, ist die Lösung für alle x definiert.

Für $x_0 = 0$ erhalten wir Folgendes.

- $g(x) = 1 + e^{y_0} (\cos(x) - 1)$,
- $g_1(0, y_0) = \arccos(1 - e^{-y_0}) = \pi - \arccos(-1 + e^{-y_0}) \rightarrow$ reell, $y_0 \in [-\ln(2), \infty)$,
- $g_1(0, -2) = \pi - 2.541i$, $g_1(0, -1) = \pi - 1.136i$, komplex,
- $g_1(0, -\ln(2)) = \pi$, Grenzfall $y_0 = -\ln(2)$, $z = \pi$,
- $g_1(0, -\frac{1}{2}) = 2.276$, Lösungsintervall $x \in (-2.276, 2.276)$,
- $g_1(0, 0) = \frac{\pi}{2}$, Lösungsintervall $x \in (-\frac{\pi}{2}, \frac{\pi}{2})$,
- $g_1(0, \frac{1}{2}) = 1.166$, $g_1(0, 1) = 0.886$, $g_1(0, 2) = 0.526$.

Die Rechnung mit $g_1(\pi, y_0)$ liefert ausschließlich komplexe Werte.

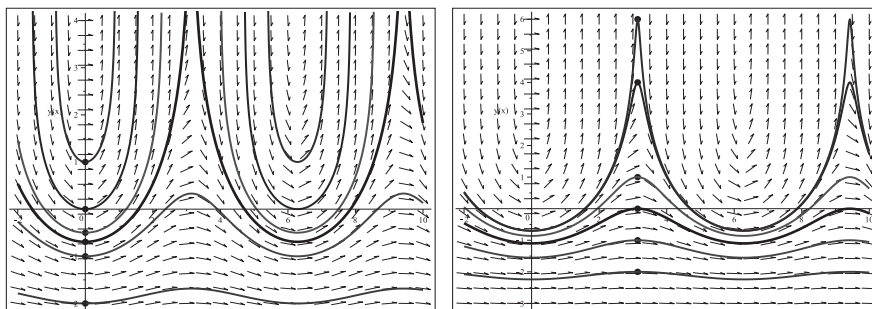


Abb. 2.83 Dateien *dber_loes_01c1..2.ps*, Richtungsfeld zur gDGL (1),
 exakte Lösungen mit periodischer Fortsetzung zu den
 AB $y(0) = -2, -1, -\ln(2), -\frac{1}{2}, 0, 1$ bzw. $y(\pi) = -2, -1, 0, 1, 4, 6$,
 Kommandos `dsolve`, `dfieldplot`, `plot`, `pointplot`

$$(2) \quad y' = e^y \cos(x)$$

$$y(x) = -\ln(-\sin(x) + c)$$

Die rechte Seite der gDGL ist überall definiert, glatt und 2π -periodisch bzgl. x .

Das Richtungsfeld zeigt für $y < 0$ flache, für $y \ll 0$ fast horizontale Linienelemente.

Bei $y \gg 0$ bemerkt man den Wechsel zwischen Linienelementen mit absolut kleinen und extrem großen Steigungen. Das erzeugt den Verdacht auf Polstellen in den Lösungen.

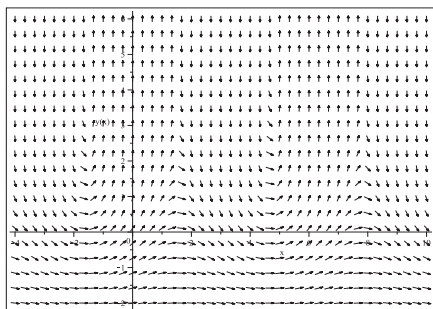


Abb. 2.84
 Datei *dber_loes_02a.ps*,
 Richtungsfeld zur gDGL (2),
 Kommando `dfieldplot`

Somit gibt es zu AWP mit beliebigen AB $y(x_0) = y_0$ die Lösung

$$y(x) = y_0 - \ln(1 - e^{y_0} (\sin(x) - \sin(x_0))).$$

Aber man muss trotzdem auf Definitionsbereich und Verhalten der Lösung achten.

In Analogie zu Teil (1) kann man erneut zwei Gruppen von AWP mittels `DEplot` rechnen und dazu die numerischen Verfahren `method=classical[rk4]` und `rkf45` verwenden. Die Fallunterscheidung bzgl. der Anfangsstellen x_0 heißt hier, dass für $x_0 = (2k + \frac{1}{2})\pi$, k ganzzahlig, und y_0 beliebig es immer Lösungen gibt. Aber für $y_0 \gg 0$ befindet sich die AB an einem "runden Pik", was zu Problemen bei der numerische Lösung führen kann.

Für die anderen x_0 ist bei hinreichend kleinen y_0 die Lösung überall definiert, bei wachsenden y_0 kommt es wieder zu Polstellen und ‐Löchern‐ im Definitionsbereich. Für $x_0 = -\frac{\pi}{2}$ ist der Grenzwert $y_0 = -\ln(2)$.

Neben der Methodenauswahl sind auch die Angabe der Bereiche und die Schrittweitenwahl `stepsize` möglich.

Die grafischen Ergebnisse sind ähnlich zu denen aus Teil (1), nur mit den kleinen Unterschied, dass man eine Verschiebung um $-\frac{\pi}{2}$ bzgl. x feststellt. Deshalb werden nur wenige Abbildungen mit zwei Varianten von AB erzeugt.

```
> pts1:=[[0,-2],[0,-1],[0,-ln(2)],[0,-1/2],[0,0],[0,1]]: # AB
#pts1:=[[-Pi/2,-2],[-Pi/2,-1],[-Pi/2,-ln(2)],[-Pi/2,-1/2],
#      [-Pi/2,0],[-Pi/2,1]];
p11:=DEplot(dgl,y(x),x=-4..10,pts1,method=classical[rk4],
            color=black,linecolor=black):
p121:=pointplot(pts1,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p121);

> pts2:=[[Pi,-2],[Pi,-1],[Pi,0],[Pi,1],[Pi,4],[Pi,6]]: # AB
#pts2:=[[Pi/2,-2],[Pi/2,-1],[Pi/2,0],[Pi/2,1],[Pi/2,4],[Pi/2,6]];
p11:=DEplot(dgl,y(x),x=-4..10,pts2,method=classical[rk4],
            color=black,linecolor=black,dirgrid=[29,20]):
p122:=pointplot(pts2,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p122);
```

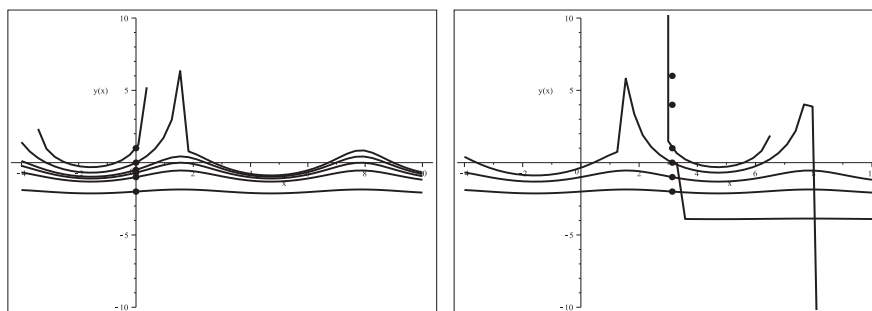


Abb. 2.85 Dateien *dbcr_Loes_02b01..02.ps*, kein Richtungsfeld zur gDGI (2), Näherungslösungen zu den AB $y(0) = -2, -1, -\ln(2), -\frac{1}{2}, 0, 1$ (pts1) bzw. $y(\pi) = -2, -1, 0, 1, 4, 6$ (pts2), Kommando `DEplot, method=classical[rk4]`

Das Kommando mit dem KRKV erzeugt kein Richtungsfeld. Einige Kurven sind ungenau und unvollständig, Sprünge in den Verläufen nicht plausibel. Besser und anschaulicher sind die Ergebnisse mit dem Kommando `DEplot, method=rkf45`, aber noch nicht zufriedenstellend, wenn man die fehlenden Lösungen zu den AB $y(\pi) = 4$ und $y(\pi) = 6$ bemerkt.

Man sollte also noch weitere Methoden testen, wie z.B. `method=gear`, sowie den x - und y -Darstellungsbereich und andere Parameter ändern.

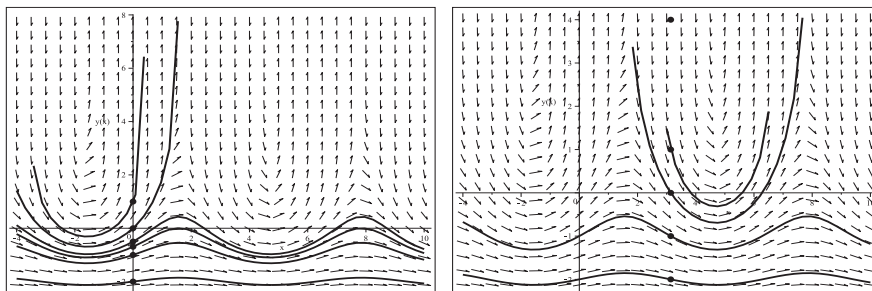


Abb. 2.86 Dateien *dber_Loes_02b03..04.ps*, Richtungsfeld zur gDGl (2), Näherungslösungen mit den AB wie in Abb. 2.85, Kommando `DEplot, method=rkf45`

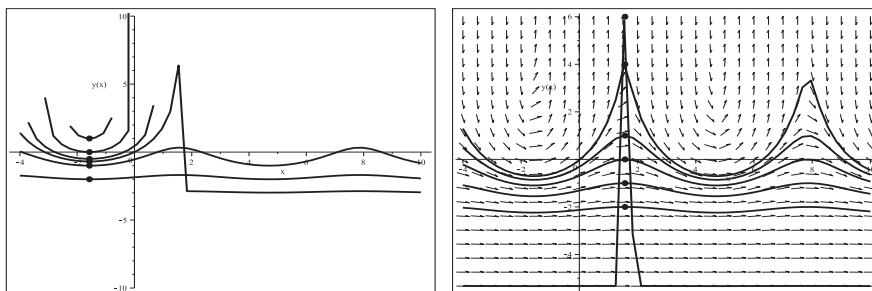


Abb. 2.87 Dateien *dber_Loes_02d01..02.ps*, nur rechts Richtungsfeld zur gDGl (2), Näherungslösungen mit den AB $y(-\frac{\pi}{2}) = -2, -1, -\ln(2), -\frac{1}{2}, 0, 1$ bzw. $y(\frac{\pi}{2}) = -2, -1, 0, 1, 4, 6$, Kommando `DEplot, method=classical[rk4]`

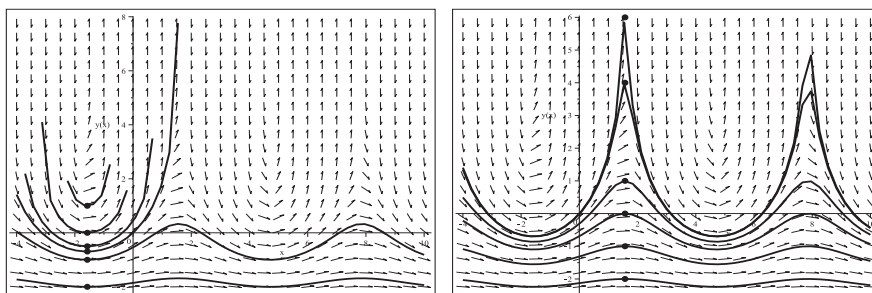


Abb. 2.88 Dateien *dber_Loes_02d03..04.ps*, Richtungsfeld zur gDGl (2), Näherungslösungen zu den AB wie in Abb. 2.87, Kommando `DEplot, method=rkf45`

$$(3) \quad y' = \sqrt{y}$$

$$\sqrt{y(x)} - \frac{1}{2}x - c = 0$$

Die implizite Darstellung der Lösung in Maple garantiert die Bedingung $y \geq 0$. Diese und die zweite Forderung $y' \geq 0$ bringt die gDGl mit sich, sodass dadurch der Lösungsbereich eingeschränkt wird.

Ausgehend von der Kenntnis der Lösung der gDGl als konvexe quadratische Parabel heißt das, dass nur ihr nichtnegativer wachsender Zweig als Lösungskurve in Frage kommt.

Das Richtungsfeld gibt es für $y \geq 0$ und ist von einfacher Form. Da für $y = 0$ alle Linienelemente auf der x -Achse liegen, existiert die spezielle Lösung $y(x) = 0$.

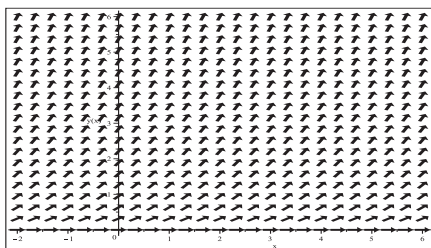


Abb. 2.89

Datei `dber_Loes_03a.ps`,
Richtungsfeld zur gDGl (3),
Kommando `dfieldplot`

Die numerische Lösung berechnet man mit dem Kommando `DEplot`, `method=rkf45`.

Die AB $y(x_0) = 0$ führt immer auf die überall definierte Nulllösung.

AB $y(x_0) = y_0 > 0$ erzeugen den wachsenden Lösungsweig der Parabel, wobei die Fortsetzung der Lösung links von der AB bis zur Stelle z mit $y(z) = 0$ erfolgt, und $z = x_0 - 2\sqrt{y_0}$ ist. Weiter kann die Methode nicht rechnen. Ein fallender Lösungsweig wäre ja auch ein Widerspruch zur Richtung der Linienelemente. So erfolgt zur AB $y(0) = 1$ die Warnung.

```
Warning, plot may be incomplete, the following error(s) were issued:
cannot evaluate the solution further left of -1.9998526,
probably a singularity
```

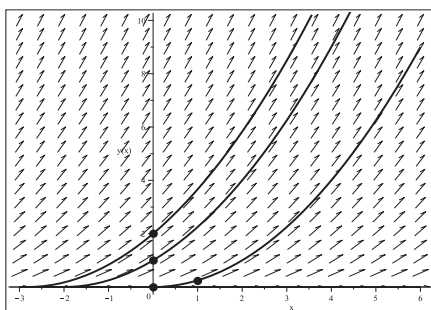


Abb. 2.90

Datei `dber_Loes_03b.ps`,
Richtungsfeld zur gDGl (3),
Näherungslösungen zu den
AB $y(0) = 0, 1, 2$ bzw. $y(1) = \frac{1}{4}$,
Kommando `DEplot`, `method=rkf45`

Es gibt zum AWP mit beliebigen AB $y(x_0) = y_0$ die Lösung

$$y(x) = \left(\frac{x - x_0 + 2\sqrt{y_0}}{2} \right)^2, \quad x \in [x_0 - 2\sqrt{y_0}, +\infty).$$

Man erkennt also den Scheitelpunkt der konvexen Parabel auf der x -Achse an der Stelle $z = x_0 - 2\sqrt{y_0}$. Für die AB $y(1) = 1$ erhalten wir die Lösung $y(x) = \left(\frac{x+1}{2}\right)^2$, $x \geq -1$.

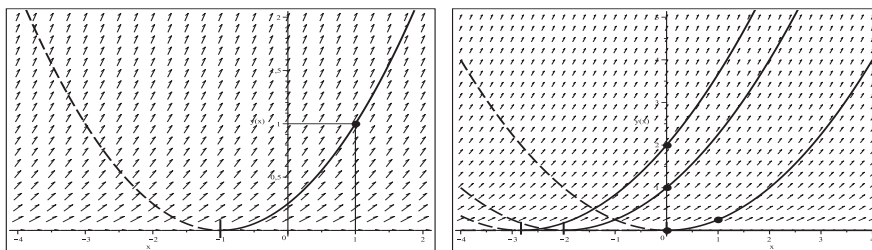


Abb. 2.91 Dateien *dber_Loes_03c1..2.ps*, Richtungsfeld zur gDGL (3),
 Lösungen (solid line) zu den AB $y(1) = 1$ bzw. $y(0) = 0, 1, 2$, $y(1) = \frac{1}{4}$,
 Kommandos `dsolve`, `dfieldplot`, `plot`, `pointplot`

Mehr noch, zur AB $y(x_0) = 0$ findet man unendlich viele Lösungen. Die Vorgehensweise ist einfach. Sei $x_0 = 0$. Dann erhält man die Lösungen $y_1(x) = 0$, $y_2(x) = \left(\frac{x}{2}\right)^2$ und

$$y_3(x) = \begin{cases} 0 & \text{für } x \in [0, c], \quad c > 0 \text{ beliebig,} \\ \left(\frac{x-c}{2}\right)^2 & \text{für } x \in (c, \infty). \end{cases}$$

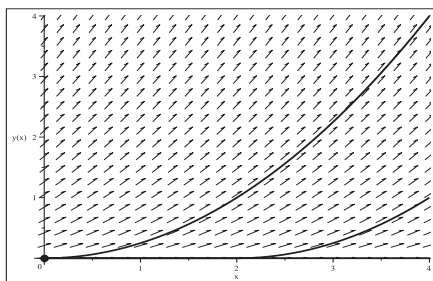


Abb. 2.92
 Datei *dber_Loes_03d.ps*,
 Richtungsfeld zur gDGL (3),
 Lösungen $y_1(x)$, $y_2(x)$, $y_3(x)$ mit $c = 2$
 zur AB $y(0) = 0$,
 Kommandos `dsolve`, `dfieldplot`,
`plot`, `pointplot`

$$(4) \quad y' = 1/(1 + \tan^2(y))$$

$$y(x) = \arctan(x + c)$$

Obwohl die Werte $y = y_k = \left(k + \frac{1}{2}\right)\pi$, k ganzzahlig, auf der rechten Seite dieser gDGL eigentlich auszuschließen sind, kann man formal wegen der Existenz des Grenzwertes $\lim_{y \rightarrow y_k} \frac{1}{1 + \tan^2(y)} = 0$ die horizontalen Linienelemente an den Stellen y_k und damit die stationären Lösungen (Gleichgewichtslagen) $y(x) = y_k$ als Grenzfälle mit einbeziehen.

Das Richtungsfeld ist π -periodisch bzgl. y .

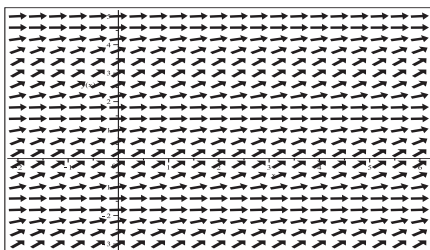


Abb. 2.93

Datei *dber_Loes_04a.ps*,
Richtungsfeld zur gDGL (4),
Kommando `dfieldplot`

Die numerische Lösung berechnet man mit dem Kommando `DEplot`, `method=rkf45`.
Zu allen AB $y(x_0) = y_0$, auch bei $y(x_0) = y_k$, werden numerische Lösungen berechnet.

```
> # AB
pts := [[0, -2], [0, -Pi/2], [0, -1], [0, 0], [0, 1], [0, Pi/2], [0, 2], [0, Pi],
        [0, 3*Pi/2]]:
pts1 := [op(pts), [0, Pi/2-0.1], [0, Pi/2+0.1]]:
pts2 := [[0, -2], [0, -Pi/2], [0, -1], [0, 0], [0, 1], [0, Pi/2-0.1], [0, Pi/2+0.1],
        [0, 2], [0, Pi]]:

> p1 := DEplot(dgl, y(x), x=-2..6, y=-3..5, pts, color=black, linecolor=black,
             dirgrid=[33,36]):
p2 := pointplot(pts1, color=black, symbol=solidcircle, symbolsize=16):
dsol41 := dsolve({dgl, y(0)=Pi/2-0.1}, y(x), range=-2..6, numeric): # rkf45
dsol42 := dsolve({dgl, y(0)=Pi/2+0.1}, y(x), range=-2..6, numeric):
p131 := odeplot(dsol41, color=black, thickness=2, linestyle=dash):
p132 := odeplot(dsol42, color=black, thickness=2, linestyle=dash):
display(p1, p2, p131, p132);
```

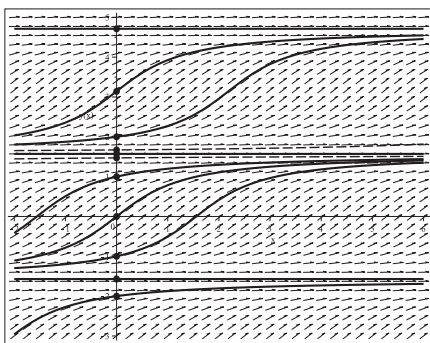


Abb. 2.94

Datei *dber_Loes_04b.ps*,
Richtungsfeld zur gDGL (4),
Näherungslösungen zu den
AB gemäß `pts1`,
Kommando `DEplot`, `method=rkf45`

Wir kontrollieren noch einige Verläufe von Näherungsfunktionen mit AB nahe oder bei y_k , die mit dem Kommando `dsolve/numeric/rkf45` erzeugt werden. Dabei sollen die Ergebnisse zum Vergleich in einem groben bzw. feinen Maßstab angezeigt werden.

```
> dsol4 := dsolve({dgl, y(0)=Pi/2-0.01}, y(x),
                numeric, method=rkf45, # Standard
                range=-2..6, # dann nur rkf45 verwendbar
                output=procedurelist): # Standard
```

```

dsol4(0); # Kontrolle der AB
p11:=odeplot(dsol4,color=black,thickness=3,view=[-2..6,-3..5]):
p12:=pointplot([0,Pi/2-0.01],color=black,symbol=solidcircle,
symbolsize=16):
display(p11,p12); # grosses Fenster

> p11:=odeplot(dsol4,color=black,thickness=3): # ohne view
p13:=plot(Pi/2,x=-2..6,color=black,linestyle=dot):
p14:=textplot([-0.5,1.5708,' Pi/2'],color=black,font=[HELVETICA,BOLD,9]):
display(p11,p12,p13,p14); # kleines Fenster, direkt betroffener Bereich

```

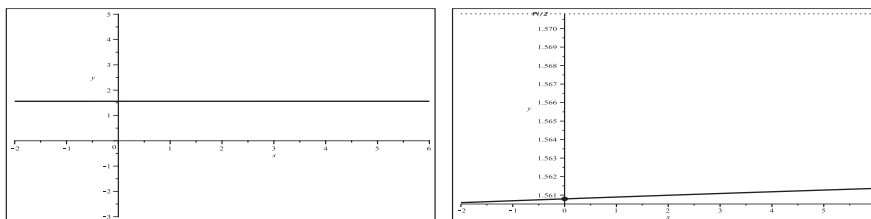


Abb. 2.95 Dateien *dber_Loes_04b11..12.ps*, gDGl (4),
 Näherungslösung zur AB $y(0) = \frac{\pi}{2} - 0.01$ im Fenster groß/klein,
 Kommandos `dsolve/numeric`, `odeplot`, `plot`, `pointplot`, `textplot`

Analog wird es auch für die AB $y(0) = \frac{\pi}{2}$, $\frac{3\pi}{2}$, $-\frac{\pi}{2}$ gemacht.

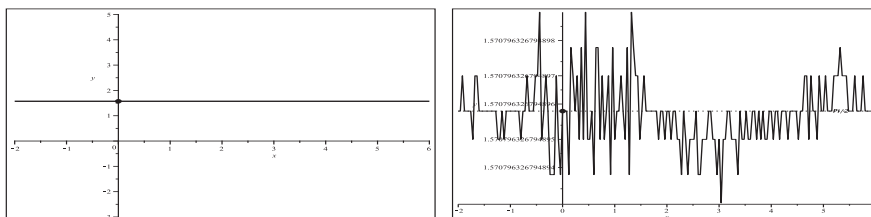


Abb. 2.96 Dateien *dber_Loes_04b21..22.ps*, gDGl (4),
 Näherungslösung zur AB $y(0) = \frac{\pi}{2}$ im Fenster groß/klein,
 Kommandos `dsolve/numeric`, `odeplot`, `plot`, `pointplot`, `textplot`

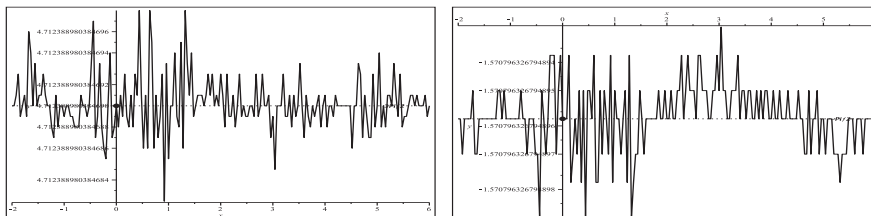


Abb. 2.97 Dateien *dber_Loes_04b31..32.ps*, gDGl (4),
 Näherungslösung zu den AB $y(0) = \frac{3\pi}{2}$ bzw. $y(0) = -\frac{\pi}{2}$ im Fenster klein,
 Kommandos `dsolve/numeric`, `odeplot`, `plot`, `pointplot`, `textplot`

Obwohl die Gleichgewichtslagen nur einseitig stabil sind, driften die Näherungslösungen nicht weg.

Bei den Verfahren `rkf45`, `ck45`, `rosenbrock` "zittern die Kurven", bei den anderen Methoden tritt dieses Verhalten nicht auf.

Kehren wir zur analytischen Lösung der gDGL zurück.

Die allgemeine explizite Lösung $y(x) = \arctan(x + c)$ beinhaltet keine der konstanten Trajektorien $y(x) = y_k$. So ist es auch nicht verwunderlich, dass die Kommandos `dsolve({dgl,y(x0)=y0},y(x))` und `dsolve({dgl,y(0)=y0},y(x))` kein Ergebnis liefern. Wählt man jedoch ein konkretes y_0 , klappt es.

Z.B. erzeugt `dsol:=dsolve({dgl,y(x0)=-2},y(x))`

$$dsol := y(x) = \arctan(x - x_0 - \tan(2)) - \pi$$

Erfolglos bleibt auch das analoge Kommando mit $y(x_0) = (k + \frac{1}{2})\pi$, bis auf eine Ausnahme, nämlich $y(x_0) = -\frac{\pi}{2}$, wo das Ergebnis natürlich $y(x) = -\frac{\pi}{2}$ ist.

Damit fehlen in der folgenden Abbildung die horizontalen Geraden $y(x) = \frac{\pi}{2}, \frac{3\pi}{2}$.

```
> dsolve({dgl,y(0)=-2},y(x));          f1:=rhs(%);
dsolve({dgl,y(0)=-Pi/2},y(x));        # -> f2
dsolve({dgl,y(0)=-1},y(x));          # -> f3
dsolve({dgl,y(0)=0},y(x));           # -> f4
dsolve({dgl,y(0)=1},y(x));           # -> f5
dsolve({dgl,y(0)=Pi/2-0.1},y(x));    # -> f6
dsolve({dgl,y(0)=Pi/2+0.1},y(x));    # -> f7
dsolve({dgl,y(0)=2},y(x));           # -> f8
dsolve({dgl,y(0)=Pi},y(x));          # -> f9
> p1:=dfieldplot(dgl=0,y,x=-2..6,y=-3..5,color=black,dirgrid=[33,36]):
p2:=plot([f1,f3,f4,f5,f8,f9],x=-2..6,y=-3..5,thickness=3,color=black):
p3:=plot([f2,f2],x=-2..6,y=-3..5,thickness=[3,1],color=black,
linestyle=[dashdot,solid]):
p4:=plot([f6,f7],x=-2..6,y=-3..5,thickness=3,color=black,
linestyle=dash):
p5:=pointplot(pts2,color=black,symbol=solidcircle,symbolsize=16):
display(p1,p2,p3,p4,p5);
```

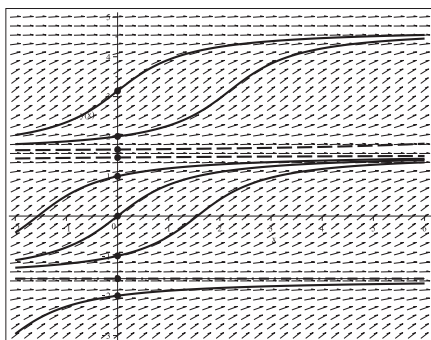


Abb. 2.98

Datei `dber_Loes_04c.ps`,

Richtungsfeld zur gDGL (4),

Lösungen f_1, f_2, \dots, f_9

zu den AB `pts2`,

Kommandos `dsolve`, `dfieldplot`,

`plot`, `pointplot`

(5) $y' = x \cos(y)$ (vergl. Abschnitt 1.8.1 Nr. 70)

$$y(x) = \arctan\left(\frac{c^2 e^{x^2} - 1}{c^2 e^{x^2} + 1}, \frac{2c^2 e^{x^2/2}}{c^2 e^{x^2} + 1}\right), \quad \arctan(y, z) = -i \ln\left(\frac{z + iy}{\sqrt{z^2 + y^2}}\right)$$

Das Richtungsfeld ist überall definiert und 2π -periodisch bzgl. y für festes x .

An den Werten $y = y_k = (k + \frac{1}{2})\pi$, k ganzzahlig, verschwindet der Cosinus und die Linienelemente sind horizontal. Damit gibt es stationäre Lösungen, die abwechselnd stabile oder instabile Gleichgewichtslagen darstellen.

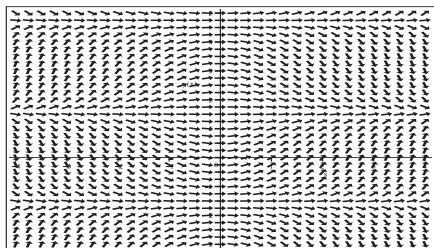


Abb. 2.99

Datei *dber_Loes_05a.ps*,
Richtungsfeld zur gDGI (5),
Kommando `dfieldplot`

Die numerische Lösung berechnet man mit dem Kommando `DEplot`.

Zu allen AB $y(x_0) = y_0$ gibt es numerische Lösungen. In kleinen Lösungsintervallen wie $[-4, 4]$ funktioniert alles recht gut. Aber bei großen Intervallen wie $[-4, 20]$ oder $[-30, 30]$, wo für $|x| \gg 1$ die Linienelemente sehr steil werden können, wird es problematisch für alle numerischen Verfahren.

```
> # AB
pts:=[[0, -2], [0, -Pi/2], [0, -1], [0, 0], [0, 1], [0, Pi/2], [0, Pi], [0, 4.5],
      [0, 3*Pi/2], [0, 5], [0, 5*Pi/2], [0, 3*Pi]]:
pl1:=DEplot(dgl, y(x), x=-4..4, # x=-4..20
            y=-4..10, pts, color=black, linecolor=black, dirgrid=[21, 21],
            method=rkf45):
            # rosenbrock, dverk78, rkf45, gear, ck45,
            # classical [abmoulton, rk4, rk3, rk2],
            # impoly, heunform, foreuler]
pl2:=pointplot(pts, color=black, symbol=solidcircle, symbolsize=16):
display(pl1, pl2);
```

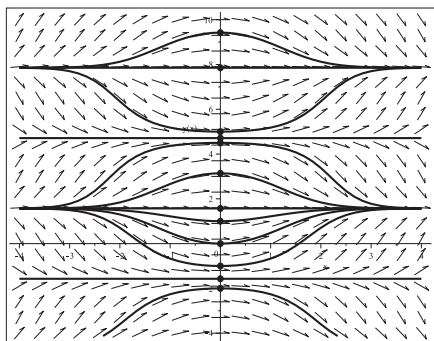


Abb. 2.100

Datei *dber_Loes_05b.ps*,
Richtungsfeld zur gDGI (5),
Näherungslösungen zu den
AB gemäß `pts`,
Kommando `DEplot`, `method=rkf45`

Wir zeigen also noch einige Verläufe von Näherungsfunktionen im Intervall $[-4, 20]$ zu den AB pts mit den oben genannten numerischen Verfahren. Die Verfahren hoher Genauigkeit erzeugen akzeptable Näherungen, die niedriger Genauigkeit beginnen schon bald wegzudriften und können auch an den stabilen Gleichgewichtslagen die gewünschten Kurven (asymptotische Verläufe) nicht “halten“.

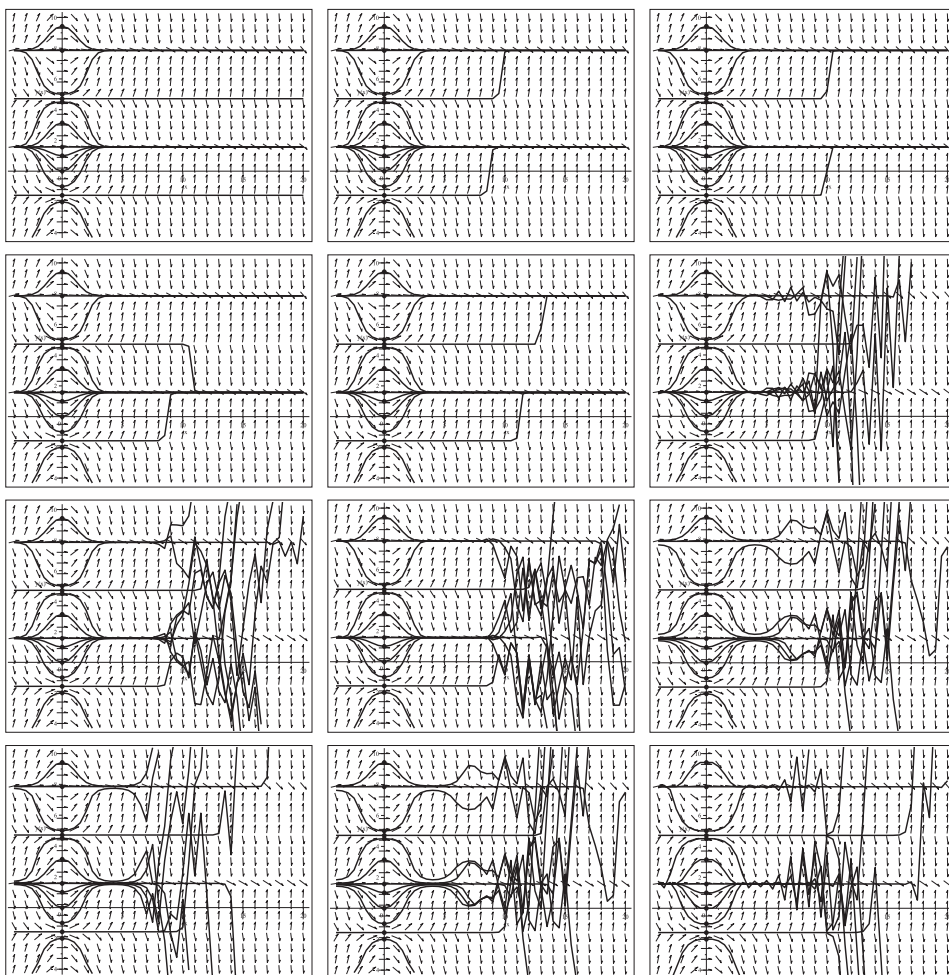


Abb. 2.101 Dateien *dber_loes_05b01..12.ps*, Richtungsfeld zur gDGl (5),
 Näherungslösungen zu den AB pts in $[-4, 20]$,
 Kommando DEplot, Methoden von l.o.n.r.u.
`method = rosenbrock,dverk78,rkf45,`
`gear,ck45,classical[abmoulton],`
`classical[rk4,rk3,rk2],`
`classical[impoly,heunform,foreuler]`

Wir kontrollieren noch einige Verläufe von Näherungsfunktionen mit AB $y(0) = y_k = (k + 1/2)\pi$, $k = 0, 1, 2$, die mit dem Kommando `dsolve/numeric` erzeugt werden.

$y(x) = (2k + 1/2)\pi = \text{const}$, $x > 0$, sind stabile stationäre und anziehende Lösungen.

$y(x) = (2k - 1/2)\pi = \text{const}$, $x > 0$, sind instabile stationäre Lösungen.

Dabei sollen die Ergebnisse zum Vergleich in einem groben bzw. feinen Maßstab angezeigt werden. Wackeleffekte bei diesen Kurven sind bei grobem Maßstab nicht erkennbar.

```
> dsol5:=dsolve({dgl,y(0)=Pi/2},y(x), # y(0)=3*Pi/2}, y(0)=5*Pi/2}
      numeric,method=rkf45,          # Standard
      range=-30..30,                # dann nur rkf45 verwendbar
      output=procedurelist):        # Standard
pl1:=odeplot(dsol5,color=black,thickness=2,view=[-30..30,-2..2]):
display(pl1);                       # grobes Fenster

> pl1:=odeplot(dsol5,color=black,thickness=3): # ohne view
pl3:=plot(Pi/2,x=-30..30,color=black,linestyle=dot):
pl4:=textplot([6,Pi/2,' Pi/2'],color=black,font=[HELVETICA,BOLD,9]):
display(pl1,pl2,pl3,pl4); # kleines Fenster, direkt betroffener Bereich
```

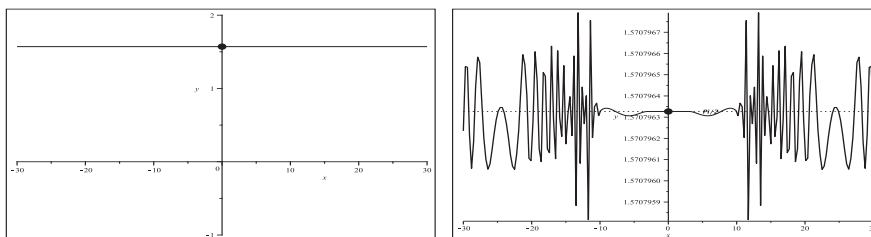


Abb. 2.102 Dateien *dber_Joes_05c11..12.ps*, gDGI (5),
Näherungslösung zur AB $y(0) = \frac{\pi}{2}$ (stabile Lage für $x > 0$)
im Fenster groß/klein,
Kommandos `dsolve/numeric/rkf45`, `odeplot`, `plot`, `textplot`

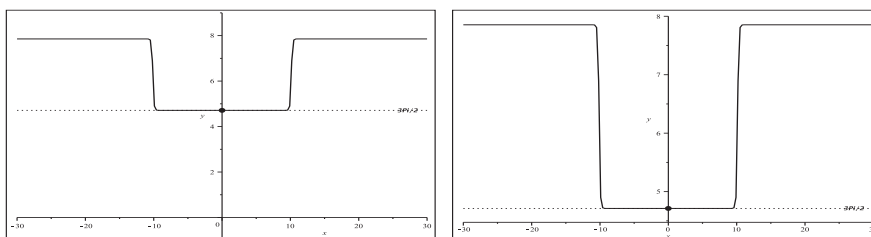


Abb. 2.103 Dateien *dber_Joes_05c21..22.ps*, gDGI (5),
Näherungslösung zur AB $y(0) = \frac{3\pi}{2}$ (instabile Lage für $x > 0$)
im Fenster groß/klein,
Kommandos `dsolve/numeric/rkf45`, `odeplot`, `plot`, `textplot`

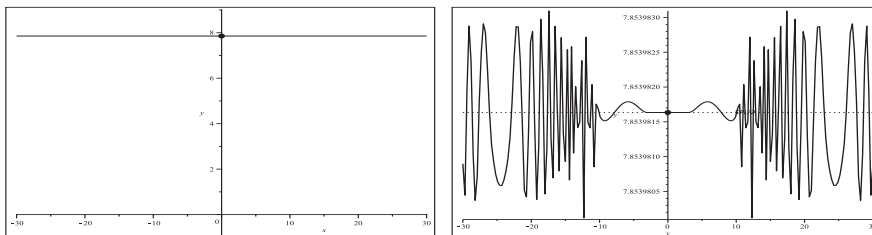


Abb. 2.104 Dateien *dber_loes_05c31..32.ps*, gDGI (5),
 Näherungslösung zur AB $y(0) = \frac{5\pi}{2}$ (stabile Lage für $x > 0$)
 im Fenster groß/klein,
 Kommandos `dsolve/numeric/rkf45`, `odeplot`, `plot`, `textplot`

Die analytische Lösung mittels `dsolve` versagt bei unbestimmten Werten y_0 in der AB. Ähnlich gibt es keine Lösungen im Fall `dsolve(dgl,y(x0)=yk,y(x))` bei unbestimmten Werten x_0 und $y_k = (k + \frac{1}{2})\pi$ mit Ausnahme von $y_0 = \frac{\pi}{2}$. Sonst werden Lösungen berechnet. Ein einfacher Fall ist `dsolve(dgl,y(x0)=0,y(x))` mit

$$y(x) = \arctan\left(\frac{e^{x^2-x_0^2} - 1}{2e^{x^2-x_0^2} + 1}, \frac{e^{x^2/2-x_0^2/2}}{e^{x^2-x_0^2} + 1}\right).$$

Gilt bei $x_0 = 0$ die Bedingung $y_0 \in ((2k+1)\pi, (2k+3/2)\pi)$, k ganzzahlig, so treten im Kurvenverlauf der Lösung des AWP symmetrische Sprünge der Höhe 2π auf.

```
> dsolve({dgl=0,y(0)=-2},y(x)); f1:=rhs(%);
# plot(f1,x=-4..4,y=-4..10,color=black,thickness=3); # Kontrolle
# analog f2,...,f12

> p11:=dfieldplot(dgl=0,y,x=-4..4,y=-4..10,color=black,dirgrid=[21,21]):
p12:=plot([f3,f4,f5,f7,f10,f12],x=-4..4,y=-4..10,thickness=3,color=black):
p13:=plot([f2,f6,f9,f11],x=-4..4,y=-4..10,thickness=3,color=black,
linestyle=dash):
p14:=plot([f1,f8],x=-4..4,y=-4..10,thickness=3,color=black,
linestyle=dashdot):
p15:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12,p13,p14,p15);

> # 5 Werte y0 aus dem Intervall (Pi,3Pi/2)^(3.14,4.71)
dsolve({dgl=0,y(0)=3.2},y(x)); f1:=rhs(%);
dsolve({dgl=0,y(0)=3.5},y(x)); f2:=rhs(%);
dsolve({dgl=0,y(0)=4.0},y(x)); f3:=rhs(%);
dsolve({dgl=0,y(0)=4.5},y(x)); f4:=rhs(%);
dsolve({dgl=0,y(0)=4.7},y(x)); f5:=rhs(%);
> p11:=dfieldplot(dgl=0,y,x=-4..4,y=-1..10,color=black,dirgrid=[21,21]):
p12:=plot([f1,f2,f3,f4,f5],x=-4..4,y=-1..10,thickness=3,color=black):
p13:=pointplot([ [0,3.2], [0,3.5], [0,4.0], [0,4.5], [0,4.7] ],
color=black,symbol=solidcircle,symbolsize=16):
```

```

p14:=plot([[0,Pi],[0,3*Pi/2]],color=black,thickness=4):
p15:=plot([Pi,3*Pi],x=-4..4,thickness=1,color=black,linestyle=dot):
p16:=plot([Pi/2,3/2*Pi,5/2*Pi],x=-4..4,thickness=1,color=black):
display(p11,p12,p13,p14,p15,p16);

```

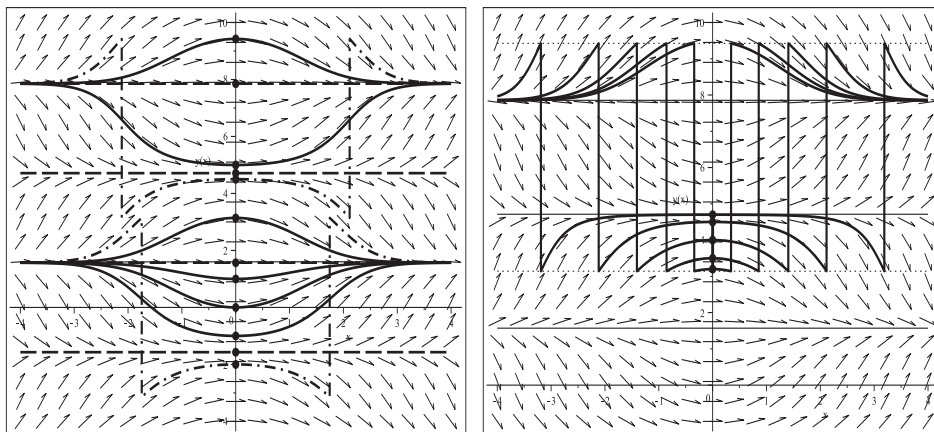


Abb. 2.105 Dateien *dber_Loes_05d1..2.ps*, Richtungsfeld zur gDGL (5),
 Lösungen zu den AB pts bzw. $[[0, 3.2], [0, 3.5], [0, 4.0], [0, 4.5], [0, 4.7]]$,
 Kommandos `dsolve`, `dfieldplot`, `plot`, `pointplot`

$$(6) \quad y' = e^y \sin^2(x) \cos(x)$$

$$y(x) = \ln(12) - \ln(\sin(3x)) - 3 \sin(x) - 12c$$

Im Abschnitt 2.2.3, Bsp. 2.4 Teil (7), ist die gDGL schon betrachtet worden, sodass hier nur kurz darauf eingegangen wird.

Mit der AB $y(0) = y_0 \geq \ln(3)$ hat die Lösung des zugehörigen AWP Polstellen, über die dann nicht hinweg integriert werden kann, sowie "Löcher" im Definitionsbereich.

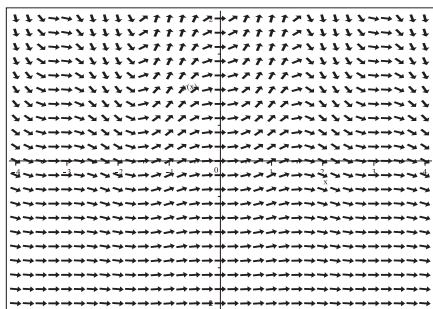


Abb. 2.106
 Datei *dber_Loes_06a.ps*,
 Richtungsfeld zur gDGL (6),
 Kommando `dfieldplot`

Die numerische Lösung berechnet man mit dem Kommando `DEplot`.

Zu allen AB $y(x_0) = y_0$ gibt es numerische Lösungen. Für gute Näherungen ist die Option `stepsize` zu verwenden.

```

> # AB
pts:=[[0,-1],[0,-1/2],[0,0],[0,1/2],[0,1],[0,ln(3)],[0,3/2]]:
pts1:=[[0,-1],[0,-1/2],[0,0],[0,1/2],[0,1],[0,3/2]]:

> p11:=DEplot(dgl,y(x),x=-2..10,y=-3/2..6,pts1,color=black,
             linecolor=black,dirgrid=[25,21]): # rkf45
dsol6:=dsolve({dgl,y(0)=ln(3)},y(x),range=-2..10,numeric): # rkf45
p12:=odeplot(dsol6,color=black,thickness=2,linestyle=dash):
p13:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12,p13);

```

Für $y(0) = \ln(3)$ erhält man den Grenzfall und die Polstelle bei $\frac{\pi}{2}$. Bei $y(0) = \frac{3}{2}$ ergibt sich das ‐Loch‐ [1.064, 2.076] mit Polstellen an den Intervallgrenzen. Das Ganze wiederholt sich 2π -periodisch.

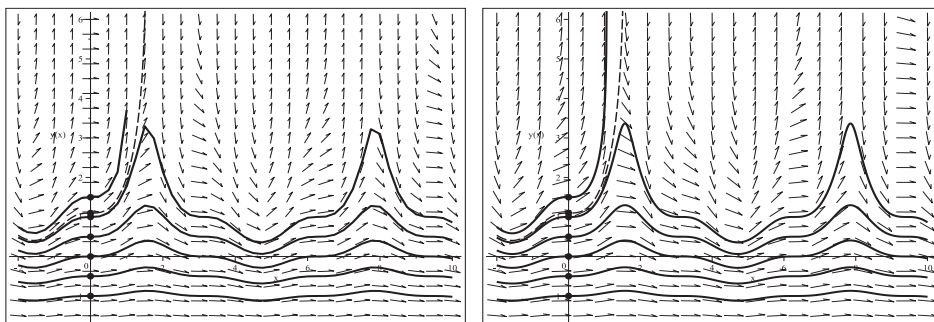


Abb. 2.107 Dateien *dber_loes_06b1..2.ps*, Richtungsfeld zur gDGl (6),
Näherungslösungen zu den AB pts,
Kommando DEplot, method=rkf45 ohne/mit stepsize=0.01

$$(7) \quad y' = \tan^2(x)$$

$$y(x) = \tan(x) - x + c$$

Das Richtungsfeld ist π -periodisch bzgl. x mit Anstiegen der Linienelemente aus $[0, \infty)$. An den Werten $x = x_k = (k + \frac{1}{2})\pi$, k ganzzahlig, hat der Tangens Polstellen und man würde vertikale Linienelemente zeichnen.

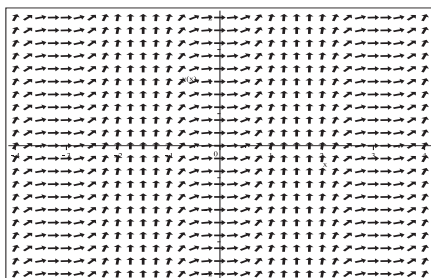


Abb. 2.108
Datei *dber_loes_07a.ps*,
Richtungsfeld zur gDGl (7),
Kommando dfieldplot

Die numerische Lösung berechnet man mit dem Kommando `DEplot`.

Zu allen AB $y(x_0) = y_0$, $x_0 \neq x_k$, gibt es numerische Lösungen. Über die Polstellen kann nicht hinweg integriert werden. Und es macht nicht viel Sinn, die AB an die Polstellen zu legen (man bekommt Fehlermeldungen, lange Rechnungen, Grenzwerte der Parameter werden überschritten, keine Lösungen).

Für gute Näherungen ist die Option `stepsize` zu verwenden.

```
> # AB
pts:= [[-4,1], [0,0], [0,4], [4,-1]]:
pts1:= [[0,-1], [0,0], [0,1]]:
pts2:= [[-4,1], [0,-1], [0,0], [0,1], [0,4], [4,-1]]:
> p11:=DEplot(dg1,y(x),x=-5..5,y=-10..10,pts,color=black,linecolor=black,
dirgrid=[25,25]): # rkf45
p12:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12);
```

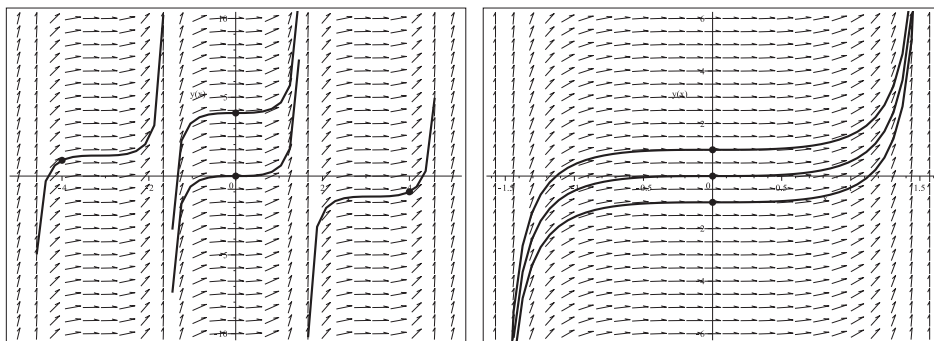


Abb. 2.109 Dateien *dber_Joes_07b1..2.ps*, Richtungsfeld zur gDGL (7),
Näherungslösungen zu den AB pts in $[-5, 5]$ bzw. pts1 in $[-\frac{\pi}{2}, \frac{\pi}{2}]$,
Kommando `DEplot`, `method=rkf45`

Die analytische Lösung ermittelt man mittels `dsolve` für alle AB $y(x_0) = y_0$ mit $x_0 \neq x_k$. Das allgemeine Kommando zum AWP `dsolve({dg1,y(x0)=y0},y(x))` liefert die Lösung $y(x) = \tan(x) - x + y_0 - \tan(x_0) + x_0$.

Die Lösungskurven sind π -periodisch.

```
> p11:=dfieldplot(dg1=0,y,x=-5..5,y=-6..6,color=black,dirgrid=[31,25]):
p12:=plot([f1,f2,f3,f4,f5,f6],x=-5..5,y=-6..6,thickness=3,color=black,
linestyle=dash,discont=true):
p131:=plot(f1,x=-3*Pi/2..-Pi/2,y=-6..6,thickness=3,color=black):
p132:=plot([f2,f3,f4,f5],x=-Pi/2..Pi/2,y=-6..6,thickness=3,color=black):
p133:=plot(f6,x=Pi/2..3*Pi/2,y=-6..6,thickness=3,color=black):
p14:=plot([seq([(-3/2+k)*Pi,-6],[(-3/2+k)*Pi,6]),k=0..3],color=black):
p15:=pointplot(pts2,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12,p131,p132,p133,p14,p15);
```

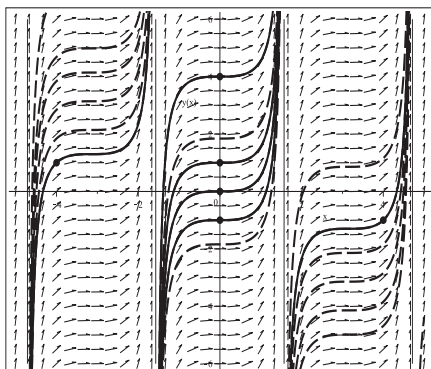



Abb. 2.110

Datei *dber_Loes_07c.ps*,
 Richtungsfeld zur gDGl (7),
 Lösungen zu den AB pts2,
 Kommandos `dsolve`,
`dfieldplot`, `plot`, `pointplot`

$$(8) \quad y' = e^y - \sin(x)$$

$$y(x) = \cos(x) - \ln\left(c - \int e^{\cos(x)} dx\right)$$

Es gibt keine allgemeine explizite Lösung der gDGl, denn in der Formel bleibt stets das Integral stehen.

Das Richtungsfeld ist überall definiert und 2π -periodisch bzgl. x .

Für $y \gg 0$ sind die absoluten Anstiege sehr groß, was ein Hinweis auf Asymptoten und Löcher im Definitionsbereich ist, für $y \ll 0$ haben wir moderate Anstiege wie $\sin(x)$.

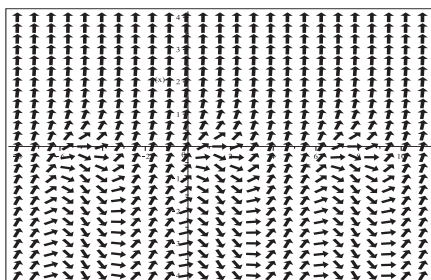


Abb. 2.111

Datei *dber_Loes_08a.ps*,
 Richtungsfeld zur gDGl (8),
 Kommando `dfieldplot`

Die numerische Lösung berechnet man mit dem Kommando `DEplot`.

Zu allen AB $y(x_0) = y_0$ gibt es welche. In kleinen Lösungsintervallen funktioniert alles recht gut. Aber bei großen Intervallen mit Polstellen (Asymptoten) kann nicht über diese hinweg integriert werden. Die Definitionsbereiche verkleinern sich dadurch entsprechend.

Geraten die numerischen Verfahren in die Bereiche mit sehr steilen Linienelementen, wird es problematisch für diese.

Mit der AB $y(-0.81102472) = 0$ erhält man fast die Asymptote bei $x = 0$. Das Verfahren `dsolve/numeric/rkf45` rechnet sehr nahe an die Null heran und meldet

```
Warning, cannot evaluate the solution further right of -0.77563791e-9,
probably a singularity
```

```

> # AB
pts:=[[0,-3/2],[0,-1],[0,-1/2],[0,0],[0,1],[-0.81102472,0]]:
pl1:=DEplot(dgl,y(x),x=-8..11,y=-4..4,pts,color=black,linecolor=black,
            dirgrid=[25,25],stepsize=0.01): # rkf45
pl2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(pl1,pl2);

> dsol8:=dsolve({dgl,y(-0.81102472)=0},y(x),range=-1..0,
               numeric,initstep=0.0001): # rkf45
pl2:=odeplot(dsol8,color=black,thickness=2,linestyle=solid):
pl3:=pointplot([-0.81102472,0],color=black,symbol=solidcircle,
               symbolsize=16):
display(pl2,pl3,view=[-1..0,-1..30]);

```

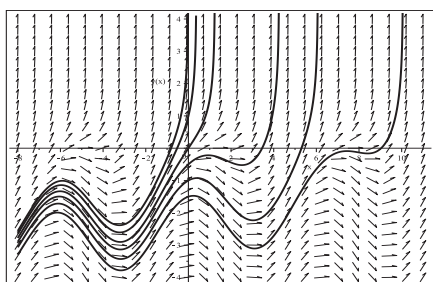


Abb. 2.112

Datei *dber_loes_08b.ps*,
 Richtungsfeld zur gDGL (8),
 Näherungslösungen zu den
 AB gemäß pts,
 Kommando DEplot, method=rkf45

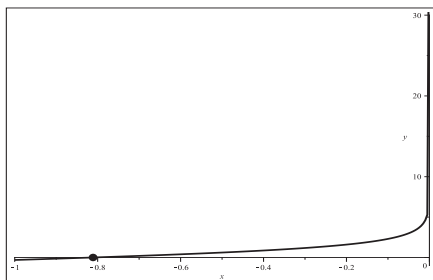


Abb. 2.113

Datei *dber_loes_08c.ps*,
 Näherungslösung zum AWP (8)
 zur AB $y(-0.81102472) = 0$,
 Kommando dsolve/numeric/rkf45

Die analytische Lösung mittels `dsolve(dgl,y(x0)=y0,y(x))` liefert nur die Formel mit dem Integralterm

$$y(x) = \cos(x) - \ln \left(e^{-y_0 + \cos(x_0)} - \int_{x_0}^x e^{\cos(z)} dz \right)$$

und keine explizite Lösung, auch nicht für konkrete Werte x_0, y_0 .

Aber das Auftreten des Logarithmus zeigt die Polstelle, wenn sein Argument zu Null wird und das ‐Loch‐ bei negativen Argumenten. Diese Situation lässt sich für jede AB leicht numerisch austesten, wie im Fall $y(0) = -\frac{3}{2}$ mit der Lösung

$$y(x) = \cos(x) - \ln \left(e^{5/2} - \int_{x_0}^x e^{\cos(z)} dz \right)$$

gezeigt wird. Analog kann man es für die anderen fünf AB in pts machen.

```

> dsolve({dgl=0,y(0)=-3/2},y(x)); f1:=rhs(%):
  pl1:=plot(f1,x=-8..10,color=black,thickness=2):
  display(pl1);
> g:=unapply(exp(2.5)-int(exp(cos(z)),z=0..x),x);
  # solve(g(x),x); # Fehler, keine Loesung
  as1:=fsolve(g(x),x=10..11);
  pl1:=plot(g(x),x=0..11,color=black,thickness=2):
  pl2:=pointplot([as1,0],color=black,symbol=solidcircle,symbolsize=20):
  display(pl1,pl2);

```

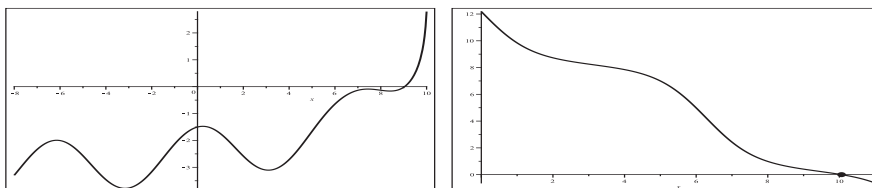


Abb. 2.114 Dateien *dber_loes_08c1..2.ps*, gDGI (8),
 Lösung zur AB $y(0) = -\frac{3}{2}$ in $[-8, 10]$ sowie Argumentfunktion
 des $\ln(\cdot)$ mit der Nullstelle bei $x \approx 10.060149885$,
 Kommandos `dsolve`, `fsolve`, `plot`

Um mit der AB $y(x_0) = 0$ bei $x_0 < 0$ die y -Achse als Asymptote zu erhalten, macht man Folgendes.

```

> dsolve({dgl=0,y(-0.81102472)=0},y(x)); f6:=rhs(%): # ausgetestet
  pl1:=plot(f6,x=-8..10,color=black,thickness=2):
  display(pl1);
> dsolve({dgl=0,y(x0)=0},y(x)); f6a:=rhs(%): # x0<0
> g:=unapply(exp(cos(x0))-int(exp(cos(z)),z=x0..x),x0);
  # solve(g(x0),x0); # analytisch nicht loesbar
  x06:=fsolve(g(x0),x0=-1..0);
  as6:=0; # Stelle der Asymptote
  pl1:=plot(g(x0),x0=-1..0,color=black,thickness=2):
  pl2:=pointplot([x06,0],color=black,symbol=solidcircle,symbolsize=20):
  display(pl1,pl2);

```

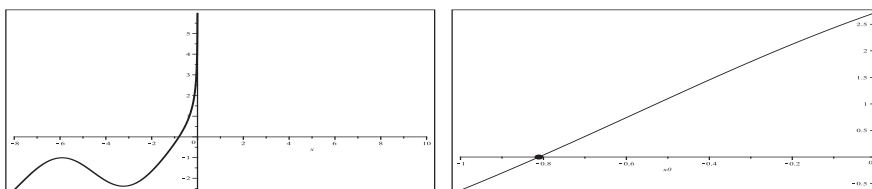


Abb. 2.115 Dateien *dber_loes_08c3..4.ps*, gDGI (8),
 Lösung mit der AB $y(-0.81102472) = 0$ in $[-8, 10]$ sowie Argumentfunktion
 $g(x_0) = e^{\cos(x_0)} - \int_{x_0}^0 e^{\cos(z)} dz$ des $\ln(\cdot)$ mit Nullstelle bei -0.811 ,
 Kommandos `dsolve`, `fsolve`, `plot`

Damit haben wir zu den AB pts der Reihe nach die sechs Polstellen

```
as:=[10.06014988535821, 6.073489823790732, 4.254179829221040,
      1.243860047248152, 0.3765307392534185, 0]
```

und die sechs Lösungskurven mit den entsprechenden Definitionsbereichen $(-\infty, \text{as}[k])$, $k = 1, 2, \dots, 6$.

```
> p11:=dfieldplot(dg1=0,y,x=-8..11,y=-4..4,color=black,dirgrid=[25,25]):
p12:=plot([f1,f2,f3,f4,f5,f6],x=-8..11,y=-4..4,thickness=3,color=black):
p13:=plot([seq([[as[k],-4],[as[k],4]],k=1..6)],color=black):
p14:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12,p13,p14);
```

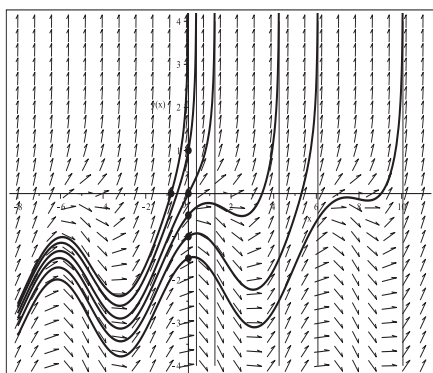


Abb. 2.116

Datei *dber_loes_08d.ps*,
Richtungsfeld zur gDGI (8),
Lösungen zu den AB pts,
Kommandos *dsolve*,
dfieldplot, *plot*, *pointplot*

$$(9) \quad y' = x e^{2+y}$$

$$y(x) = -2 - \ln(c - x^2/2)$$

Die rechte Seite der gDGI sowie ihre allgemeine Lösung verweisen auf absolut sehr große Anstiege der Linienelemente im überall definierten Richtungsfeld, sowie damit auf Asymptoten und Löcher im Definitionsbereich. Für $y \ll -2$ gibt es moderate oder kleine Anstiege, die noch von x abhängig sind.

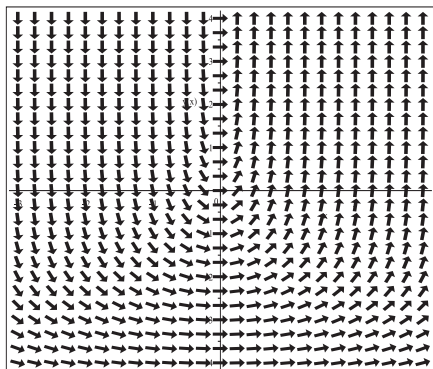


Abb. 2.117

Datei *dber_loes_09a.ps*,
Richtungsfeld zur gDGI (9),
Kommando *dfieldplot*

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$ und man berechnet sie mit dem Kommando `DEplot`.

In kleinen Lösungsintervallen funktioniert alles recht gut, wenn also der x -Bereich keine Polstellen von Lösungen enthält. Aber bei großen Intervallen mit Polstellen (Asymptoten) kann nicht über diese hinweg integriert werden. Die Definitionsbereiche verkleinern sich auf endliche Intervalle. Geraten die numerischen Verfahren in die Bereiche mit sehr steilen Linienelementen, wird es problematisch für diese.

Bei Methoden niedriger Genauigkeit, beginnend bei `classical[rk4]`, ist in `DEplot` die Angabe des y -Bereichs für eine Grafik entscheidend, nicht die Option `stepsize`.

```
> # AB
pts:=[[0,-1],[0,-1/2],[0,0],[0,1]]:
> p1:=DEplot(dgl,y(x),x=-2..2,pts,
#method=classical[rk4],stepsize=0.01, # keine Grafik
method=rkf45,
color=black,linecolor=black):
p2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p1,p2);
> p1:=DEplot(dgl,y(x),x=-1..1,y=-1..3,pts,color=black,linecolor=black):
display(p1,p2);
```

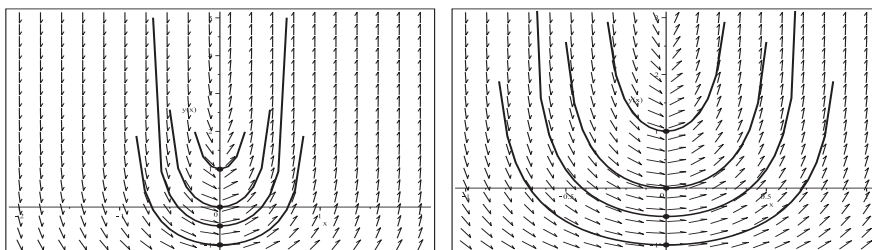


Abb. 2.118 Dateien `dber_loes_09b01..02.ps`, Richtungsfeld zur gDGl (9), Näherungslösungen zu den AB $y(0) = -1, -\frac{1}{2}, 0, 1$, $x \in [-2, 2]$ bzw. $(x, y) \in [-1, 1] \times [-1, 3]$, Kommando `DEplot`, `method=rkf45`

Die analytische Lösung mittels `dsolve(dgl,y(x0)=y0,y(x))` liefert die Formel

$$y(x) = -2 - \ln\left(\frac{x_0^2}{2} + e^{-2-y_0} - \frac{x^2}{2}\right),$$

die Asymptoten an den Stellen $x = \pm c$, $c = \sqrt{x_0^2 + 2e^{-2-y_0}}$ und somit den endlichen Definitionsbereich $(-c, c)$ hat. Damit haben wir zu den AB `pts` der Reihe nach die Polstellen `as[k]`, $k=1, 2, \dots, 8$, mit den Werten

$$\begin{aligned} \pm\sqrt{\frac{2}{e}} &= \pm 0.8577638849607069, \quad \pm\sqrt{\frac{2}{e^{3/2}}} = \pm 0.6680271852977689, \\ \pm\sqrt{\frac{2}{e}} &= \pm 0.5202600950228890, \quad \pm\sqrt{\frac{2}{e^3}} = \pm 0.3155536986563902. \end{aligned}$$

```

> dsolve({dgl=0,y(0)=-1},y(x)); f1:=rhs(%):
p1:=plot(f1,x=-1..1,color=black,thickness=3):
display(p1);
# analog f2,f3,f4

> p1:=dfieldplot(dgl=0,y,x=-1..1,y=-1..3,color=black,dirgrid=[25,25]):
p2:=plot([f1,f2,f3,f4],x=-1..1,y=-1..3,thickness=3,color=black):
p3:=plot([seq([as[k],-1],[as[k],3],k=1..8)],color=black):
p4:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p1,p2,p3,p4);

```

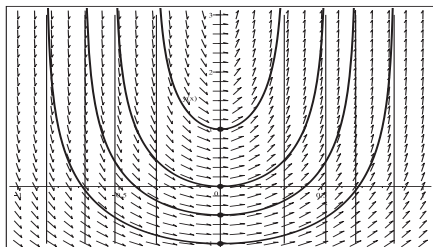


Abb. 2.119

Datei *dber_Loes_09c.ps*,
 Richtungsfeld zur gDGL (9),
 Lösungen zu den AB $y(0) = -1, -\frac{1}{2}, 0, 1$,
 Kommandos `dsolve`,
`dfieldplot`, `plot`, `pointplot`

$$(10) \quad y' = 2xy(y-1)$$

$$y(x) = 1/(1 + ce^{x^2/2})$$

Das Richtungsfeld der gDGL ist überall definiert.

Man bemerkt sofort die stationären Lösungen $y(x) = 0$ und $y(x) = 1$.

Die rechte Seite der gDGL sowie ihre allgemeine Lösung verweisen neben moderaten Lösungen bei $c > 0$ auch auf absolut sehr große Anstiege der Linienelemente sowie damit auf Polstellen und Asymptoten. Zu letzteren sind Definitionsbereiche der Form $(-d, d)$, $(-\infty, -d)$ bzw. (d, ∞) zu erwarten.

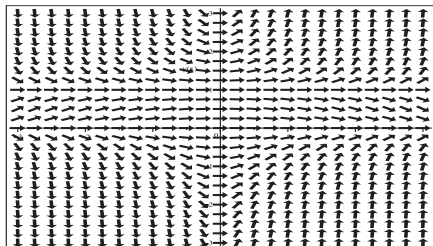


Abb. 2.120

Datei *dber_Loes_10a.ps*,
 Richtungsfeld zur gDGL (10),
 Kommando `dfieldplot`

```

> # AB
pts:=[[0,-3],[0,-2],[0,-1],[0,0],[0,1/2],[0,1],[0,2],[0,3],
[1,-1],[1,-1/(sqrt(exp(1))-1)],[1,-3],[-2,-3]]:

> p11:=DEplot(dgl,y(x),x=-3..3,y=-4..4,pts,dirgrid=[25,25],
method=classical[rk4], # stepsize=0.01, # Chaos ausserhalb des DB
#method=rkf45, # stepsize=0.01,
#method=gear, # stepsize=0.01,

```

```

color=black,linecolor=black):
pl2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(pl1,pl2);

```

Wenn man in den Parametern einen großen y -Bereich oder auch keinen angibt, dann verweisen die Warnungen des numerischen Verfahrens in DEplot auf die vorhandenen Polstellen der Lösungen hin. Über diese kann nicht hinweg integriert werden.

Zur AB $y(1) = -1/(\sqrt{e} - 1) = -1.5414940825368$ ist genau die y -Achse die Asymptote und $y(x) = -1/(e^{x^2/2} - 1)$ die Lösung.

```

Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the sol. further right of 1.1774096, probably a singularity
cannot evaluate the sol. further left of -1.1774096, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the sol. further right of .90051638, probably a singularity
cannot evaluate the sol. further left of -.90051638, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the sol. further left of .65164086, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the sol. further right of -1.850577, probably a singularity

```

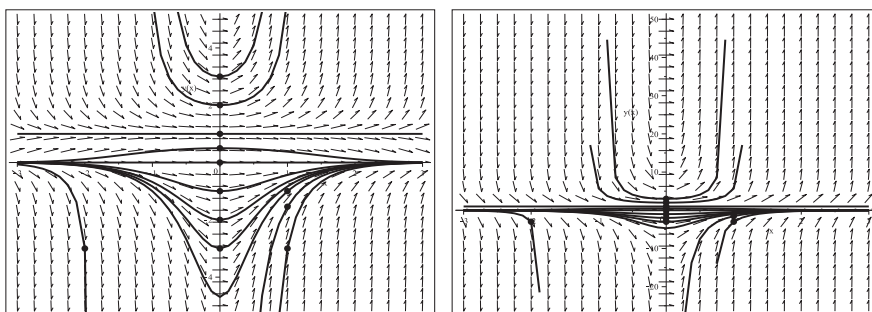


Abb. 2.121 Dateien *dber_Loes_10b1..2.ps*, Richtungsfeld zur gDGl (10), Näherungslösungen zu den AB pts, $(x, y) \in [-3, 3] \times [-5, 5]$ bzw. $[-3, 3] \times [-25, 50]$, Kommando DEplot, method=rkf45

Die analytische Lösung mittels `dsolve(dg1, y(x0)=y0, y(x))` liefert die Formel

$$y(x) = -\frac{y_0 e^{x_0^2/2}}{-y_0 e^{x_0^2/2} - (1 - y_0) e^{x^2/2}} = \frac{1}{1 + (1 - y_0)/y_0 e^{(x^2 - x_0^2)/2}},$$

speziell für die AB $y(0) = y_0$ die Lösung

$$y(x) = \frac{1}{1 + (1 - y_0)/y_0 e^{x^2/2}}.$$

Zur AB $y(x_0) = 0$ wird die stationäre Lösung $y(x) = 0$ erzeugt.

Fallunterscheidung bezüglich der Definitionsbereiche(a) AB $y(0) = y_0 = 1, 0$ Stationäre Lösungen sind $y(x) = 1$ bzw. $y(x) = 0$.(b) AB $y(0) = y_0 \in (-\infty, 0) \cup (0, 1)$ Moderate Lösung ist $y(x) = 1/(1 + (1 - y_0)/y_0 e^{x^2/2})$, weil der Term $(1 - y_0)/y_0 < -1$ bzw. $(1 - y_0)/y_0 > 0$ ist, und damit der Nenner überall definiert ist.(c) AB $y(x_0) = y_0 > 1$ Lösung $y(x)$ ist im symmetrischen Intervall $(-d, d)$, wobei $d = \sqrt{x_0^2 + 2 \ln(y_0/(y_0 - 1))}$. Die Asymptoten liegen bei $\pm d$.(d) AB $y(x_0) = y_0 < 0$ bei $x_0 \neq 0$ Ausgangspunkt ist die Betrachtung des Nennerterms des allgemeinen AWP mit der Bedingung $0 = 1 + (1 - y_0)/y_0 e^{(x^2 - x_0^2)/2}$. Daraus folgt nach Umstellung

$$x = \sqrt{x_0^2 + 2 \ln(y_0/(y_0 - 1))} = \sqrt{x_0^2 - 2 \ln((y_0 - 1)/y_0)}.$$

Um die Polstelle genau an der Stelle $x = 0$ zu erhalten, folgt für die Parameter in der AB die Beziehung

$$y_0 = -1/(e^{x_0^2/2} - 1) = y_g < 0.$$

(d1) AB $y(x_0) = y_0 \in (y_g, 0)$ bei $x_0 \neq 0$ Der Radikand $x_0^2 + 2 \ln(y_0/(y_0 - 1))$ wird negativ und damit gibt es keine Polstellen.Der Definitionsbereich der Lösung ist $(-\infty, \infty)$.(d2) AB $y(x_0) = y_0 = y_g$ bei $x_0 \neq 0$ Der Radikand $x_0^2 + 2 \ln(y_0/(y_0 - 1))$ verschwindet und damit liegt die Polstelle bei Null.Der Definitionsbereich der Lösung ist gemäß dem Vorzeichen von x_0 entweder $(-\infty, 0)$ oder $(0, \infty)$.(d3) AB $y(x_0) = y_0 < y_g$ bei $x_0 \neq 0$ Mit dem Radikanden wird die Polstelle $d = \sqrt{x_0^2 + 2 \ln(y_0/(y_0 - 1))}$, $x_0 > 0$, und der Definitionsbereich (d, ∞) bzw. bei $x_0 < 0$ der Bereich $(-\infty, -d)$ ermittelt.

Aus den obigen AB und weiteren erhält man folgende Definitionsbereiche.

$$y(0) = 2, \quad d = \sqrt{2 \ln(2)} = 1.1774100225154, \quad (-d, d),$$

$$y(0) = 3, \quad d = \sqrt{2 \ln(3/2)} = 0.9005166385005492, \quad (-d, d),$$

$$y(0) = 4, \quad d = \sqrt{2 \ln(4/3)} = 0.7585276164409318, \quad (-d, d),$$

$$y(0) = 10, \quad d = \sqrt{2 \ln(10/9)} = 0.4590436050264205, \quad (-d, d),$$

$$y(-1) = -1/(\sqrt{e} - 1), \quad d = 0, \quad (-\infty, d),$$

$$y(-2) = -3, \quad d = \sqrt{4 - 2 \ln(4/3)} = 1.850577168100925, \quad (-\infty, -d),$$

$$\begin{aligned}
 y(1) &= -1/(\sqrt{e} - 1) & d = 0, & & (d, \infty), \\
 y(1) &= -2, & d = \sqrt{1 - 2 \ln(3/2)} = 0.4348215539548048, & & (d, \infty), \\
 y(1) &= -3, & d = \sqrt{1 - 2 \ln(4/3)} = 0.6516408942787727, & & (d, \infty), \\
 y(1) &= -4, & d = \sqrt{1 - 2 \ln(5/4)} = 0.7441188731456691, & & (d, \infty), \\
 y(1) &= -10, & d = \sqrt{1 - 2 \ln(11/10)} = 0.8996552897590000, & & (d, \infty).
 \end{aligned}$$

Evtl. Polstellen zu den AB sind im Vektor `as` abgelegt.

Was die analytischen Lösungen der AWP mittels `dsolve(dg1,y(x0)=y0,y(x))` betrifft, so kann man diese mit allen Zweigen zeigen oder auch nur den relevanten Lösungszweig, evtl. noch dazu die Asymptoten.

```

> dsolve({dg1=0,y(0)=-3},y(x)); f1:=rhs(%);
ash:=solve(1+(1-(-3))/(-3)*exp((x^2-0^2)/2),x);
# imaginaer, d.h. keine Einschraenkung
p1:=plot(f1,x=-3..3,color=black,thickness=3):
display(p1);
# analog f2,f3,...,f11,f12

> dsolve({dg1=0,y(-2)=-3},y(x)); f12:=rhs(%);
ash:=solve(1+(1-(-3))/(-3)*exp((x^2-(-2)^2)/2),x);
# Polstelle bei -sqrt(4-2*ln(4/3))=-sqrt(4+2*ln(3/4))=-1.850577...
ash:=evalf(ash);
as5:=ash[2]:

> p1:=dfieldplot(dg1=0,y,x=-3..3,y=-5..5,color=black,dirgrid=[25,41]):
p2:=plot([f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12],x=-3..3,y=-5..5,
color=black,thickness=3,discont=true):
p8:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
p9:=plot({[[as[1],0],[as[1],5]],[[as[2],0],[as[2],5]],[[as[3],0],[as[3],5]],[[as[4],0],[as[4],5]],[[as[5],0],[as[5],-5]],[[as[6],0],[as[6],-5]],[[as[7],0],[as[7],-5]]},color=black):
display(p1,p2,p8,p9);

> p1:=dfieldplot(dg1=0,y,x=-3..3,y=-5..5,color=black,dirgrid=[25,41]):
p2:=plot([f1,f2,f3,f4,f5,f6,f9],x=-3..3,y=-5..5,
thickness=3,color=black):
p3:=plot(f7,x=-1.177..1.177,y=-5..5,thickness=3,color=black): # y(0)=2
p4:=plot(f8,x=-0.9..0.9,y=-5..5,thickness=3,color=black): # y(0)=3
p5:=plot(f10,x=0..3,y=-5..5,thickness=3,color=black):
# y(1)=-1/(sqrt(exp(1))-1)
p6:=plot(f11,x=0.652..3,y=-5..5,thickness=3,color=black): # y(1)=-3
p7:=plot(f12,x=-3..-1.851,y=-5..5,thickness=3,color=black): # y(-2)=-3
display(p1,p2,p3,p4,p5,p6,p7,p8,p9);

```

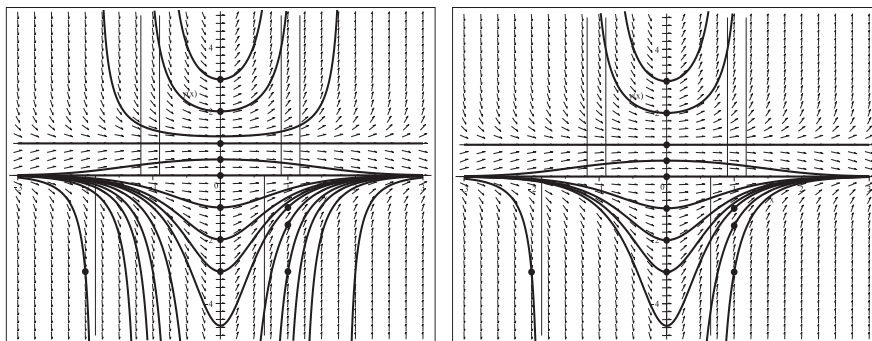


Abb. 2.122 Dateien *dber_Loes_10c1..2.ps*, Richtungsfeld zur gDGl (10),
Lösungen mit den AB pts, links “komplette“ Funktionen,
rechts nur relevante Lösungsbranche mit Asymptoten,
Kommandos `dsolve`, `dfieldplot`, `plot`, `pointplot`

$$(11) \quad y' = -2xy^2 \quad (\text{vergl. Beispiel 2.4 (15)})$$

$$y(x) = 1/(x^2 + c)$$

Das Richtungsfeld der gDGl ist überall definiert.

Man bemerkt sofort die stationäre Lösung $y(x) = 0$.

Die rechte Seite der gDGl sowie ihre allgemeine Lösung verweisen neben moderaten Lösungen bei $c > 0$ auch auf absolut sehr große Anstiege der Linienelemente sowie damit auf Polstellen und Asymptoten. Zu letzteren sind Definitionsbereiche der Form $(-d, d)$, $(-\infty, -d)$ bzw. (d, ∞) zu erwarten.

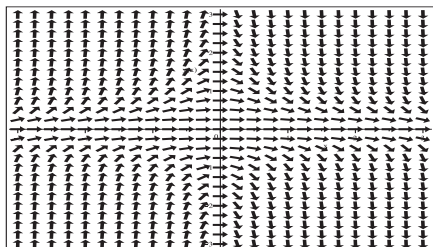


Abb. 2.123

Datei *dber_Loes_11a.ps*,
Richtungsfeld zur gDGl (11),
Kommando `dfieldplot`

```
> # AB
pts:=[[0, -3], [0, -2], [0, -1], [0, -1/2], [0, 0], [0, 1/2], [0, 1], [0, 2], [0, 3],
      [1, 1], [1, 3], [-2, 3]]:
> p11:=DEplot(dg1,y(x),x=-3..3,y=-4..4,pts,dirgrid=[25,33],
             method=rkf45,stepsize=0.01,
             color=black,linecolor=black):
p12:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12);
> p11:=DEplot(dg1,y(x),x=-3..3,y=-10..20,pts,dirgrid=[25,31],
             method=rkf45,color=black,linecolor=black):
display(p11,p12);
```

Wenn man in den Parametern einen großen y -Bereich oder auch keinen angibt, dann verweisen die Warnungen des numerischen Verfahrens in DEplot auf die vorhandenen Polstellen der Lösungen hin. Über diese kann nicht hinweg integriert werden.

Zur AB $y(1) = 1$ ist genau die y -Achse die Asymptote und $y(x) = 1/x^2$ die Lösung.

```
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the sol. further right of .70709745, probably a singularity
cannot evaluate the sol. further left of -.70709745, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the sol. further right of .99998643, probably a singularity
cannot evaluate the sol. further left of -.99998643, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the sol. further right of 1.4141933, probably a singularity
cannot evaluate the sol. further left of -1.4141933, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the sol. further left of .81649656, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the sol. further right of -1.914854, probably a singularity
```

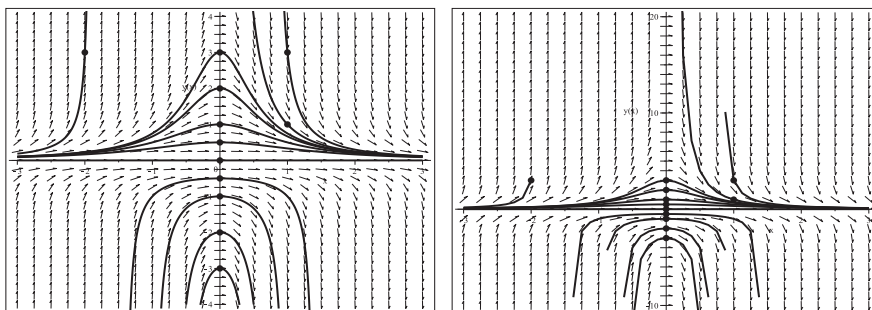


Abb. 2.124 Dateien *dber_loes_11b1.2.ps*, Richtungsfeld zur gDGI (11),
 Näherungslösungen zu den AB pts,
 $(x, y) \in [-3, 3] \times [-4, 4]$ bzw. $[-3, 3] \times [-10, 20]$,
 Kommando DEplot, method=rkf45

Die analytische Lösung mittels `dsolve(dgl, y(x0)=y0, y(x))` liefert die Formel

$$y(x) = -\frac{y_0}{1 + x^2 y_0 - x_0^2 y_0} = \frac{1}{1/y_0 - x_0^2 + x^2}, \text{ falls } y_0 \neq 0,$$

speziell für die AB $y(0) = y_0$ die Lösung

$$y(x) = \frac{y_0}{1 + x^2 y_0}.$$

Fallunterscheidung bezüglich der Definitionsbereiche

(a) AB $y(0) = y_0 = 0$

Die stationäre Lösung lautet $y(x) = 0$.

(b) AB $y(0) = y_0 \in (0, \infty)$

Die moderate Lösung die $y(x) = 1/(1/y_0 + x^2)$, weil der Nenner überall positiv ist.

(c) AB $y(x_0) = y_0 < 0$

Die Lösung $y(x)$ liegt im symmetrischen Intervall $(-d, d)$, wobei $d = \sqrt{x_0^2 - 1/y_0}$. Die Asymptoten sind bei $\pm d$.

(d) AB $y(x_0) = y_0 > 0$ bei $x_0 \neq 0$

Ausgangspunkt ist die Betrachtung des Nennerterms des allgemeinen AWP mit der Bedingung $0 = 1/y_0 - x_0^2 + x^2$. Daraus folgt nach Umstellung

$$x = \sqrt{x_0^2 - 1/y_0}.$$

Um die Polstelle genau an der Stelle $x = 0$ zu erhalten, folgt für die Parameter in der AB die Beziehung

$$y_0 = 1/x_0^2 = y_g > 0.$$

(d1) AB $y(x_0) = y_0 \in (0, y_g)$ bei $x_0 \neq 0$

Der Radikand $x_0^2 - 1/y_0$ wird negativ und damit gibt es keine Polstellen. Der Definitionsbereich der Lösung ist $(-\infty, \infty)$.

(d2) AB $y(x_0) = y_0 = y_g$ bei $x_0 \neq 0$

Der Radikand $x_0^2 - 1/y_0$ verschwindet und damit liegt die Polstelle bei Null.

Der Definitionsbereich der Lösung ist gemäß dem Vorzeichen von x_0 entweder $(-\infty, 0)$ oder $(0, \infty)$.

(d3) AB $y(x_0) = y_0 > y_g$ bei $x_0 \neq 0$

Mit dem Radikanden wird die Polstelle $d = \sqrt{x_0^2 - 1/y_0}$, $x_0 > 0$, und der Definitionsbereich (d, ∞) bzw. bei $x_0 < 0$ der Bereich $(-\infty, -d)$ ermittelt.

Aus den obigen AB und weiteren erhält man folgende Definitionsbereiche.

$$y(0) = -1/2, \quad d = \sqrt{2} = 1.414213562373095, \quad (-d, d),$$

$$y(0) = -1, \quad d = 1, \quad (-d, d),$$

$$y(0) = -2, \quad d = \sqrt{2}/2 = 0.7071067811865475, \quad (-d, d),$$

$$y(0) = -3, \quad d = \sqrt{3}/3 = 0.57735026918962, \quad (-d, d),$$

$$y(0) = -4, \quad d = 1/2, \quad (-d, d),$$

$$y(0) = -10, \quad d = \sqrt{10}/10 = 0.3162277660168379, \quad (-d, d),$$

$$y(1) = 1, \quad d = 0, \quad (0, \infty),$$

$$y(1) = 2, \quad d = \sqrt{2}/2 = 0.7071067811865475, \quad (d, \infty),$$

$$y(1) = 3, \quad d = \sqrt{6}/3 = 0.8164965809277259, \quad (d, \infty),$$

$$y(1) = 4, \quad d = \sqrt{3}/2 = 0.8660254037844385, \quad (d, \infty),$$

$$y(1) = 10, \quad d = 3\sqrt{10}/10 = 0.9486832980505137, \quad (d, \infty),$$

$$y(-2) = 3, \quad d = \sqrt{33}/3 = 1.914854215512676, \quad (-\infty, -d).$$

Evtl. Polstellen zu den AB sind im Vektor as abgelegt.

Was die analytischen Lösungen der AWP mittels $dsolve(dg1, y(x0)=y0, y(x))$ betrifft, so kann man diese mit allen Zweigen zeigen oder auch nur den relevanten Lösungszweig, evtl. noch dazu die Asymptoten.

```
> p1:=dfieldplot(dg1=0,y,x=-3..3,y=-4..4,color=black,dirgrid=[25,33]):
p2:=plot([f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12],x=-3..3,y=-4..4,
color=black,thickness=3,discont=true):
p10:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
p11:=plot({[[as[1],0],[as[1],-4]],[as[2],0],[as[2],-4]],
[[as[3],0],[as[3],-4]],[as[4],0],[as[4],-4]],
[[as[5],0],[as[5],-4]],[as[6],0],[as[6],-4]],
[[as[7],0],[as[7],-4]],[as[8],0],[as[8],-4]],
[[as[9],0],[as[9],4]],[as[10],0],[as[10],4]],
[[as[11],0],[as[11],4]]},color=black):
display(p1,p2,p10,p11);

> p11:=dfieldplot(dg1=0,y,x=-3..3,y=-4..4,color=black,dirgrid=[25,33]):
p12:=plot([f5,f6,f7,f8,f9],x=-3..3,y=-4..4,thickness=3,color=black):
p13:=plot(f1,x=-0.577..0.577,y=-4..4,thickness=3,color=black):# y(0)=-3
p14:=plot(f2,x=-0.707..0.707,y=-4..4,thickness=3,color=black):# y(0)=-2
p15:=plot(f3,x=-1..1,y=-4..4,thickness=3,color=black):# y(0)=-1
p16:=plot(f4,x=-1.414..1.414,y=-4..4,thickness=3,color=black):#y(0)=-1/2
p17:=plot(f10,x=0..3,y=-4..4,thickness=3,color=black):# y(1)=1
p18:=plot(f11,x=0.8165..3,y=-4..4,thickness=3,color=black):# y(1)=3
p19:=plot(f12,x=-3..-1.915,y=-4..4,thickness=3,color=black):# y(-2)=3
display(p11,p12,p13,p14,p15,p16,p17,p18,p19,p10,p11);
```

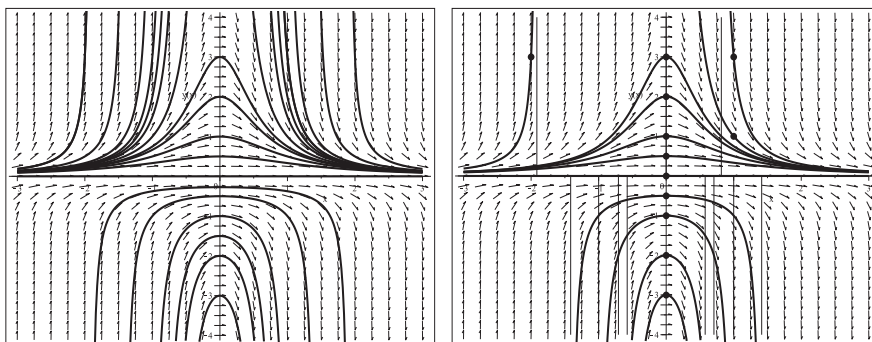


Abb. 2.125 Dateien *dber.loes.11c1..2.ps*, Richtungsfeld zur gDGL (11), Lösungen zu den AB pts, links “komplette“ Funktionen, rechts nur relevante Lösungszweige mit Asymptoten, Kommandos `dsolve`, `dfieldplot`, `plot`, `pointplot`

Für die verwandte gDGL $y' = 2xy^2$ kann man analoge Betrachtungen machen. Die Ergebnisse sind wie Spiegelbilder der obigen (Spiegelung an der x -Achse).

$$(12) \quad y' = y - 2x/y$$

$$y^2(x) = 1 + 2x + ce^{2x} \quad \text{bzw.} \quad y(x) = \pm\sqrt{1 + 2x + ce^{2x}}$$

Das Richtungsfeld der gDGL ist für $y \neq 0$ definiert. Der Koordinatenursprung stellt einen speziellen singulären Punkt dar.

Die rechte Seite der gDGL sowie ihre allgemeine implizite Lösung verweisen auf zwei separate Halbebenen mit Lösungen. Man erkennt die Grenzkurve $y(x)^2 = 1 + 2x$. Wenn das AWP seine AB von "innerhalb" nimmt, dann ist der Definitionsbereich ein endliches Intervall (d_1, d_2) , $d_1 \leq 0 < d_2$, mit vertikalen Linienelementen an den Intervallenden. Das ist der Fall von zwei Nullstellen der Funktion $1 + 2x + ce^{2x}$. Sonst hat die Lösungskurve das x -Intervall (d, ∞) , $d < 0$, und die Funktion $1 + 2x + ce^{2x}$ eine Nullstelle.

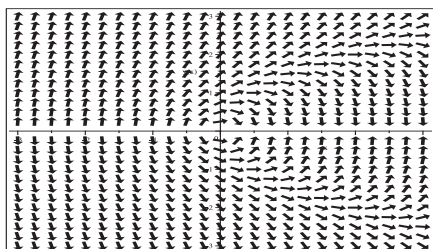


Abb. 2.126

Datei *dber_loes_12a.ps*,

Richtungsfeld zur gDGL (12),

Kommando `dfieldplot`

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0 \neq 0$ und wir berechnen sie mit dem Kommando `DEplot`. Mit der AB $y(x_0) = 0$ gibt es bei den verschiedenen Methoden keine Lösung, Fehlermeldungen oder den Versuch zur Lösung, jedoch mit Abbruch.

Geraten die numerischen Verfahren in die Bereiche mit sehr steilen Linienelementen, wird es problematisch für diese. Bei Methoden niedriger Genauigkeit, beginnend bei `classical[rk4]`, werden auch unbrauchbare Ergebnisse ("Chaos") außerhalb des Definitionsbereiches erzeugt. Da nutzt auch wenig die Angabe des y -Bereichs für eine Grafik sowie die Option `stepsize`. Es sind also Verfahren höherer Genauigkeit mit entsprechend kleiner Schrittweite `stepsize` zu empfehlen. Die Warnungen des numerischen Verfahrens in `DEplot` verweisen dann auf die vorhandenen Grenzen des Bereiches hin. Über diese wird nicht hinweg integriert.

```
> # AB
pts:=[[0, -3], [0, -2], [0, -1], [0, -1/2], [0, 1/2], [0, 1], [0, 2], [0, 3]]:
pts1:=[[0,0]]: pts2:=[[ -1/2,0]]:

> p11:=DEplot(dgl,y(x),x=-3..3,y=-4..4,pts,dirgrid=[25,25],
#method=classical[rk4], # stepsize=0.01, # Chaos ausserhalb des DB
#method=rkf45, # stepsize=0.01,
#method=gear, # stepsize=0.01,
color=black,linecolor=black):
p12:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12);
```

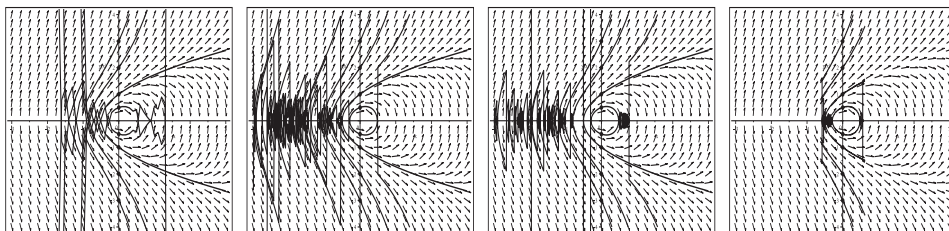


Abb. 2.127 Dateien *dber_loes_12b11..14.ps*, Richtungsfeld zur gDGl (12),
 Näherungslösungen zu den AB pts,
 $(x, y) \in [-3, 3] \times [-4, 4]$, `stepsize=0.1,0.01,0.001,0.0001`,
 Kommando `DEplot, method=classical[rk4]`

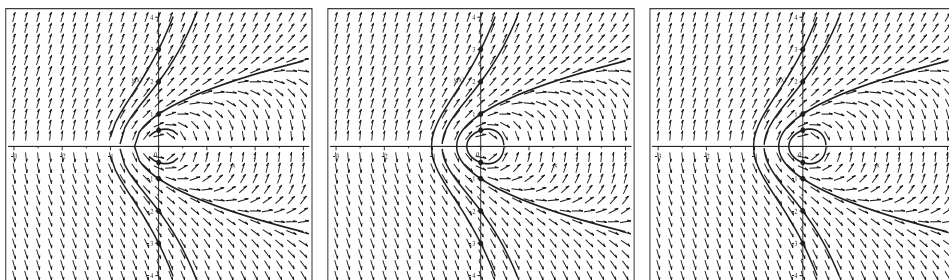


Abb. 2.128 Dateien *dber_loes_12b21..23.ps*, Richtungsfeld zur gDGl (12),
 Näherungslösungen zu den AB pts,
 $(x, y) \in [-3, 3] \times [-4, 4]$, `stepsize=0.1,0.01,0.001`,
 Kommando `DEplot, method=rkf45 bzw. gear`

Das Kommando `dsolve(dg1,y(x0)=y0,y(x))` bei beliebigem y_0 liefert keine Lösung, weil die Integrationskonstante nicht eindeutig bestimmbar ist.

Erst mittels `dsolve(dg1,y(x0)=y0,y(x)) assuming y0>0` erhält man die Formel

$$y(x) = \sqrt{1 + 2x + (y_0^2 - 1 - 2x_0) e^{2(x-x_0)}},$$

analog mit `assuming y0<0` den negativen Lösungsweig.

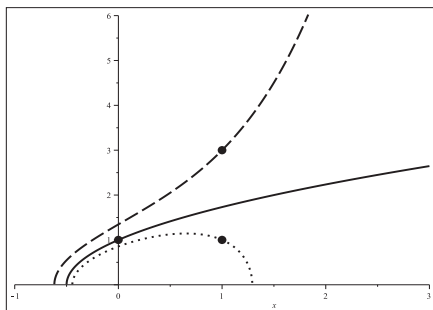
Ist zusätzlich $x_0 = 0$, folgen die Lösungen $\pm \sqrt{1 + 2x + (y_0^2 - 1) e^{2x}}$. Das Kommando `dsolve(dg1,y(x0)=0,y(x))` liefert beide Zweige $\pm e^{-x_0} \sqrt{e^{2x_0}(1 + 2x) - e^{2x}(1 + 2x_0)}$, die auch bei $x_0 \neq 0$ existieren. Wenn jedoch die AB $y(0) = 0$ ist, folgt zwar die Lösung $\pm \sqrt{1 + 2x - e^{2x}}$, aber wegen $1 + 2x - e^{2x} \leq 0$ reduziert sich der Definitionsbereich im Reellen nur auf den Punkt Null.

Man erkennt in der Lösung den Term $y_0^2 - 1 - 2x_0$ für die Fallunterscheidung.

Die AB liegt genau dann auf der Grenzkurve, falls $y_0^2 - 1 - 2x_0 = 0$.

Wenn $y_0^2 - 1 - 2x_0 < 0$, dann befindet man sich "innerhalb", sonst "außerhalb".

Weiterhin sind die exakten Lösungen an den endlichen Bereichsgrenzen berechenbar.

**Abb. 2.129**

Datei *dber_Loes_12c1.ps*,

Lösungen zur gDGI (12) zu den AB

$y(0) = 1$ (Grenzkurve $\sqrt{1+2x}$), $y(1) = 1, 3$,

Kommandos `dsolve`, `plot`, `pointplot`

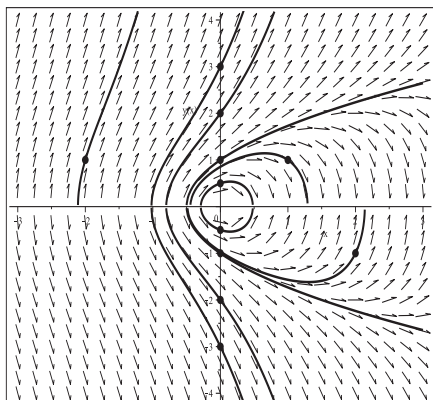
Was die analytischen Lösungen der AWP mittels `dsolve(dgl, y(x0)=y0, y(x))` betrifft, so kann man diese in ihren Definitionsbereichen zeigen.

```
AB [0,+-3]      DB [-1.020056291117846, infty)
AB [0,+-2]      DB [-0.8017728697679180, infty)
AB [0,+-1]      DB [-0.5, infty)
AB [0,+-1/2]    DB [-0.2900656995129885, 0.4806393815573885]
AB [1, 1]       DB [-0.4443516780967109, 1.291536938018345]
AB [2, -1]      DB [-0.4861454804793492, 2.138539107538423]
AB [-2, 1]      DB [-2.108822361000275, infty)
```

```
> pts3:=[[0, -3], [0, -2], [0, -1], [0, -1/2], [0, 1/2], [0, 1], [0, 2], [0, 3],
        [1, 1], [2, -1], [-2, 1]];
```

```
> dsolve({dgl=0, y(0)=-3}, y(x)); f1:=rhs(%);
solve(f1=0, x); as1:=fsolve(f1=0, x); # as1=-1.020056291117846
# analog f2, f3, ..., f12
```

```
> pl1:=dfieldplot(dgl=0, y, x=-3..3, y=-4..4, color=black, dirgrid=[25, 25]):
pl2:=plot([f1, f2, f3, f4, f6, f7, f8, f9, f10, f11, f12], x=-3..3, y=-4..4,
          color=black, thickness=3):
pl9:=pointplot(pts3, color=black, symbol=solidcircle, symbolsize=16):
display(pl1, pl2, pl9);
```

**Abb. 2.130**

Datei *dber_Loes_12d1.ps*,

Richtungsfeld zur gDGI (12),

Lösungen zu den AB `pts3`,

Kommandos `dsolve`,

`dfieldplot`, `plot`, `pointplot`

$$(13) \quad y' = -y/x + 1$$

$$y(x) = x/2 + c/x$$

Das Richtungsfeld der gDGI ist für $x \neq 0$ definiert. Der Koordinatenursprung stellt einen speziellen Punkt dar. Die rechte Seite der gDGI sowie ihre allgemeine implizite Lösung verweisen auf die Halbebenen $x < 0$ und $x > 0$ mit den dortigen Lösungen.

Man erkennt die Grenzkurve $y(x) = \frac{x}{2}$, die auch im Koordinatenursprung definiert ist. Auf der Geraden $y = x$ sind die Anstiege Null.

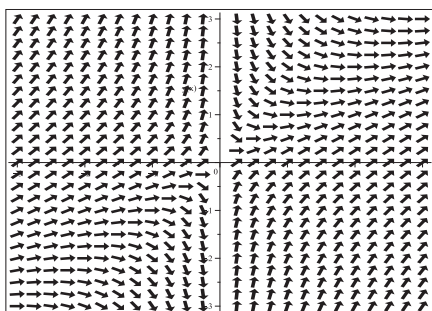


Abb. 2.131

Datei *dber_Loes_13a.ps*,
Richtungsfeld zur gDGI (13),
Kommando `dfieldplot`

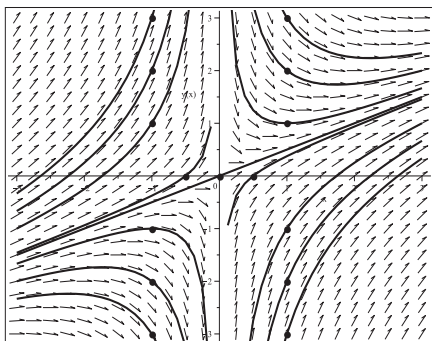
Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$ mit $x_0 \neq 0$ und wir berechnen sie mit dem Kommando `DEplot`. Aber auch mit der AB $y(0) = y_0$ gibt es bei den verschiedenen Methoden Ergebnisse. Die Lösung für die AB $y(0) = 0$ ist erwartungsgemäß $y(x) = \frac{x}{2}$. Für $y(0) = y_0 \neq 0$ gibt es keine Lösung (rechte Grafik in der Abb. 2.133). Maple liefert dazu die folgende Meldung, die auf eine behebbare Singularität verweist (AB ist beschränkt auf $y(x) = 0$), aber nichts Genaueres zur veränderten AB aussagt.

```
Error, (in dsolve/numeric/checksing) ode system has a removable
singularity at x=0. Initial data is restricted to {y(x) = 0}
```

Geraten die numerischen Verfahren in die Bereiche mit sehr steilen Linienelementen, wird es problematisch für diese. Die Warnungen in `DEplot` weisen dann auf die vorhandene Grenze $x = 0$ des Bereichs hin. Über diese wird nicht hinweg integriert.

```
> # AB
pts:=[[1, -3], [1, -2], [1, -1], [1, 1], [1, 2], [1, 3], [-1, -3], [-1, -2], [-1, -1],
      [-1, 1], [-1, 2], [-1, 3], [0, 0], [0.5, 0], [-0.5, 0]];
pts5:=[[1, -3], [1, -2], [1, -1], [1, 1], [1, 2], [1, 3], [-1, -3], [-1, -2], [-1, -1],
      [-1, 1], [-1, 2], [-1, 3], [0, 0], [0.5, 0], [-0.5, 0], [0, 1], [0, -1], [1, 0.5]];

> p11:=DEplot(dgl, y(x), x=-3..3, y=-3..3, pts, dirgrid=[25, 25],
             method=rkf45, # method=classical[rk4], method=gear,
             color=black, linecolor=black):
p12:=pointplot(pts, color=black, symbol=solidcircle, symbolsize=16):
display(p11, p12);
```

**Abb. 2.132**

Datei *dber_Loes_13b1.ps*,
 Richtungsfeld zur gDGI (13),
 Näherungslösungen zu den AB pts,
 $(x, y) \in [-3, 3] \times [-3, 3]$,
 Kommando `DEplot, method=rkf45`

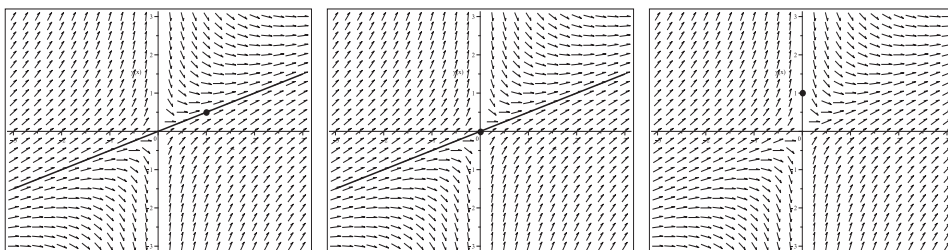


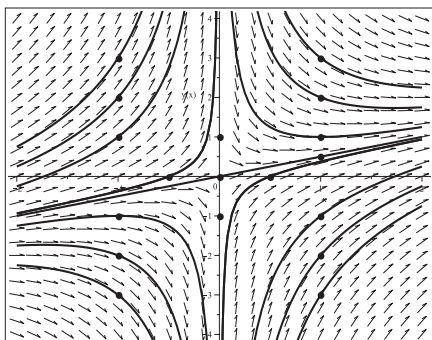
Abb. 2.133 Dateien *dber_Loes_13b2..4.ps*, Richtungsfeld zur gDGI (13),
 Näherungslösung zu $y(x) = \frac{x}{2}$ mit den AB $y(1) = \frac{1}{2}$, $y(0) = 0$,
 keine Lösung zu $y(0) = 1$ (v.l.n.r), $(x, y) \in [-3, 3] \times [-3, 3]$,
 Kommando `DEplot, method=rkf45`

Das Kommando `dsolve(dgl, y(x0)=y0, y(x))` liefert die Lösung

$$y(x) = \frac{x}{2} + \frac{x_0(y_0 - x_0/2)}{x}.$$

Je nach $x_0 < 0$ oder $x_0 > 0$ liegen die relevanten Lösungsbranche in der linken bzw. rechten Halbebene des Koordinatensystems.

Mit der AB $y(0) = y_0$ bei beliebigem y_0 ergibt sich immer die Grenzlösung $y(x) = \frac{x}{2}$, die natürlich auch im Punkt $(0, 0)$ funktioniert, aber eben nicht die AB bei $y_0 \neq 0$ erfüllt.

**Abb. 2.134**

Datei *dber_Loes_13c1.ps*,
 Richtungsfeld zur gDGI (13),
 Lösungen zu den AB pts3,
 Kommandos `dsolve,`
`dfieldplot, plot, pointplot`

$$(14) \quad y' = xy + x^3 y^2$$

$$y(x) = 1/(2 - x^2 + c e^{-x^2/2})$$

Die allgemeine Lösung mittels `dsolve` umfasst nicht die erkennbare stationäre Nulllösung. Das Richtungsfeld der gDGL ist überall definiert.

Für kleine Werte $|x|, |y|$ erhält man moderate Linienelemente, für $|x| > 0, |y| \gg 0$ sehr steile Linienelemente mit Steigungen aus $(-\infty, \infty)$. Das ist ein Hinweis auf Einschränkungen bei den Definitionsbereichen der Form $(-d, d)$, $(-\infty, -d)$ bzw. (d, ∞) und auf entsprechende Asymptoten.

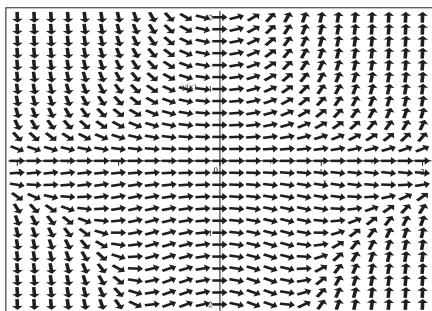


Abb. 2.135

Datei `dbcr_loes_14a.ps`,
Richtungsfeld zur gDGL (14),
Kommando `dfieldplot`

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$ und sie werden hinreichend genau mit dem Kommando `DEplot, method=rkf45, stepsize=0.01` berechnet.

Damit nähert man sich auch gut an die evtl. vorhandenen Polstellen der Lösungen. Aber es wird nicht über diese Stellen hinweg integriert.

Für die AB $y(x_0) = 0$ erhält man die Nulllösung.

Eine weitere spezielle AB ist $y_0 = -1/(2e^{-x_0^2/2} - (2 - x_0)) < 0$, wobei $x_0 \neq 0$ ist.

Damit hat die Lösung die y -Achse als Asymptote. Bei $x_0 = 1$ ist $y_0 = -1/(2/\sqrt{e}-1) < 0$, die als AB auch in der folgenden Abbildung berücksichtigt wurde. Dieser Fall entspricht dem Parameter $c = -2$ in der allgemeinen Lösung, wo der Nenner bei $x = 0$ verschwindet.

```
> # AB
pts:=[[0,-10],[0,-5],[0,-3],[0,-2],[0,-1],[0,0],[0,1],[0,2],[0,4],
      [1,-4],[1,-1/(2/sqrt(exp(1))-1)],[1,-8]];
pts1:=[[0,2]]; pts2:=[[1,-1/(2/sqrt(exp(1))-1)]];

> p1:=DEplot(dgl,y(x),x=-2..2,y=-20..10,pts,dirgrid=[25,25],
             # method=classical[rk4], # method=gear,
             method=rkf45,stepsize=0.01,
             color=black,linecolor=black):
p2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p1,p2);

> p1:=DEplot(dgl,y(x),x=0..2,y=-20..5,pts,dirgrid=[25,25],
             method=rkf45,stepsize=0.01,color=black,linecolor=black):
```

```
display(pl1,pl2);
```

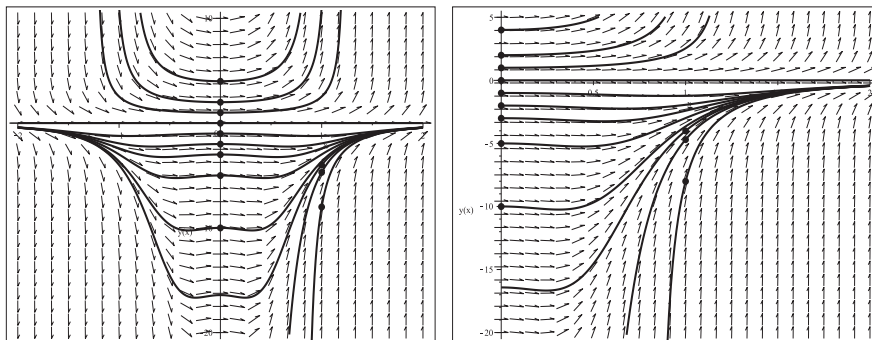


Abb. 2.136 Dateien *dber_Joes_14b1..2.ps*, Richtungsfeld zur gDGl (14),
Näherungslösungen zu den AB pts,
 $(x, y) \in [-2, 2] \times [-20, 10]$ bzw. $[0, 2] \times [-20, 5]$,
Kommando `DEplot, method=rkf45, stepsize=0.01`

Die analytische Lösung mittels `dsolve(dgl, y(x0)=y0, y(x))` liefert die Formel

$$y(x) = \frac{y_0}{y_0(2-x^2) + (1-y_0(2-x_0^2))e^{(x_0^2-x^2)/2}},$$

speziell für die AB $y(0) = y_0$ die Lösung

$$y(x) = \frac{y_0}{y_0(2-x^2) + (1-2y_0)e^{-x^2/2}}.$$

Zur AB $y(x_0) = 0$ wird die stationäre Lösung $y(x) = 0$ erzeugt.

Fallunterscheidung bezüglich der Definitionsbereiche

(a) AB $y(0) = y_0 \in (-\infty, 0]$

Es liegt die moderate Lösung $y(x) = -y_0 / (-y_0(2-x^2) - (1-2y_0)e^{-x^2/2}) < 0$ vor,
weil der Nennerterm darin immer negativ für $y_0 < 0$ ist.

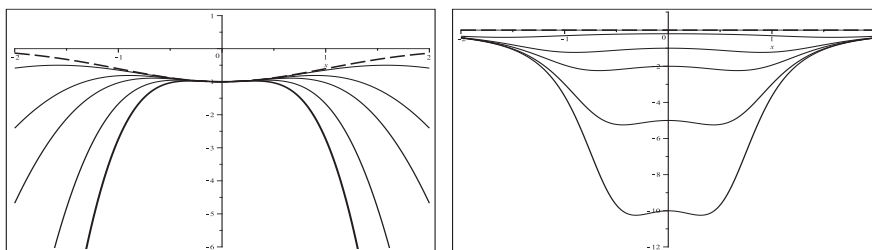


Abb. 2.137 Dateien *dber_Joes_14c1..2.ps*, gDGl (14),
Nennerterm (links) sowie Lösungen zu den AB $y(0) = -\frac{1}{5}, -1, -2, -5, -10$
und Grenzfall $y(0) = 0$ (gestrichelt), $x \in [-2, 2]$,
Kommandos `dsolve, plot`

(b) AB $y(x_0) = y_0 > 0$, speziell bei $x_0 = 0$

Man erhält die Lösung $y(x) > 0$ im symmetrischen Intervall $(-d, d)$, wobei d die positive Nullstelle von $y_0(2 - x^2) + (1 - 2y_0)e^{-x^2/2} = 0$ ist. Die Asymptoten liegen bei $\pm d$.

Es werden auch die Grenzkurve $y(x) = 0$ und die Funktionszweige in (d, ∞) und $(-\infty, -d)$ gezeigt. Aus den AB erhält man folgende Definitionsbereiche:

$$y(0) = \frac{1}{5}, \quad d = 1.660174379673303, \quad (-d, d),$$

$$y(0) = 1, \quad d = 1.239386176309439, \quad (-d, d),$$

$$y(0) = 2, \quad d = 1.077154955450679, \quad (-d, d),$$

$$y(0) = 4, \quad d = 0.929371009912406, \quad (-d, d),$$

$$y(0) = 5, \quad d = 0.885052219099605, \quad (-d, d),$$

$$y(0) = 10, \quad d = 0.757691598357659, \quad (-d, d).$$

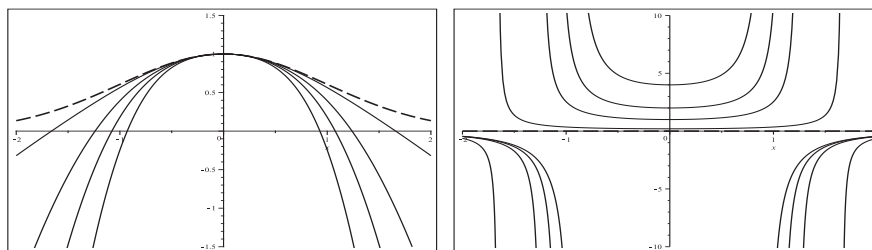


Abb. 2.138 Dateien *dber_Loes_14c3.4.ps*, gDGI (14),

Nennerterm (links) sowie Lösungen mit den AB $y(0) = \frac{1}{5}, 1, 2, 4$ mit äußeren Zweigen sowie Grenzfall $y(0) = 0$ (gestrichelt), $x \in [-2, 2]$,
Kommandos `dsolve`, `plot`

(c) AB $y(x_0) = y_0 < 0$ bei $x_0 \neq 0$, speziell $x_0 = 1$

Der Nenner $-y_0(2 - x^2) - (1 - y_0)e^{(1-x^2)/2}$ der Lösung ist auf Null zu testen.

Es zeigt sich, dass es für AB $y(x_0) = y_0 \in (y_{0g}, 0)$, $y_{0g} = -1/(2\sqrt{e} - 1) = -4.693\dots$, keine reellen Lösungen der Nullstellenaufgabe gibt, der Nenner nicht verschwindet und damit die Lösungen überall definiert sind.

Die AB $y(x_0) = y_{0g}$ ist der Grenzfall mit dem Definitionsbereich $(0, \infty)$.

AWP mit $y(x_0) < y_{0g}$ haben Lösungen im Definitionsbereich (d, ∞) , $d > 0$.

Aus den AB erhält man folgende Lösungsintervalle:

$$y(1) = y_{0g}, \quad d = 0, \quad (d, \infty),$$

$$y(1) = -5, \quad d = 0.5291819981691424, \quad (d, \infty),$$

$$y(1) = -8, \quad d = 0.8242646534157171, \quad (d, \infty),$$

$$y(1) = -10, \quad d = 0.8714073884153985, \quad (d, \infty),$$

$$y(1) = -20, \quad d = 0.9441500862707414, \quad (d, \infty).$$

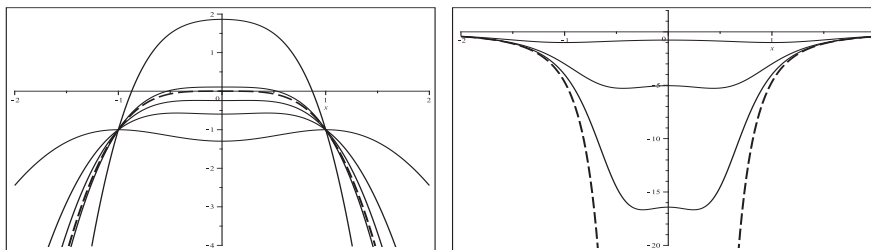


Abb. 2.139 Dateien *dber_Joes_14c5..6.ps*, gDGl (14),
 Nennerterm (links) zu den AB $y(1) = -1, -3, -4, y_{0g}, -5, -10$
 sowie Lösungen mit den AB $y(1) = -1, -3, -4$
 und Grenzfall $y(1) = y_{0g}$ (gestrichelt), $x \in [-2, 2]$,
 Kommandos `dsolve`, `plot`

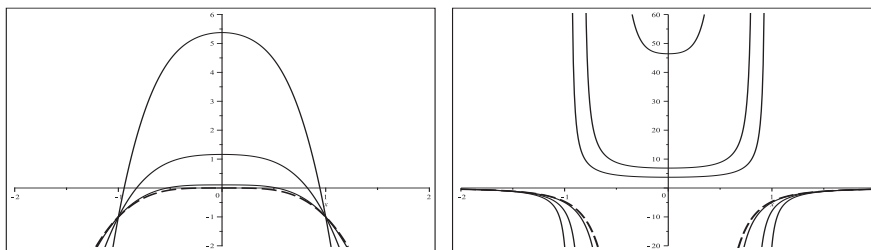


Abb. 2.140 Dateien *dber_Joes_14c7..8.ps*, gDGl (14),
 Nennerterm (links) sowie Lösungen zu den AB $y(1) = y_{0g}, -5, -8, -20$
 mit Zweigen sowie Grenzfall $y(1) = y_{0g}$ (gestrichelt) in $x \in [-2, 2]$,
 Kommandos `dsolve`, `plot`

Was die analytischen Lösungen der AWP mittels `dsolve(dgl, y(x0)=y0, y(x))` betrifft, so kann man diese mit allen Zweigen zeigen oder auch nur den relevanten Lösungszweig, evtl. noch dazu die Asymptoten. Hier werden alle Zweige geplottet.

Mögliche Polstellen zu den AB sind im Vektor `as` abgelegt.

```
> dsolve({dgl=0, y(0)=-10}, y(x)); f1:=rhs(%);
ash:=solve(-20+10*x^2+21*exp(-x^2/2), x);
# imaginaer, d.h. keine Einschraenkung
pl1:=plot(f1, x=-3..3, color=black, thickness=3);
display(pl1);
# analog f2, f3, ..., f11, f12

> pl1:=dfieldplot(dgl=0, y, x=-2..2, y=-30..20, color=black, dirgrid=[25,25]):
pl2:=plot([f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12], x=-2..2, y=-30..20,
color=black, thickness=3, discont=true):
pl8:=pointplot(pts, color=black, symbol=solidcircle, symbolsize=16):
pl9:=plot({[[as[1], -30], [as[1], 20]], ...}, color=black):
display(pl1, pl2, pl8, pl9);
```

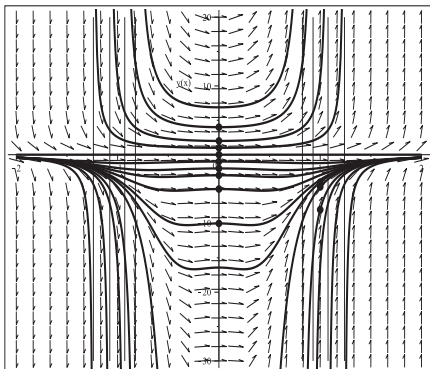


Abb. 2.141

Datei *dber_Loes_14d1.ps*,
 Richtungsfeld zur gDGL (14),
 Lösungen zu den AB pts, mit zusätzlichen
 Funktionszweigen und Asymptoten,
 Kommandos `dsolve`,
`dfieldplot`, `plot`, `pointplot`

$$(15) \quad y' = -y \cos(x) + e^{-\sin(x)}$$

$$y(x) = (x + c) e^{-\sin(x)}$$

Das Richtungsfeld der gDGL ist überall definiert.

Für große Werte $|y|$ und $\cos(x) \approx 1$ erhält man steile Linienelemente und insgesamt sich aufschwingende Lösungskurven.

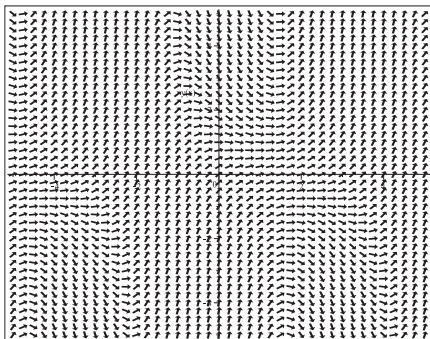


Abb. 2.142

Datei *dber_Loes_15a.ps*,
 Richtungsfeld zur gDGL (15),
 Kommando `dfieldplot`

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$. Man berechnet sie wegen der steilen Abschnitte in den Lösungskurven mittels `DEplot` mit guten Näherungsmethoden und/oder hinreichend kleinen Schrittweiten, z.B. `method=rkf45,stepsize=0.01`.

```
> # AB
pts:=[[0,-2],[0,-1],[0,0],[0,1],[0,2],[0,5],[0,10]];
pts1:=[[0,-2],[0,0],[0,2]];

> p11:=DEplot(dgl,y(x),x=-15..15,y=-50..60,pts,
              # x=-10..15,pts1,
              color=black,linecolor=black,dirgrid=[31,41],
              method=rkf45,stepsize=0.01):
p12:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
      # pts1,
display(p11,p12);
```

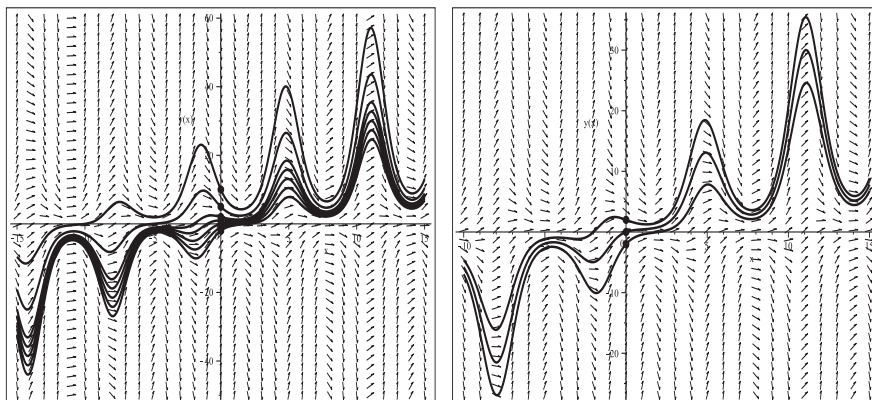


Abb. 2.143 Dateien *dber_loes_15b1..2.ps*, Richtungsfeld zur gDGl (15),
 Näherungslösungen zu den AB pts bzw. pts1,
 $(x, y) \in [-15, 15] \times [-50, 60]$ bzw. $x \in [-10, 15]$,
 Kommando `DEplot, method=rkf45, stepsize=0.01`,

Die analytische Lösung mittels `dsolve(dgl, y(x0)=y0, y(x))` liefert die Formel

$$y(x) = (x - x_0 + y_0 e^{\sin(x_0)}) e^{-\sin(x)}.$$

Damit gibt es zwei die Lösungskurven begrenzende Geraden

$$f_o(x) = e(x - x_0 + y_0 e^{\sin(x_0)}), \quad f_u(x) = e^{-1}(x - x_0 + y_0 e^{\sin(x_0)}),$$

die sich im Punkt $(x_s, y_s) = (x_0 - y_0 e^{\sin(x_0)}, 0)$ schneiden.

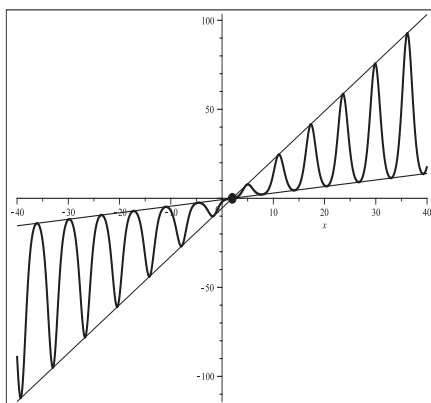


Abb. 2.144

Datei *dber_loes_15c1.ps*, gDGl (15),
 Lösung zur AB $y(0) = -2$
 in $[-40, 40] \times [-110, 100]$
 mit den begrenzenden Geraden
 $f_o(x) = e(x - 2)$, $f_u(x) = e^{-1}(x - 2)$,
 Kommandos `dsolve, plot, pointplot`

Die grafische Darstellung der analytischen Lösungen der entsprechenden AWP ist sehr ähnlich zu der Abb. 2.143.

$$(16) \quad y' = 2(y+1) \tan(x)$$

$$y(x) = \frac{2c - \cos(2x)}{1 + \cos(2x)}$$

Die allgemeine parameterabhängige Lösung mittels `dsolve` umfasst die erkennbare stationäre Lösung $y(x) = -1$, das entspricht dem Parameter $c = -\frac{1}{2}$.

Das Richtungsfeld der gDGL ist an den Stellen $x = (2k+1)\frac{\pi}{2}$, k ganzzahlig, nicht definiert. Das sind Polstellen der rechten Seite der gDGL und auch für die Lösung. Die Periode π der Tangensfunktion überträgt sich auf die Lösung.

Bis auf die konstante Lösung wird jede andere je nach AB in einem offenen Intervall $((2k-1)\frac{\pi}{2}, (2k+1)\frac{\pi}{2})$ definiert sein und mit der Periode π kann man nach links und rechts weitere Zweige zeichnen.

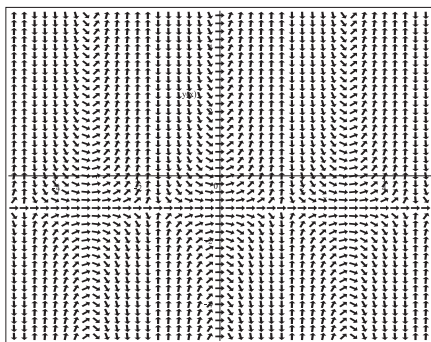


Abb. 2.145

Datei `dber_loes_16a.ps`,
Richtungsfeld zur gDGL (16),
Kommando `dfieldplot`

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$ mit $x_0 \neq (2k+1)\frac{\pi}{2}$ sowie $y(x_0) = -1$, x_0 beliebig, und wir berechnen sie hinreichend genau mit dem Kommando `DEplot, method=rkf45, stepsize=0.01`.

Damit nähert man sich im ersten Fall auch gut an die vorhandenen Polstellen der Lösungen, aber es wird nicht über diese Stellen hinweg integriert.

Für die AB $y(x_0) = -1$ erhält man die Lösung $y(x) = -1$.

```
> # AB
pts:=[[0,-3],[0,-2],[0,-1],[0,0],[0,1],[0,2],[2,0],[2,-2],
      [-5/2,0],[-5/2,-2]];
pts1:=[[0,-1],[0,0],[0,1],[5/4,-1/2],[1,-3/2]];

> p1:=DEplot(dgl,y(x),x=-5..5,y=-5..5,pts,
             # x=-2..2,y=-3..2,pts1,
             color=black,linecolor=black,dirgrid=[41,41],
             method=rkf45,stepsize=0.01):
p2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
      # pts1,
display(p1,p2);
```

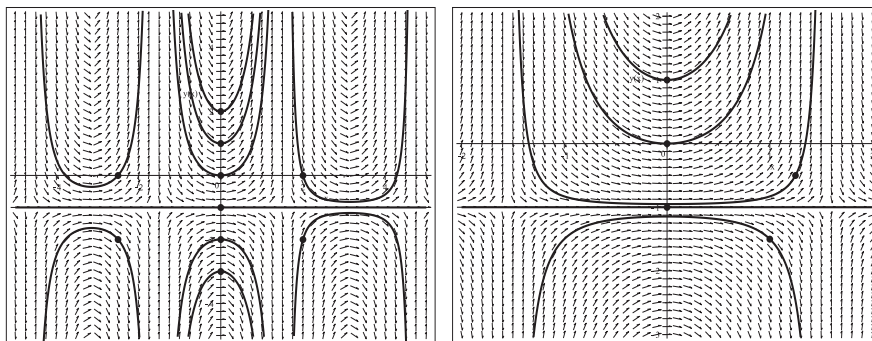


Abb. 2.146 Dateien *dber_Loes_16b1..2.ps*, Richtungsfeld zur gDGl (16),
 Näherungslösungen zu den AB pts bzw. pts1,
 $(x, y) \in [-5, 5] \times [-5, 5]$ bzw. $[-2, 2] \times [-3, 2]$,
 Kommando `DEplot, method=rkf45, stepsize=0.01`

Die analytische Lösung mittels `dsolve(dgl, y(x0)=y0, y(x))` liefert die Formel

$$y(x) = -1 + (1 + y_0) \frac{1 + \cos(2x_0)}{1 + \cos(2x)} = -1 + (1 + y_0) \left(\frac{\cos(x_0)}{\cos(x)} \right)^2,$$

speziell für die AB $y(0) = y_0$ die Lösung

$$y(x) = \frac{2y_0 + 1 - \cos(2x)}{1 + \cos(2x)}.$$

Zur AB $y(x_0) = -1$ wird die stationäre Lösung $y(x) = -1$ erzeugt, insbesondere mit dem Kommando `dsolve({dgl=0, y((2*k+1)*Pi/2)=-1}, y(x)) assuming k::integer`.

Der Befehl `dsolve({dgl=0, y(Pi/2)=y0}, y(x))` liefert bei $y_0 \neq -1$ keine Lösung.

Umso verwunderlicher ist es, dass bei $x_0 = (2k+1)\frac{\pi}{2}$, k ganzzahlig, und $y_0 = 0$ der Befehl `dsolve({dgl=0, y((2*k+1)*Pi/2)=0}, y(x)) assuming k::integer` die Lösung -1 erzeugt, während für andere y_0 kein Ergebnis geliefert wird.

Die grafische Darstellung der analytischen Lösungen der entsprechenden AWP enthält auch die periodischen Fortsetzungen (Nachbarzweige) der Lösungen.

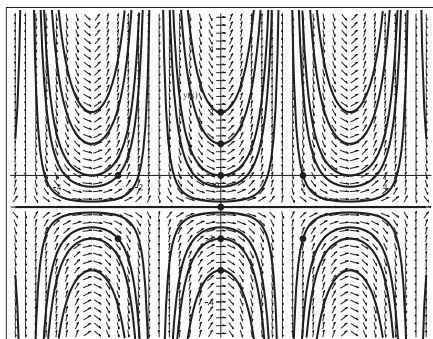


Abb. 2.147

Datei *dber_Loes_16d1.ps*,
 Richtungsfeld zur gDGl (16),
 Lösungen zu den AB pts,
 dazu die periodischen Fortsetzungen,
 Kommandos `dsolve,`
`dfieldplot, plot, pointplot`

$$(17) \quad y' = y^4 \sin(x)$$

$$y(x) = \frac{1}{\sqrt[3]{c + 3 \cos(x)}}$$

Die allgemeine parameterabhängige Lösung mittels `dsolve` umfasst die erkennbare stationäre Lösung $y(x) = 0$, das entspricht dem Parameter $c = \infty$.

Mittels `dsolve` entstehen noch die beiden konjugiert komplexen Lösungen

$$-\frac{1}{2 \sqrt[3]{c + 3 \cos(x)}} \pm \frac{i \sqrt{3}}{2 \sqrt[3]{c + 3 \cos(x)}}.$$

Besonders ist die dritte Wurzel im Nenner der Lösung zu beachten. Darauf wird später noch eingegangen.

Das Richtungsfeld der gDGL ist zwar überall definiert, aber die vierte Potenz von y auf der rechten Seite lässt moderate Lösungen für Werte $|y| < 1$ vermuten. Weiterhin wird es auch sehr steile Verläufe, Lösungszweige und entsprechende Polstellen geben.

Die Periode 2π der Sinusfunktion überträgt sich auf die Lösung.

Die konstanten sowie moderaten Lösungen sind für alle reellen x definiert, jede andere je nach AB in einem offenen Intervall $(k\pi - \delta, k\pi + \delta)$. Diese kann man sich mit der Periode 2π nach links und rechts fortgesetzt denken. An den dazwischen liegenden Intervallen kann es aber auch Lösungszweige geben.

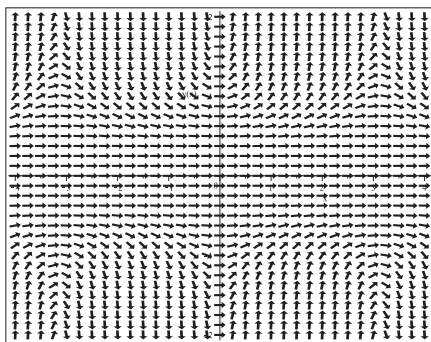


Abb. 2.148

Datei `dber_Loes_17a.ps`,
Richtungsfeld zur gDGL (17),
Kommando `dfieldplot`

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$. Für die AB $y(x_0) = 0$ erhält man die Lösung $y(x) = 0$.

Die zu erwartenden steilen Lösungsverläufe lassen sich hinreichend genau mit dem Kommando `DEplot, method=rkf45, stepsize=0.01` berechnen. Damit nähert man sich auch gut an die vorhandenen Polstellen der Lösungen, jedoch wird nicht über diese Stellen hinweg integriert.

```
> # AB
pts:=[[0, -3], [0, -2], [0, -1], [0, -1/2], [0, 0], [0, 1/3], [0, 1/2], [0, 0.549],
      [0, 2/3], [0, 1], [0, 2], [4.5, 1], [-4.5, 1], [Pi, -1], [-Pi, -1]];
```

```

pts1:=[[0,-1],[0,-1/2],[0,0],[0,1/3],[0,1/2],[0,1/6^(1/3)],[0,1],
      [2*Pi,1/6^(1/3)],[-2*Pi,1/6^(1/3)],[Pi,-1/6^(1/3)],
      [-Pi,-1/6^(1/3)],[2*Pi,1],[-2*Pi,1],[Pi,-1],[-Pi,-1]];

> p11:=DEplot(dg1,y(x),x=-7..7,y=-3.5..3.5,pts,
             # x=-7..7,y=-1.5..1.5,pts1,
             color=black,linecolor=black,dirgrid=[29,29], #dirgrid=[29,31]
             method=rkf45,stepsize=0.01):
p12:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
      # pts1,
display(p11,p12);

```

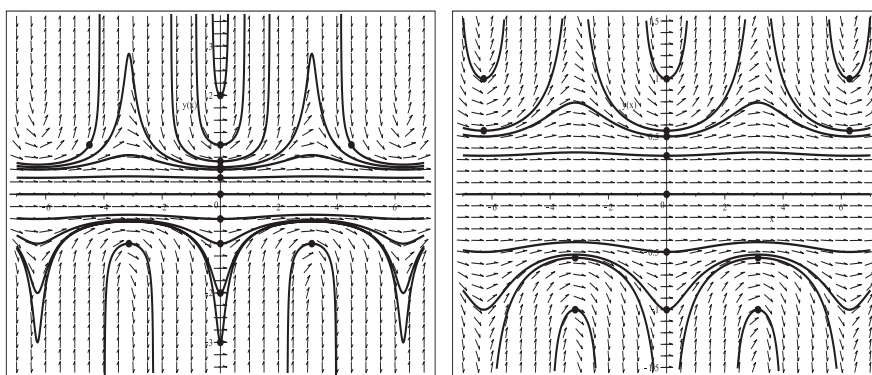


Abb. 2.149 Dateien *dber_Joes_17b1..2.ps*, Richtungsfeld zur gDGL (17),
 Näherungslösungen zu den AB pts bzw. pts1,
 $(x, y) \in [-7, 7] \times [-3.5, 3.5]$ bzw. $[-7, 7] \times [-1.5, 1.5]$,
 Kommando DEplot, method=rkf45,stepsize=0.01

Die analytische Lösung mittels $\text{dsolve}(\text{dg1}, y(x_0)=y_0, y(x))$ liefert die Formel

$$y(x) = \frac{y_0}{\sqrt[3]{1 + 3y_0^3 (\cos(x) - \cos(x_0))}},$$

speziell für die AB $y(2k\pi) = y_0$, $y((2k+1)\pi) = y_0$ bzw. $y(x_0) = 0$ die Lösungen

$$y(x) = \frac{y_0}{\sqrt[3]{1 + 3y_0^3 (\cos(x) - 1)}}, \quad y(x) = \frac{y_0}{\sqrt[3]{1 + 3y_0^3 (\cos(x) + 1)}} \quad \text{bzw.} \quad y(x) = 0.$$

In der rechten Grafik in der Abb. 2.149 werden u.a. die AB $y(2k\pi) = y_{0g} = 1/\sqrt[3]{6} = 0.550321\dots$, $k = -1, 0, 1$, und $y(k\pi) = -y_{0g} = -1/\sqrt[3]{6}$, $k = -1, 1$, verwendet. Erstere erzeugt eine ‘Grenzkurve nach oben’, nämlich die Näherungslösung zu

$$y(x) = \frac{y_{0g}}{\sqrt[3]{1 + 3y_{0g}^3 (\cos(x) - 1)}} = \frac{1}{\sqrt[3]{3 + 3\cos(x)}},$$

die bis auf die Polstellen bei $(2k+1)\pi$, k ganzzahlig, definiert und positiv ist.

Analog erhält man die “Grenzkurve nach unten“ mit den anderen AB, nämlich die Näherungslösung zu

$$y(x) = \frac{y_{0g}}{\sqrt[3]{1 + 3y_{0g}^3 (\cos(x) + 1)}} = -\frac{1}{\sqrt[3]{3 - 3\cos(x)}},$$

die bis auf die Polstellen bei $2k\pi$, k ganzzahlig, definiert und negativ ist.

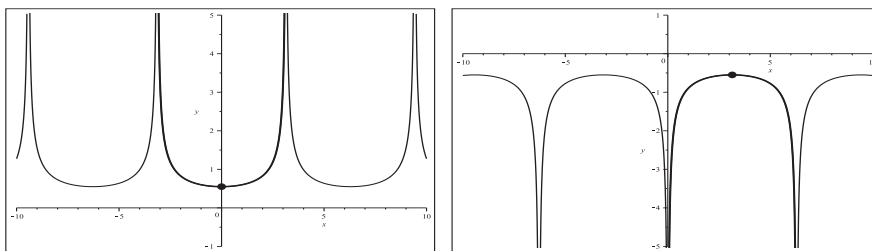


Abb. 2.150 Dateien *dber_Loes_17c1..2.ps*, gDGI (17), Grenzlösungen zu den AB $y(0) = y_{0g} = 1/\sqrt[3]{6} = 0.550321\dots$ bzw. $y(\pi) = -y_{0g}$, mit Fortsetzung der Lösung (Zweige nach links und rechts), $(x, y) \in [-10, 10] \times [-1, 5]$ bzw. $[-10, 10] \times [-5, 1]$, Kommandos `dsolve`, `plot`, `pointplot`

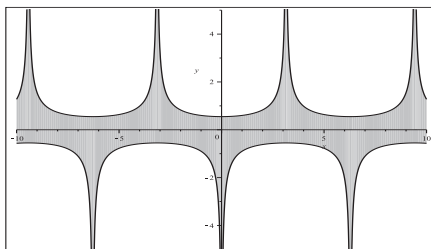


Abb. 2.151

Datei *dber_Loes_17c5.ps*, gDGI (17), moderate Lösungen sowie Näherungslösungen mit guten Verfahren zu AB $y(x_0) = y_0$ aus dem grauen Gebiet

Betrachten wir nun zwei Lösungen mit AB außerhalb des grauen Bereichs in Abb. 2.151. Zu $y(0) = 1$ bzw. $y(\pi) = -1$ erhält man

$$y(x) = \frac{1}{\sqrt[3]{3\cos(x) - 2}}, \quad x \in (-d, d), \quad d = \arccos(2/3) = 0.841068\dots,$$

bzw.

$$y(x) = \frac{1}{\sqrt[3]{-3\cos(x) - 2}}, \quad x \in (d, 2\pi - d), \quad d = \arccos(-2/3) = 2.300523\dots$$

Formal kann man die Lösung durch Zweige nach links und rechts mit der Periode 2π fortsetzen. Da sind aber noch die “Zwischenräume“. Da die dritte Wurzel aus einer negativen Zahl existiert, kann man auch für negative Radikanden die dazwischen liegenden Kurvenstücke zeichnen. Das wurde in der folgenden Abbildung auch getan.

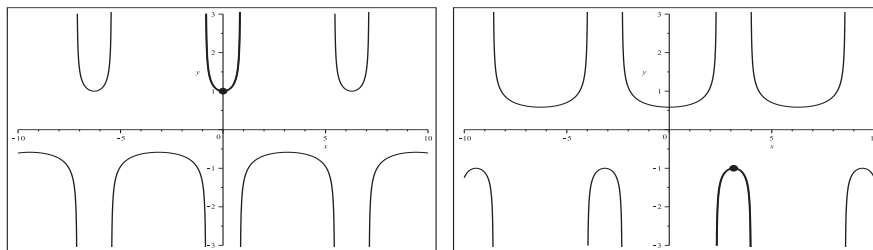


Abb. 2.152 Dateien *dber_Joes_17c3..4.ps*, gDGl (17),
 Lösungen zu den AB $y(0) = 1$ bzw. $y(\pi) = -1$,
 mit Fortsetzung der Lösung (Zweige nach links, rechts und
 dazwischen), $(x, y) \in [-10, 10] \times [-3, 3]$,
 Kommandos `dsolve`, `plot`, `pointplot`

Die Zwischenstücke muss man extra berücksichtigen. Das hängt damit zusammen, dass der Maple-Plot `plot(x^(1/3), x=-3..3)` nur den Lösungszweig für $x \geq 0$ zeigt.

Ähnlich bleibt in Maple (analog in MATLAB) der Term $(-1)^{(1/3)}$ unverändert, das Kommando `simplify((-1)^(1/3))` liefert die komplexe Wurzel $\frac{1}{2} + \frac{1}{2}i\sqrt{3}$ (nicht -1) und erst der Befehl `solve(x^3+1,x)` alle drei Wurzeln.

Zu allen AB pts kann man nun mittels `dsolve` die Lösung, dazu den Definitionsbereich sowie die zusätzlichen Zweige bestimmen.

Die analytische Lösung des entsprechenden AWP wird möglicherweise in komplexer Form erzeugt, aber wertemäßig wird sie reell sein. Ihre grafische Darstellung enthält, wenn vorhanden, auch die 2π -periodischen Fortsetzungen der Lösung, nicht aber die Stücke, wo der Radikand unter der dritten Wurzel negativ wird.

```
> dsolve({dgl=0,y(0)=-3},y(x));
f1:=rhs(%):
simplify(subs(x=0,f1));      # Kontrolle
simplify(subs(y(x)=f1,dgl));
ff:=unapply(f1,x):
evalf([ff(-2),ff(0),ff(2)]); # Rechnung im Komplexen
                               # Ergebnisse sind reelle Werte

> p1:=plot(f1,x=-7..7,y=-5..1,color=black,thickness=3):
p2:=pointplot([[0,-3]],color=black,symbol=solidcircle,symbolsize=20):
display(p1,p2);
```

$$-\frac{3}{2(81 \cos(x) - 82)^{1/3}} - \frac{3i\sqrt{3}}{2(81 \cos(x) - 82)^{1/3}}$$

-3

0

$[-0.6156495863360843 + 0. I, -3.0000000000000000 + 0. I, -0.6156495863360843 + 0. I]$

Bei der AB $y(0) = \frac{2}{3}$ ergibt sich die Lösungsformel

$$\frac{2}{3} \frac{9^{1/3} ((8 \cos(x) + 1)^2)^{1/3}}{8 \cos(x) + 1},$$

wo ausnahmsweise der Radikand unter der dritten Wurzel nicht mehr in der ursprünglichen Form steht, sondern sich ein Quadrat unter der Wurzel "eingeschlichen" hat. Damit werden im Plot alle Zweige und Zwischenstücke angezeigt.

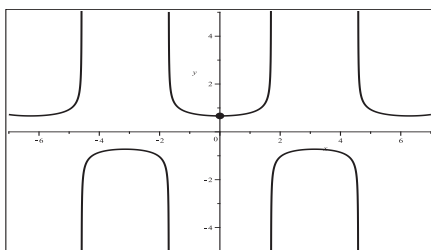


Abb. 2.153

Datei *dber_Loes_17d1.ps*, gDGI (17),

Lösung zur AB $y(0) = \frac{2}{3}$,

dazu die periodischen Fortsetzungen
und Zwischenstücke,

Kommandos `dsolve, plot, pointplot`

Aus den AB pts erhält man folgende Definitionsbereiche (Angaben mit 6 Nachkommastellen):

$y(0) = -3, -2, -1, -\frac{1}{2}, 0, \frac{1}{3}, \frac{1}{2}, 0.549,$	$(-\infty, \infty),$
$y(0) = \frac{2}{3},$	$d = \arccos(-\frac{1}{8}) = 1.696124,$
$y(0) = 1,$	$d = \arccos(\frac{2}{3}) = 0.841068,$
$y(4.5) = 1,$	$d = 2\pi - \arccos((3 \cos(4.5) - 1)/3) = 4.137038,$
$y(-4.5) = 1,$	$d = \arccos((3 \cos(4.5) - 1)/3) - 2\pi = -4.137038,$
$y(\pi) = -1,$	$d = \arccos(-\frac{2}{3}) = 2.300523,$
$y(-\pi) = -1,$	$d = \arccos(-\frac{2}{3}) - 2\pi = -3.982661,$

In der folgenden Abbildung werden die Lösungen und die 2π -periodischen Zweige gezeigt.

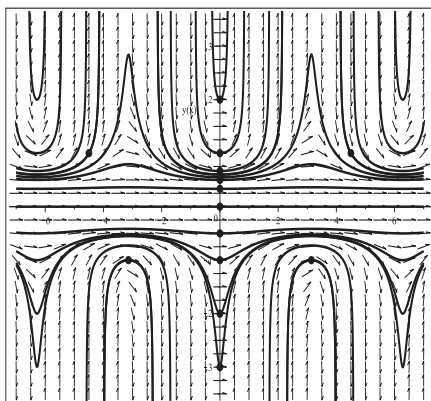


Abb. 2.154

Datei *dber_Loes_17d2.ps*,

Richtungsfeld zur gDGI (17),

Lösungen zu den AB pts,

dazu die periodischen Fortsetzungen,

Kommandos `dsolve,`

`dfieldplot, plot, pointplot`

$$(18) \quad y' = xy \cos(y)$$

Die nichtlineare gDGL ist vergleichbar mit der etwas einfacheren Gestalt (5), es existiert hier aber keine allgemeine explizite analytische Lösung.

Das Richtungsfeld ist überall definiert. An den Werten $y = y_k = (k + \frac{1}{2})\pi$, k ganzzahlig, verschwindet der Cosinus und die Linienelemente sind horizontal, letzteres auch für die Werte $x = 0$ und $y = 0$. Damit gibt es stationäre Lösungen, die abwechselnd stabile oder instabile Gleichgewichtslagen darstellen.

Die rechte Seite lässt moderate Lösungen für kleine Werte $|x|$ und $|y| < 1$ vermuten, aber es wird auch sehr steile Verläufe für $x \gg 1$ und/oder $y \gg 1$ geben. Die Cosinusfunktion verhindert, dass Polstellen zu den Lösungen entstehen können.

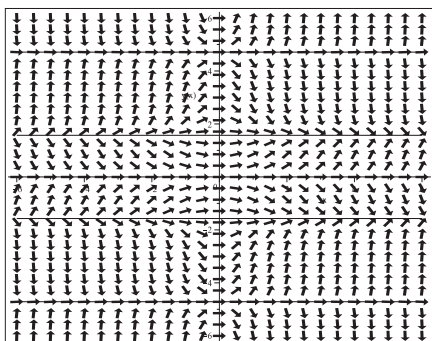


Abb. 2.155

Datei *dber_loes_18a.ps*,

Richtungsfeld zur gDGL (18),

Kommando `dfieldplot`

Mit dem Kommando `DEplot` berechnet man die numerischen Lösungen, die es zu allen AB $y(x_0) = y_0$ gibt. In kleinen Lösungsintervallen funktioniert alles recht gut. Aber bei großen Bereichen, wo für $|x|, |y| \gg 1$ die Linienelemente immer steiler werden können, wird es problematisch für alle numerischen Verfahren.

Verfahren hoher Genauigkeit erzeugen zumeist akzeptable Näherungen. Aber manchmal beginnen auch diese und insbesondere solche mit niedriger Genauigkeit schon bald wegzudriften und können an den stabilen Gleichgewichtslagen die gewünschten Kurven (asymptotische Verläufe) nicht halten. Hier wird auch auf die Abb. 2.101 zur gDGL (5) verwiesen.

Wir zeigen Verläufe von Näherungsfunktionen aus numerischen Verfahren und mit verschiedenen AB.

```
> # AB
pts:=[[0,-9],[0,-6],[0,-5],[0,-4],[0,-2],[0,-1],[0,0],
      [0,1],[0,2],[0,4],[0,5],[0,6],[0,9],
      [0,Pi/2],[0,3*Pi/2],[0,5*Pi/2],[0,-Pi/2],[0,-3*Pi/2],[0,-5*Pi/2]];
pts1:=[[0,-9],[0,-6],[0,-5],[0,-4],[0,-2],[0,-1],[0,0],
       [0,1],[0,2],[0,4],[0,5],[0,6],[0,9]];
pts2:=[[0,0],[0,Pi/2],[0,3*Pi/2],[0,5*Pi/2],[0,-Pi/2],[0,-3*Pi/2],
       [0,-5*Pi/2]];
```



```

pts3:=[[0,0.02+evalf(15*Pi/2)],[0,2+evalf(17*Pi/2)],
       [0,0.02+evalf(19*Pi/2)],[0,2+evalf(21*Pi/2)],
       [0,0.02+evalf(23*Pi/2)],[0,2+evalf(25*Pi/2)],
       [0,0.02+evalf(27*Pi/2)],[0,2+evalf(29*Pi/2)]];
pts4:=[[0,-50],[0,-30],[0,-10],[0,0],[0,10],[0,30],[0,50]];

> p11:=DEplot({dg1},[y(x)],x=-10..10,y=-10..10,pts,
              color=black,linecolor=black,dirgrid=[31,31],
              method=rkf45,stepsize=0.01):
p12:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
p13:=plot([seq((2*k+1)*Pi/2,k=-3..2)],x=-10..10,color=black):
display(p11,p12,p13);

```

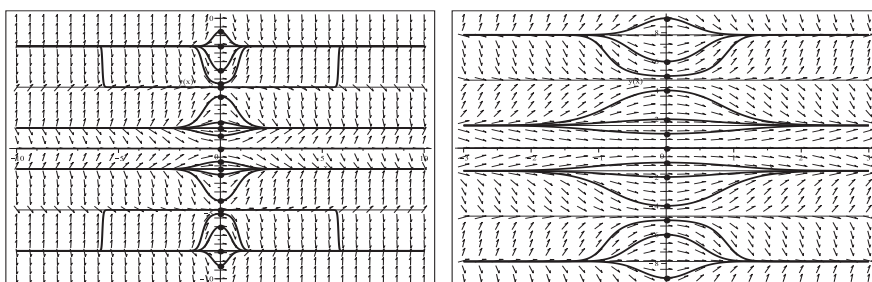


Abb. 2.156 Dateien *dber_loes_18b1..2.ps*, Richtungsfeld zur gDGI (18),
 Näherungslösungen zu den AB *pts* bzw. *pts1*,
 $(x, y) \in [-10, 10] \times [-10, 10]$ bzw. $[-3, 3] \times [-9, 9]$,
 Kommando `DEplot, method=rkf45, stepsize=0.01`

Die nachfolgende Abbildung zeigt, wie mit wachsendem $|y|$ die relevanten Abschnitte
 “schrumpfen“ und dabei sich die Anstiege der Näherungskurven vergrößern.

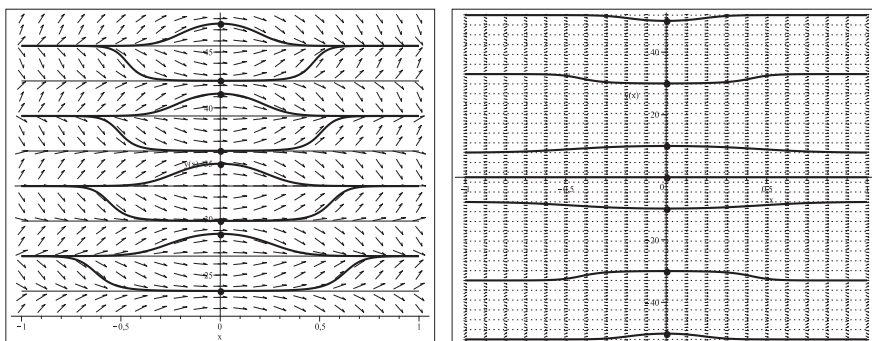


Abb. 2.157 Dateien *dber_loes_18b4..5.ps*, Richtungsfeld zur gDGI (18),
 Näherungslösungen zu den AB *pts3* bzw. *pts4*,
 $(x, y) \in [-1, 1] \times [22, 48]$ bzw. $[-1, 1] \times [-52, 52]$,
 Kommando `DEplot, method=rkf45, stepsize=0.01`

Das unterschiedliche Verhalten der Verfahren an den Gleichgewichtslagen demonstriert die folgende Abbildung. Bei größeren Bereichen können weitere Effekte auftreten.

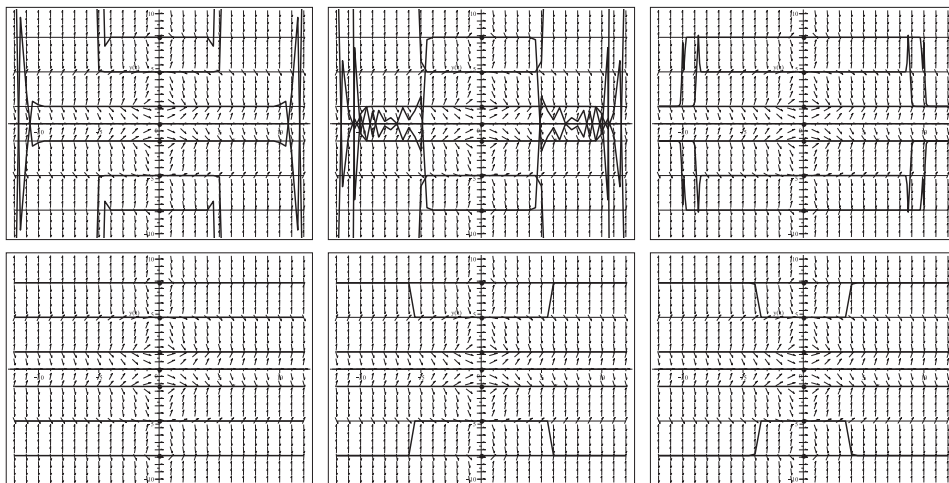


Abb. 2.158 Dateien *dber_Loes_18b30..35.ps*, Richtungsfeld zur gDGl (18), Näherungslösungen zu den AB pts2 in $[-12, 12] \times [-10, 10]$, Kommando DEplot, Methoden von l.o.n.r.u.

```
method = classical[rk3], von geringer Genauigkeit,
        classical[k4],
        classical[rk4], stepsize=0.1,
        classical[rk4], stepsize=0.01, rosenbrock,
        rkf45, rkf45, stepsize=0.01, gear,
        dverk78
```

Analytische Lösungen mittels `dsolve` gibt es nur mit den AB $y(x_0) = y_0$ in den speziellen Fällen $y(x_0) = 0$ bzw. $y(x_0) = (k + \frac{1}{2})\pi$, k ganzzahlig, mit konkreten Werten x_0 .

Die Lösungen sind die horizontalen Kurven $y(x) = 0$ bzw. $y(x) = (k + \frac{1}{2})\pi$.

Da es mit anderen AB keine expliziten Lösungen gibt, wird der Befehl `dsolve/numeric` zur näherungsweise Berechnung herangezogen.

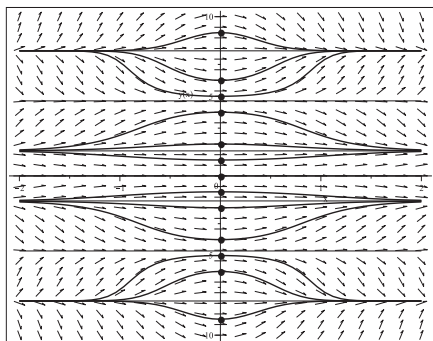


Abb. 2.159

Datei *dber_Loes_18c1.ps*,
Richtungsfeld zur gDGl (18),
Näherungslösungen zu den AB pts1,
 $(x, y) \in [-2, 2] \times [-10, 10]$,
Kommandos `dsolve/numeric/rkf45`,
`dfieldplot`, `plot`, `pointplot`

$$(19) \quad y' = y^2 e^x$$

$$y(x) = 1/(c - e^x)$$

Das Richtungsfeld der gDGL ist überall definiert.

Man bemerkt sofort die stationäre Lösung $y(x) = 0$, die mit dem Parameter $c = \infty$ korrespondiert. Sie ist eine stabile Gleichgewichtslage von unten und zugleich Grenzverlauf.

Die allgemeine parameterabhängige Lösung verweist neben moderaten Lösungen bei $c \leq 0$ auch auf absolut sehr große Anstiege der Linienelemente und damit auf Polstellen und Asymptoten. Zu letzteren sind Definitionsbereiche der Form $(-\infty, d)$ bzw. (d, ∞) zu erwarten.

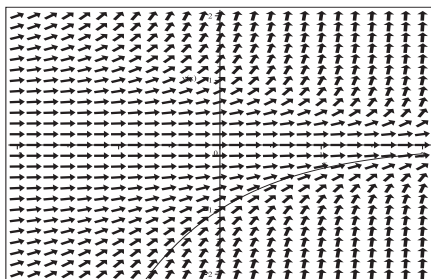


Abb. 2.160

Datei *dber_Loes_19a.ps*,
Richtungsfeld zur gDGL (19),
Kommando `dfieldplot`

Wenn man in den Parametern einen großen y -Bereich oder auch keinen angibt, dann verweisen die Warnungen des numerischen Verfahrens in `DEplot` auf die vorhandenen Polstellen der Lösungen. Über diese kann nicht hinweg integriert werden.

Zur AB $y(0) = -1$ gibt es die Lösung und zweite Grenzkurve $y(x) = -e^{-x} < 0$.

```
> # AB
pts:=[[0,-3],[0,-3/2],[0,-1],[0,-3/4],[0,-1/2],[0,0],[0,1/3],[0,1],[0,2]];
> p11:=DEplot(dgl,y(x),x=-2..2,y=-6..6,pts,dirgrid=[25,25],
method=rkf45,color=black,linecolor=black):
p12:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p11,p12);
```

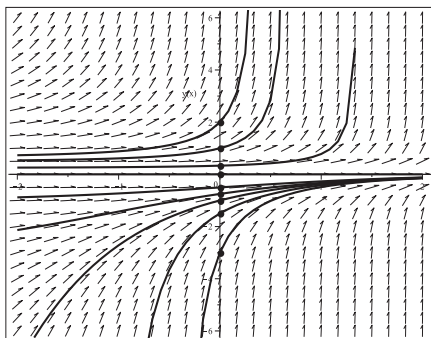


Abb. 2.161

Datei *dber_Loes_19b1.ps*,
Richtungsfeld zur gDGL (19),
Näherungslösungen zu den AB pts,
Kommando `DEplot`, `method=rkf45`

Die analytische Lösung mittels `dsolve(dg1,y(x0)=y0,y(x))` liefert die Formel

$$y(x) = \frac{y_0}{1 + y_0(e^{x_0} - e^x)},$$

speziell für die AB $y(0) = y_0$ die Lösung

$$y(x) = \frac{y_0}{1 + y_0(1 - e^x)}.$$

Außer der Nulllösung bei $y_0 = 0$ entscheiden die AB und damit die Funktion im Nenner, ob die Lösung überall definiert ist oder der Definitionsbereich ein einseitig beschränktes Intervall wird.

Fallunterscheidung bezüglich der Definitionsbereiche

(a) AB $y(x_0) = y_0 = 0$

Die stationäre Lösung lautet $y(x) = 0$.

(b) AB $y(x_0) = y_0 > 0$

Lösung ist $y(x) > 0$ im Intervall $(-\infty, d)$, $d = \ln((1 + y_0 e^{x_0})/y_0)$, weil dort der Nenner $g(x) = 1 + y_0(e^{x_0} - e^x)$ positiv und $g(d) = 0$ ist.

(c) AB $y(x_0) = y_0 \in (-e^{-x_0}, 0)$

Lösung ist $y(x) < 0$ im Intervall $(-\infty, \infty)$, weil dort der Nenner $g(x) = 1 + y_0(e^{x_0} - e^x)$ positiv ist, speziell an den Grenzsituationen $g(x) = 1$ für $y_0 = 0$, $g(x) = e^{x-x_0}$ für $y_0 = -e^{-x_0}$. Damit ist für die AB der Bereich zwischen den Kurven $y(x) = 0$ und $y(x) = -e^{-x}$ beschrieben.

(d) AB $y(x_0) = y_0 \in (-\infty, -e^{-x_0})$

Die Lösung $y(x) < 0$ liegt im Intervall (d, ∞) , $d = \ln((1 + y_0 e^{x_0})/y_0)$, weil dort der Nenner $g(x) = 1 - y_0(e^x - e^{x_0})$ wiederum positiv ist, speziell $g(x) = e^{x-x_0}$ für $y_0 = -e^{-x_0}$.

Mit dem Nenner wird also die mögliche Polstelle $d = \ln((1 + y_0 e^{x_0})/y_0)$ und der Definitionsbereich (∞, d) bzw. (d, ∞) ermittelt.

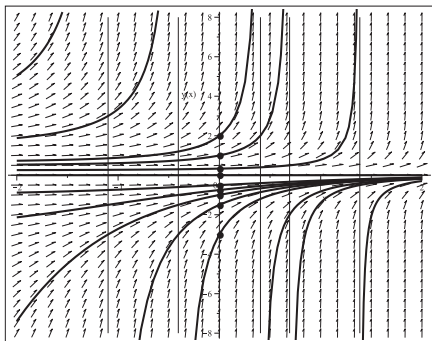


Abb. 2.162

Datei `dber_loes_19c1.ps`,

Richtungsfeld zur gDGI (19),

Lösungen zu den AB pts,

Kommandos `dsolve`, `dfieldplot`

`plot`, `pointplot`

$$(20) \quad y' = y^2 + 1$$

$$y(x) = \tan(x + c)$$

Das Richtungsfeld der gDGl ist überall definiert. Da die gDGl autonom ist, sind ihre Isoklinen (Kurven, wo der Anstieg konstant ist) horizontale Linien.

Wegen der π -periodischen Polstellen der Tangensfunktion wird das AWP zur gDGl den Definitionsbereich $(d, d + \pi)$ haben. Links und rechts davon kann man weitere Zweige der Lösung periodisch versetzt zeichnen.

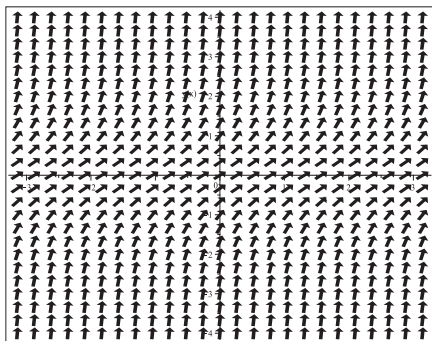


Abb. 2.163

Datei *dber_loes_20a.ps*,
Richtungsfeld zur gDGl (20),
Kommando `dfieldplot`

Wegen der meist sehr steilen Anstiege der Linienelemente sind die Berechnungen mit guten Verfahren durchzuführen. Wenn man in den Parametern einen großen y -Bereich oder auch keinen angibt, dann weisen die Warnungen des numerischen Verfahrens in DEplot auf die vorhandenen Polstellen der Lösungen hin. Über diese kann nicht hinweg integriert werden.

```
> # AB
pts:=[[0, -3], [0, -2], [0, -1], [0, 0], [0, 1], [0, 2], [0, 3]];
> p1:=DEplot(dgl,y(x),x=-Pi..Pi,y=-4..4,pts,dirgrid=[25,25],
method=rkf45,color=black,linecolor=black):
p2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p1,p2);
```

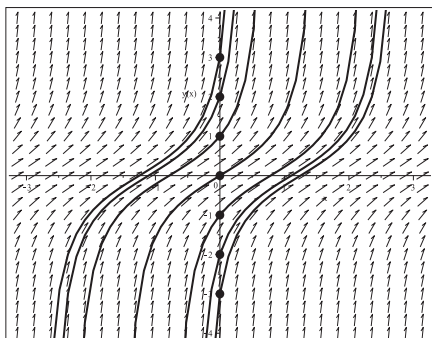


Abb. 2.164

Datei *dber_loes_20b1.ps*,
Richtungsfeld zur gDGl (20),
Näherungslösungen zu den AB pts,
Kommando `DEplot`, `method=rkf45`

Die analytische Lösung mittels `dsolve(dgl, y(x0)=y0, y(x))` liefert die Formel

$$y(x) = \tan(x - x_0 + \arctan(y_0)),$$

speziell für die AB $y(0) = y_0$, $y(0) = 0$, die Lösungen

$$y(x) = \tan(x + \arctan(y_0)) \text{ bzw. } y(x) = \tan(x).$$

Die AB entscheiden über den Definitionsbereich der Lösung. Es ist das endliche Intervall $(d, d + \pi)$, wobei $d = x_0 - \frac{\pi}{2} - \arctan(y_0)$ ist.

Aus den AB pts erhält man die Definitionsbereiche $(d, d + \pi)$ mit folgenden Werten d (Angaben mit 6 Nachkommastellen):

$$y(0) = -3, \quad d = -\frac{\pi}{2} - \arctan(-3) = -0.321750,$$

$$y(0) = -2, \quad d = -\frac{\pi}{2} - \arctan(-2) = -0.463647,$$

$$y(0) = -1, \quad d = -\frac{\pi}{2} - \arctan(-1) = -0.785398,$$

$$y(0) = 0, \quad d = -\frac{\pi}{2} - \arctan(0) = -1.570796,$$

$$y(0) = 1, \quad d = -\frac{\pi}{2} - \arctan(1) = -2.356194,$$

$$y(0) = 2, \quad d = -\frac{\pi}{2} - \arctan(2) = -2.677945,$$

$$y(0) = 3, \quad d = -\frac{\pi}{2} - \arctan(3) = -2.819842.$$

In der folgenden Abbildung werden die Lösungen sowie zusätzlich die π -periodischen Fortsetzungen und Asymptoten gezeigt.

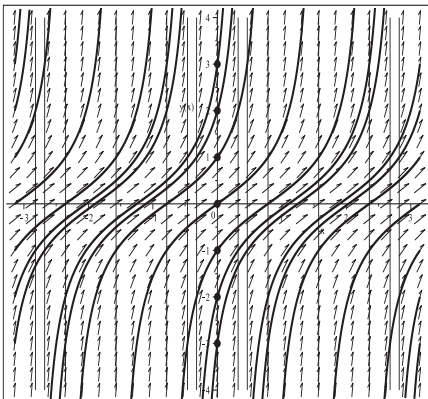


Abb. 2.165

Datei `dber_loes_20c1.ps`,

Richtungsfeld zur gDGI (20),

Lösungen zu den AB pts,

Kommandos `dsolve`, `dfieldplot`

`plot`, `pointplot`

2.4 Eindeutige Lösung von Anfangswertproblemen

Bei der Lösung von gDGI oder AWP achtet man auf den Bereich, in dem die gDGI definiert ist. Die Gestalt der gDGI zeigt oft schon an, wo Einschränkungen des Definitionsbereichs auftreten. So kann dieser z.B. eine Halbebene, ein unendlicher Streifen sein oder auch in viele Teilbereiche zerfallen. Es macht dann nur Sinn, AWP mit geeigneten AB zu lösen.

An den Bereichsgrenzen hat man manchmal spezielle Situationen wie Polstellen, asymptotisches Verhalten, steile Verläufe, Grenzwerte der Lösungen u.a. Deshalb sollte man i. Allg. die AB aus dem Inneren des Definitionsbereichs nehmen, damit die Lösungen selber und die Näherungslösungen der numerischen Verfahren "nach links und rechts" fortgesetzt werden können. Damit lässt sich gut das Verhalten an den "Rändern" beobachten.

In den bisherigen Beispielen wurden zumeist AWP mit eindeutigen Lösungen betrachtet. Das soll durch weitere AWP ergänzt werden, aber nicht mehr in dieser Ausführlichkeit wie in den Beispielen 2.4 und 2.5.

In den Betrachtungen sind geeignete Methoden, Parameter, Optionen, Bereiche u.ä. zu wählen, um verständliche Darstellungen zu erzeugen. Die Standardeinstellungen der Kommandos erweisen sich in manchen Fällen als unzureichend.

Beispiel 2.6

$$(1) \quad y' = -2y e^{-x} \cos(x)$$

$$y(x) = c e^{e^{-x} (\cos(x) - \sin(x))}$$

Das Richtungsfeld der gDGI ist überall definiert, symmetrisch zur x -Achse und zu allen AB wird es Lösungen geben. Dabei ist auch die Nulllösung.

Für $x \geq 0$ sind moderate Lösungsverläufe mit asymptotischen Verhalten bei $x \gg 0$ zu erwarten. Jedoch zu $x < 0$ schwankt der Exponent $g(x) = e^{-x} (\cos(x) - \sin(x))$ im Intervall $(-d_1, d_2)$, $d_i > 0$, und mit weiter fallendem x ist man bald nahe $(-\infty, \infty)$. Damit folgt $e^{g(x)} \in (0, \infty)$.

Die extrem unterschiedlichen Funktionswerte in solchen Bereichen erschweren natürlich sowohl die Grafik der Lösungskurven als auch die Berechnung und Darstellung von Näherungslösungen.

Die analytische Lösung mittels `dsolve(dg1, y(x0)=y0, y(x))` liefert die Formel

$$y(x, x_0, y_0) = y_0 \frac{e^{e^{-x} (\cos(x) - \sin(x))}}{e^{e^{-x_0} (\cos(x_0) - \sin(x_0))}}$$

Wegen der sehr steilen Anstiege der Kurven in Bereichen mit $x < 0$ sind die Berechnungen in `DEplot` mit sehr guten Verfahren durchzuführen. Am besten man löst einzelne AWP und tastet sich langsam an die schwierigen Situationen heran. Trotzdem versagen dann die Verfahren bei der Ermittlung der Näherung nach links.

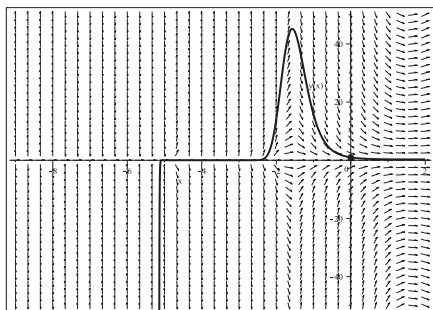


Abb. 2.166

Datei *eind_Loes_01b.ps*,
 Richtungsfeld zur gDGI (1),
 Näherungslösung zur AB $y(0) = 1$,
 Kommando `DEplot`,
`method=rkf45,stepsize=0.001`

Den Verlauf der auftretenden Funktionen demonstriert die folgende Abbildung.

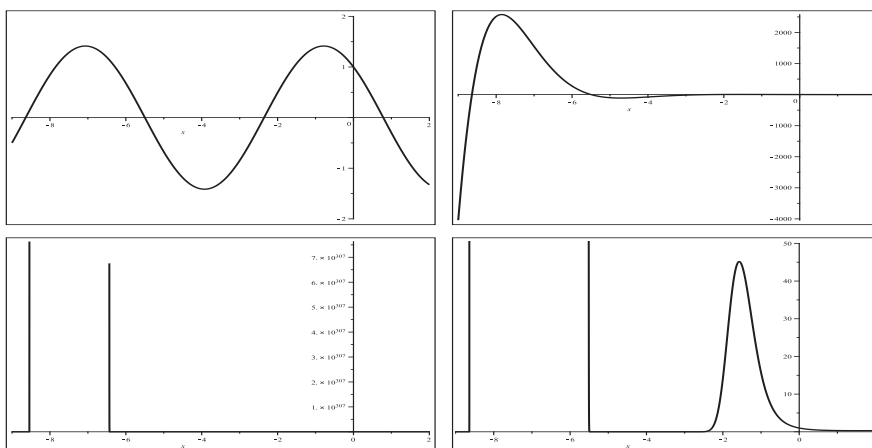


Abb. 2.167 Dateien *eind_Loes_01c1..4.ps*, gDGI (1),
 Lösung $y(x) = e^{-1} e^{e^{-x}(\cos(x) - \sin(x))}$ zur AB $y(0) = 1$, $x \in [-9, 2]$,
 Grafiken von l.o.n.r.u.
 Funktionen $\cos(x) - \sin(x)$, $e^{-x}(\cos(x) - \sin(x))$,
 $y(x)$ ohne y -Bereich, $y(x)$ mit $(x, y) \in [-9, 2] \times [0, 50]$

$y(x, 0, 1)$ mit einigen Werten, um die Größenunterschiede zu verdeutlichen:

$$\begin{aligned}
 y(\infty, 0, 1) &= e^{-1}, \\
 y(\pi/2, 0, 1) &= e^{-1} e^{-e^{-\pi/2}} = 0.298830, \\
 y(0, 0, 1) &= 1, \\
 y(-\pi/4, 0, 1) &= e^{-1} e^{e^{\pi/4} \sqrt{2}} = 8.180607, \\
 y(-\pi/2, 0, 1) &= e^{-1} e^{e^{\pi/2}} = 45.171997, \\
 y(-3\pi/4, 0, 1) &= e^{-1} = 0.367879, \\
 y(-3\pi/2, 0, 1) &= e^{-1} e^{-e^{3\pi/2}} = 1.663445e-49 \approx 0, \\
 y(-5\pi/3, 0, 1) &= e^{-1} e^{-e^{5\pi/3}(\sqrt{3}-1)/2} = 4.946153e-31, \\
 y(-7\pi/4, 0, 1) &= e^{-1} = 0.367879, \\
 y(-2\pi, 0, 1) &= e^{-1} e^{e^{2\pi}} = 1.338987e+232 \approx \infty.
 \end{aligned}$$

$$(2) \quad y' = x^{2/3}$$

$$y(x) = c + \frac{3}{5} x^{5/3}$$

Das Richtungsfeld der gDGL ist in der Halbebene $x \geq 0$ definiert und zu allen AB $y(x_0) = y_0$ mit $x_0 \geq 0$ wird es Lösungen geben, die moderate Verläufe aufweisen.

Die analytische Lösung mittels `dsolve(dg1,y(x0)=y0,y(x))` liefert die Formel

$$y(x, x_0, y_0) = y_0 + \frac{3}{5} (x^{5/3} - x_0^{5/3}).$$

Die numerischen Berechnungen mit `DEplot` bei beliebigen Verfahren kann man für $x \geq 0$ durchführen. Die Näherungen stellen eine sehr genaue Approximation der exakten Lösungen dar.

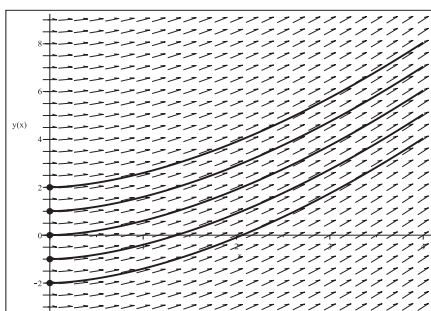


Abb. 2.168

Datei `eind_loes_02b.ps`,
 Richtungsfeld zur gDGL (2),
 Näherungslösungen zu den
 AB $y(0) = -2, -1, 0, 1, 2$,
 Kommando `DEplot, method=rkf45`

$$(3) \quad y' \cos^2(x) - y = 0$$

$$y(x) = c e^{\frac{\sin(x)}{\cos(x)}} = c e^{\tan(x)}$$

Wegen der Nullstellen der Cosinusfunktion bzw. Polstellen der Tangensfunktion sind das Richtungsfeld der gDGL und damit die Lösung in den Bereichen (unendliche Streifen) $((k - \frac{1}{2})\pi, (k + \frac{1}{2})\pi) \times (-\infty, \infty)$, k ganzzahlig, definiert. Zu allen AB $y(x_0) = y_0$ mit $x_0 \in ((k - \frac{1}{2})\pi, (k + \frac{1}{2})\pi)$ wird es Lösungen geben, die betragsmäßig wachsende Verläufe aufweisen. Die Nulllösung ergibt sich zur AB $y(x_0) = 0$ mit freiem x_0 .

Die analytische Lösung mittels `dsolve({dg1,y(x0)=y0},y(x))` liefert die Formel

$$y(x, x_0, y_0) = y_0 e^{\tan(x) - \tan(x_0)},$$

speziell $y(x, x_0, 0) = 0$.

Aber das Kommando `dsolve(dg1,y(-Pi/2)=y0,y(x))`, allgemein mit den AB $y(-\frac{\pi}{2} + k\pi) = y_0$, ergibt keine Lösung, während `dsolve(dg1,y(-Pi/2)=0,y(x))` das Ergebnis

$$y(x) = \begin{cases} 0 & \text{für } x \leq -\frac{\pi}{2}, \\ c e^{\tan(x)} & \text{für } -\frac{\pi}{2} < x < \frac{\pi}{2} \end{cases}$$

liefert, letzteres bedeutet aber nicht die Nulllösung überall.

Das ist zumindest eine differenzierbare Funktion im Intervall $(-\infty, \frac{\pi}{2})$, denn

$$\lim_{x \rightarrow -\pi/2+0} y(x) = \lim_{x \rightarrow -\pi/2+0} c e^{\tan(x)} = 0, \quad \lim_{x \rightarrow -\pi/2+0} y'(x) = \lim_{x \rightarrow -\pi/2+0} c \frac{e^{\tan(x)}}{\cos^2(x)} = 0.$$

Im Definitionsbereich werden die numerischen Berechnungen mit `DEplot` mit guten Verfahren durchgeführt. Die Näherungen stellen eine hinreichend genaue Approximation der exakten Lösungen dar.

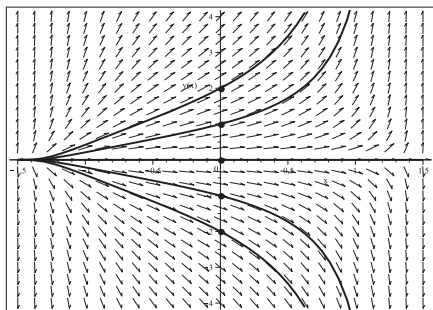


Abb. 2.169

Datei `eind_Loes_03b.ps`,

Richtungsfeld zur gDGL (3),

Näherungslösungen für $x \in (-\frac{1}{2}\pi, \frac{1}{2}\pi)$

zu den AB $y(0) = -2, -1, 1, 2$,

für $x \in (-\infty, \infty)$ mit der AB $y(0) = 0$,

Kommando `DEplot, method=rkf45`

$$(4) \quad y' = y \cos(x)$$

$$y(x) = c e^{\sin(x)}$$

Das Richtungsfeld der gDGL ist überall definiert und zu allen AB $y(x_0) = y_0$ wird es Lösungen geben, die moderate oszillierende Verläufe aufweisen. Die Nulllösung ergibt sich zur AB $y(x_0) = 0$.

Die analytische Lösung mittels `dsolve(dg1, y(x0)=y0, y(x))` liefert die Formel

$$y(x, x_0, y_0) = y_0 e^{\sin(x) - \sin(x_0)}.$$

Die numerischen Berechnungen erfolgen mit `DEplot` bei geeigneten Verfahren. Die Näherungen stellen eine sehr genaue Approximation der exakten Lösungen dar.

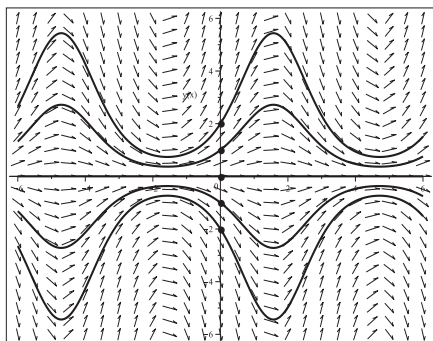


Abb. 2.170

Datei `eind_Loes_04b.ps`,

Richtungsfeld zur gDGL (4),

Näherungslösungen zu den

AB $y(0) = -2, -1, 0, 1, 2$,

Kommando `DEplot,`

`method=rkf45, stepsize=0.01`

$$(5) \quad y' = -y \sin(x)$$

$$y(x) = c e^{\cos(x)}$$

Das Richtungsfeld der gDGl ist überall definiert und zu allen AB $y(x_0) = y_0$ wird es Lösungen geben, die moderate oszillierende Verläufe aufweisen. Die Nulllösung ergibt sich zur AB $y(x_0) = 0$.

Die analytische Lösung mittels `dsolve(dgl,y(x0)=y0,y(x))` liefert die Formel

$$y(x, x_0, y_0) = y_0 e^{\cos(x) - \cos(x_0)}.$$

Die numerischen Berechnungen erfolgen mit `DEplot` bei geeigneten Verfahren. Die Näherungen stellen eine sehr genaue Approximation der exakten Lösungen dar.

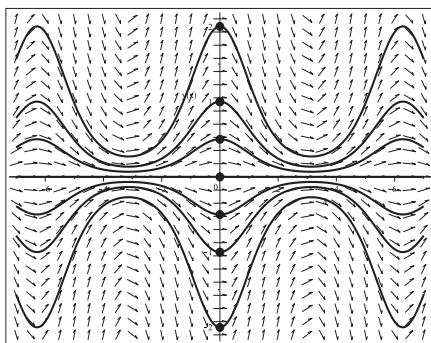


Abb. 2.171

Datei `eind_loes_05b.ps`,
 Richtungsfeld zur gDGl (5),
 Näherungslösungen zu den
 AB $y(0) = -2, -1, -\frac{1}{2}, 0, \frac{1}{2}, 1, 2$,
 Kommando `DEplot`,
`method=rkf45, stepsize=0.01`

$$(6) \quad y' = -200 x y^2$$

$$y(x) = \frac{1}{c + 100 x^2}$$

Das Richtungsfeld der gDGl ist überall definiert und zu allen AB $y(x_0) = y_0$ wird es Lösungen geben. Die Nulllösung ergibt sich zur AB $y(x_0) = 0$ (korrespondiert mit dem Parameter $c = \infty$), aber die Linienelemente können extrem steil werden, sodass es auch Situationen von Lösungen mit Polstellen und endlichen Definitionsbereichen geben wird. Dann wird es auch Zweige zu den Lösungen außerhalb dieser Intervalle geben.

Die analytische Lösung mittels `dsolve(dgl,y(x0)=y0,y(x))` liefert die Formel

$$y(x, x_0, y_0) = \frac{y_0}{1 + 100 y_0 (x^2 - x_0^2)}.$$

Fallunterscheidung bezüglich der Definitionsbereiche

(a) AB $y(x_0) = y_0 \in [0, \frac{1}{100 x_0^2})$, x_0 beliebig

Man erhält eine moderate und nicht negative Lösung $y(x, x_0, y_0)$ in $(-\infty, \infty)$, weil der Nenner überall positiv ist. Dazu gehören die speziellen Fälle der Nulllösung für die AB $y(x_0) = 0$ sowie der Lösung $y(x, 0, y_0) = \frac{y_0}{1 + 100 y_0 x^2}$ für $y(0) = y_0$.

(b) AB $y(x_0) = y_0 = \frac{1}{100x_0^2}$, $x_0 \neq 0$

Die positive Lösung ist $y(x, x_0, y_0)$ in $(0, \infty)$ bzw. $(-\infty, 0)$ mit der Polstelle $x = 0$.

(c) AB $y(x_0) = y_0 \in (\frac{1}{100x_0^2}, \infty)$, $x_0 \neq 0$

Die positive Lösung $y(x, x_0, y_0)$ liegt in (d, ∞) bzw. $(-\infty, -d)$, wobei $d = \sqrt{x_0^2 - 1/(100y_0)}$, mit den Polstellen $\pm d$.

(d) AB $y(x_0) = y_0 < 0$, x_0 beliebig

Die negative Lösung ist $y(x, x_0, y_0)$ in $(-d, d)$, wobei $d = \sqrt{x_0^2 - 1/(100y_0)}$ mit den Polstellen $\pm d$, $(-d, d) \approx (-|x_0| - \delta, |x_0| + \delta)$, $\delta > 0$.

Die numerischen Berechnungen erfolgen mit **DEplot** bei geeigneten Verfahren. Die Näherungen stellen eine sehr genaue Approximation der exakten Lösungen dar.

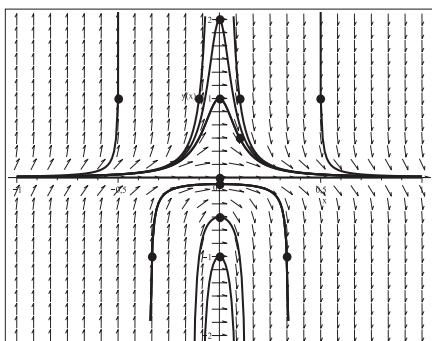


Abb. 2.172

Datei *eindLoes_06b.ps*,

Richtungsfeld zur gDGl (6),

Näherungslösungen zu den

AB $y(0) = -1, -\frac{1}{2}, -\frac{9}{109}, 0, 1, 2$, $y(\frac{1}{10}) = \frac{1}{2}$,
 $y(\pm\frac{1}{2}) = 1$, $y(\pm\frac{1}{10}) = 1$, $y(\pm\frac{1}{3}) = -1$,

Kommando **DEplot**,

`method=rkf45,stepsize=0.01`

(7) $y' = -y \sin(x) \cos^2(x)$

$$y(x) = c e^{\cos(3x)/12 + \cos(x)/4}$$

Das Richtungsfeld der gDGl ist überall definiert und zu allen AB $y(x_0) = y_0$ gibt es Lösungen, die moderate 2π -periodische Verläufe aufweisen. Die Nulllösung ergibt sich zur AB $y(x_0) = 0$. Die analytische Lösung mittels `dsolve(dg1,y(x0)=y0,y(x))` ist

$$y(x, x_0, y_0) = y_0 e^{\cos(3x)/12 + \cos(x)/4 - \cos(3x_0)/12 - \cos(x_0)/4}.$$

Die numerischen Berechnungen mittels **DEplot** liefern sehr gute Näherungen der exakten Lösungen.

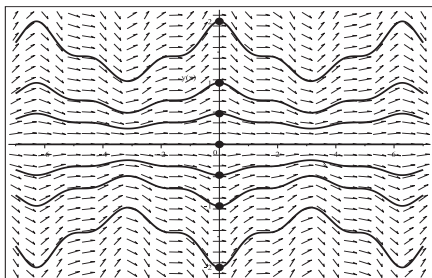


Abb. 2.173

Datei *eindLoes_07b.ps*,

Richtungsfeld zur gDGl (7),

Näherungslösungen zu den

AB $y(0) = -2, -1, -\frac{1}{2}, 0, \frac{1}{2}, 1, 2$,

Kommando **DEplot**,

`method=rkf45,stepsize=0.01`

$$(8) \quad y' = \sqrt{|x|}$$

$$y(x) = c + \begin{cases} -\frac{2}{3}(-x)^{3/2} & \text{für } x \leq 0, \\ \frac{2}{3}x^{3/2} & \text{für } x > 0 \end{cases}$$

Das Richtungsfeld der gDGl ist überall definiert und zu allen AB $y(x_0) = y_0$ gibt es Lösungen, die moderate und wachsende Verläufe aufweisen.

Die analytische Lösung mittels `dsolve(dgl, y(x0)=y0, y(x))` ist

$$y(x, x_0, y_0) = \begin{cases} -\frac{2}{3}(-x)^{3/2} & \text{für } x \leq 0, \\ \frac{2}{3}x^{3/2} & \text{für } x > 0, \end{cases} + y_0 - \begin{cases} -\frac{2}{3}(-x_0)^{3/2} & \text{für } x_0 \leq 0, \\ \frac{2}{3}x_0^{3/2} & \text{für } x_0 > 0. \end{cases}$$

Die erste Ableitung sieht man oben, die zweite lautet $y''(x) = \text{sign}(x)/(2\sqrt{|x|})$ und existiert für $x \neq 0$.

Die numerischen Berechnungen erfolgen mit `DEplot` bei geeigneten Verfahren. Die Näherungen stellen eine sehr genaue Approximation der exakten Lösungen dar.

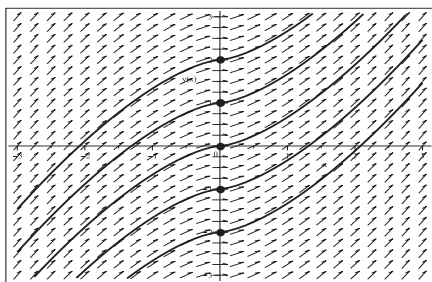


Abb. 2.174

Datei `eind_loes_08b.ps`,
 Richtungsfeld zur gDGl (8),
 Näherungslösungen zu den
 AB $y(0) = -2, -1, 0, 1, 2$,
 Kommando `DEplot, method=rkf45`

$$(9) \quad y' = 1/(1 + \tan^2(y))$$

$$y(x) = \arctan(x + c)$$

Hier wird auf die Untersuchungen im Abschnitt 2.3, Teil (4), verwiesen.

Definiert man $\frac{1}{\infty} = 0$, so ist das Richtungsfeld der gDGl überall definiert, π -periodisch bzgl. y und zu jeder AB $y(x_0) = y_0$ gibt es eine eindeutige Lösung. Die allgemeine explizite Lösung beinhaltet jedoch keine der konstanten Trajektorien $y(x) = y_k = (k + \frac{1}{2})\pi$, k ganzzahlig.

So ist es auch nicht verwunderlich, dass die Kommandos `dsolve({dgl, y(x0)=y0}, y(x))`, `dsolve({dgl, y(0)=y0}, y(x))` und `dsolve({dgl, y(x0)=(k+1/2)*Pi}, y(x))` (Ausnahme bei $k = -1$) kein Ergebnis liefern. Dies funktioniert jedoch mit einem fest gewählten Anfangswert y_0 . Die numerischen Berechnungen mit `DEplot` bei geeigneten Verfahren und Optionen liefern für beliebige AB Näherungslösungen.

2.5 Mehrdeutige Lösung von Anfangwertproblemen

Singuläre oder spezielle Stellen in der gDGL sind nicht nur Anzeichen von Einschränkungen des Definitionsbereichs der Lösungen, sondern auch verantwortlich für Situationen der Mehrdeutigkeit der Lösungen. Obwohl es zu den meisten AB des AWP eindeutige Lösungsverläufe in entsprechenden Bereichen gibt, sind es meist “wenige“ AB, von denen mehrere oder auch unendlich viele Trajektorien ausgehen können.

Im Gegensatz zur analytischen Betrachtung wird bei der numerischen Lösung des AWP nur eine Trajektorie verfolgt, meistens die konstante oder eine einfache Lösung.

Beispiel 2.7

$$(1) \quad y' \sin(x) = 2y \cos(x) \quad \text{bzw.} \quad y' \tan(x) = 2y$$

$$y(x) = \frac{c}{2} (1 - \cos(2x)) = c(1 - \cos^2(x))$$

Das Richtungsfeld der gDGL ist für $x \neq k\pi$, k ganzzahlig, definiert. Die Punkte $k\pi$ sind zuständig für die Mehrdeutigkeit der Lösung und haben Einfluss auf ihren Definitionsbereich. I. Allg. wird die Lösung eines AWP mit der AB $y(x_0) = y_0$, $x_0 \in (k\pi, (k+1)\pi)$, in diesem Intervall betrachtet.

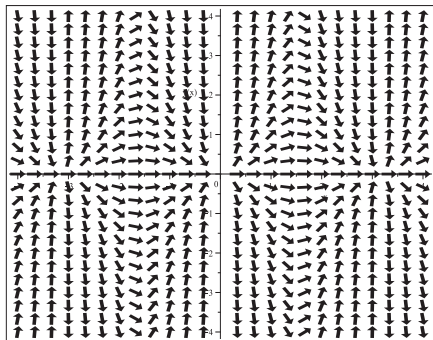


Abb. 2.175

Datei *mehrd_loes_01a.ps*,
Richtungsfeld zur gDGL (1),
Kommando `dfieldplot`

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$ mit $x_0 \neq k\pi$ sowie in den Sonderfällen $y(k\pi) = 0$ und wir berechnen sie mit dem Kommando `DEplot`. Die Lösung für die AB $y(k\pi) = 0$ ist erwartungsgemäß $y(x) = 0$, aber mit der AB $y(k\pi) = y_0 \neq 0$ erhält man keine Ergebnisse (rechte Grafik in der Abb. 2.177). Maple liefert dazu die folgende Meldung, die auf eine behebbare Singularität verweist.

```
Error, (in dsolve/numeric/checksing) ode system has a removable singularity at x=0. Initial data is restricted to {y(x) = 0}
```

Es ist sinnvoll, die Näherungslösungen nur in den entsprechenden Definitionsbereichen $(k\pi, (k+1)\pi)$ zu betrachten. Geraten die numerischen Verfahren an die Grenzen der Bereiche, wird es problematisch für diese. Abhängig vom Verfahren und den Optionen wird über die Grenzen hinweg integriert und/oder es gibt nahe der Grenzen “Ausreißer“ der Kurven.

```

> # AB
pts:=[[Pi/2,-3],[Pi/2,-2],[Pi/2,-1],[Pi/2,0],[Pi/2,1],[Pi/2,2],[Pi/2,3]];
pts1:=[[1,1]]; pts2:=[[0,0]]; pts3:=[[0,1]];

> p1:=DEplot(dg1,y(x),x=-1..4,y=-3.1..3.1,pts,dirgrid=[25,25],
            method=rkf45,stepsize=0.01,color=black,linecolor=black):
p2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p1,p2);

```

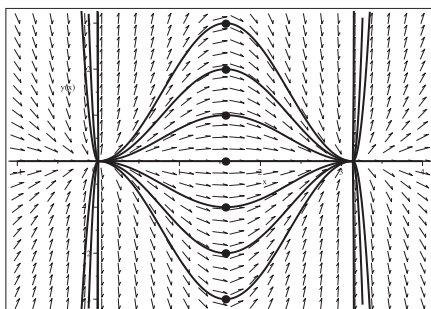


Abb. 2.176

Datei *mehrd_loes_01b1.ps*,
 Richtungsfeld zur gDGI (1),
 Näherungslösungen zu den AB pts ,
 $(x, y) \in [-1, 4] \times [-3.1, 3.1]$,
 Kommando `DEplot`,
`method=rkf45, stepsize=0.01`

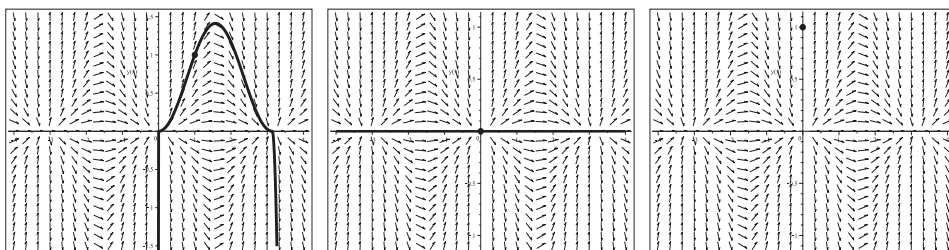


Abb. 2.177 Dateien *mehrd_loes_01b2..4.ps*, Richtungsfeld zur gDGI (1),
 Näherungslösung zu $y(x) = \frac{1 - \cos(2x)}{1 - \cos(2)}$, AB $y(1) = 1$ in $[-4, 4] \times [-1.5, 1.5]$,
 zu $y(x) = 0$, AB $y(0) = 0$, aber keine Lösung mit $y(0) = 1$ (v.l.n.r),
 Kommando `DEplot`, `method=rkf45, stepsize=0.01`

Das Kommando `dsolve(dg1, y(x0)=y0, y(x))` liefert die Lösung

$$y(x) = y_0 \frac{1 - \cos(2x)}{1 - \cos(2x_0)} = y_0 \frac{1 - \cos^2(x)}{1 - \cos^2(x_0)},$$

`dsolve(dg1, y(x0)=0, y(x))` ergibt $y(x) = 0$. Lösungen gibt es zu allen AB $y(x_0) = y_0$ mit $x_0 \neq k\pi$.

Der Nenner in der Lösungsformel verbietet die Anfangsstelle $x_0 = k\pi$.

Aber in der parameterabhängigen Darstellung $\frac{c}{2}(1 - \cos(2x))$ erkennt man beliebig viele Lösungen zum reellen Faktor c , z.B. mit der AB $y(0) = 0$.

So liefert `dsolve(dg1, y(k*Pi)=0, y(x))` und insbesondere `dsolve(dg1, y(0)=0, y(x))` wie gewünscht die mehrdeutige Lösung $y(x) = \frac{c}{2}(1 - \cos(2x))$ und das Kommando `dsolve(dg1, y(0)=1, y(x))` kein Ergebnis.

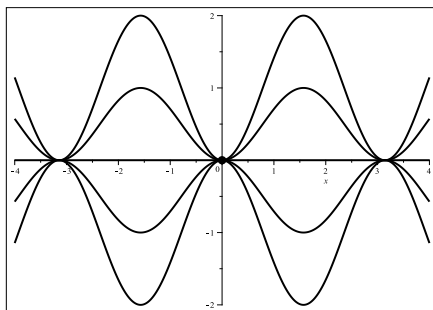


Abb. 2.178

Datei *mehrd_Loes_01c1.ps*,
 Richtungsfeld zur gDGL (1),
 Lösungen $\frac{c}{2}(1 - \cos(2x))$, $c = -2(1)2$,
 zur AB $y(0) = 0$,
 Kommandos `dsolve`, `plot`, `pointplot`

$$(2) \quad y' \sin(x) = y \cos(x) \quad \text{bzw.} \quad y' \tan(x) = y$$

$$y(x) = c \sin(x)$$

Man würde meinen, diese gDGL liefert analoge Ergebnisse zu Teil (1), aber wir bemerken doch einige kleine Unterschiede.

Das Richtungsfeld der gDGL ist für $x \neq k\pi$, k ganzzahlig, definiert. Die Punkte $k\pi$ sind verantwortlich für die Mehrdeutigkeit der Lösung und haben Einfluss auf ihren Definitionsbereich. Im Allg. wird die Lösung eines AWP mit der AB $y(x_0) = y_0$, $x_0 \in (k\pi, (k+1)\pi)$, in diesem Intervall betrachtet.

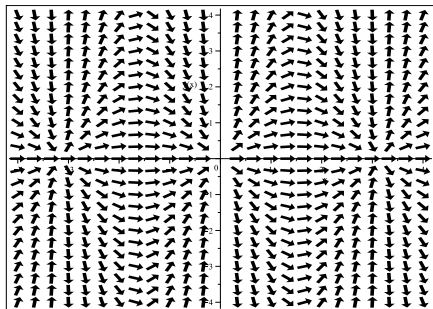


Abb. 2.179

Datei *mehrd_Loes_02a.ps*,
 Richtungsfeld zur gDGL (2),
 Kommando `dfieldplot`

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$ mit $x_0 \neq k\pi$ sowie in den Sonderfällen $y(k\pi) = 0$, $k \neq 0$, und wir berechnen sie mit dem Kommando `DEplot`.

Die Lösung für die AB $y(k\pi) = 0$, $k \neq 0$, ist erwartungsgemäß $y(x) = 0$. Aber weder mit der einfachen AB $y(0) = 0$ noch mit $y(k\pi) = y_0 \neq 0$ erhält man ein Ergebnis (beide rechte Grafiken in der Abb. 2.181). Maple liefert dazu die folgende Fehlermeldung.

```
Error, (in dsolve/numeric/make_proc) ode system is singular
at the initial point
```

Es ist zwar sinnvoll, die Näherungslösungen nur in den entsprechenden Definitionsbereichen $(k\pi, (k+1)\pi)$ zu betrachten. Geraten aber die numerischen Verfahren an die Grenzen der Bereiche, wird meist geschickt über die Grenzen hinweg integriert, sodass eine gute Fortsetzung der Kurve entsteht.


```

> # AB
pts:=[[Pi/2,-3],[Pi/2,-2],[Pi/2,-1],[Pi/2,0],[Pi/2,1],[Pi/2,2],[Pi/2,3]];
pts1:=[[1,1]]; pts2:=[[Pi,0]]; pts3:=[[0,0]]; pts4:=[[0,1]];

> p1:=DEplot(dg1,y(x),x=-1..4,y=-3.1..3.1,pts,dirgrid=[25,25],
            method=rkf45,stepsize=0.01,color=black,linecolor=black):
p2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=16):
display(p1,p2);

```

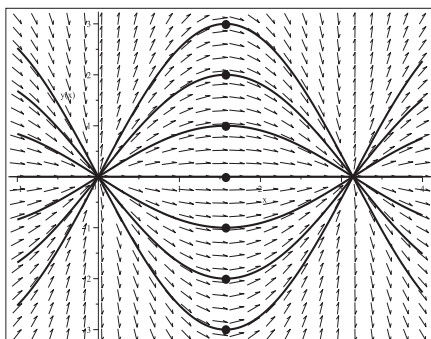


Abb. 2.180

Datei *mehrd_loes_02b1.ps*,
 Richtungsfeld zur gDGL (2),
 Näherungslösungen zu den AB pts,
 $(x, y) \in [-1, 4] \times [-3.1, 3.1]$,
 Kommando DEplot,
 method=rkf45, stepsize=0.01

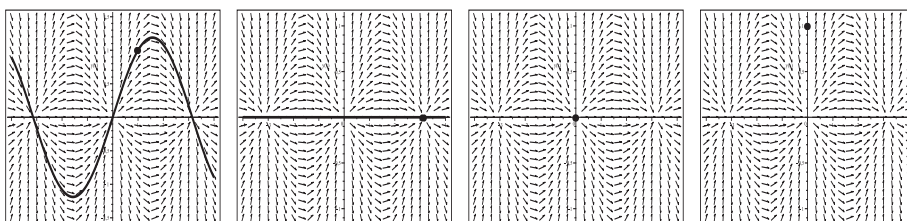


Abb. 2.181 Dateien *mehrd_loes_02b2..5.ps*, Richtungsfeld zur gDGL (2),
 Näherungslösung zu $y(x) = \frac{\sin(x)}{\sin(1)}$, AB $y(1) = 1$ in $[-4, 4] \times [-1.5, 1.5]$,
 zu $y(x) = 0$, AB $y(\pi) = 0$, aber keine Lösung mit AB $y(0) = 0, y(0) = 1$
 (v.l.n.r), Kommando DEplot, method=rkf45, stepsize=0.01

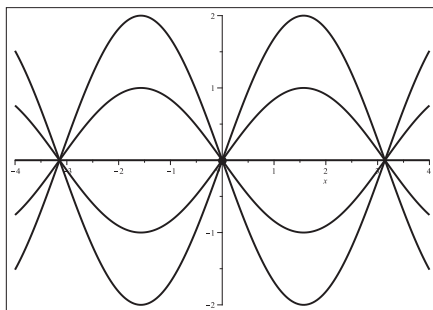
Das Kommando `dsolve(dg1,y(x0)=y0,y(x))` liefert die Lösung

$$y(x) = y_0 \frac{\sin(x)}{\sin(x_0)},$$

`dsolve(dg1,y(x0)=0,y(x))` ergibt $y(x) = 0$. Lösungen gibt es zu allen AB $y(x_0) = y_0$ mit $x_0 \neq k\pi$. Der Nenner in der Lösungsformel verbietet die Anfangsstelle $x_0 = k\pi$.

Aber in der parameterabhängigen Darstellung $c \sin(x)$ erkennt man beliebig viele Lösungen zum reellen Faktor c , z.B. mit der AB $y(0) = 0$.

So liefert `dsolve(dg1,y(k*Pi)=0,y(x))` sowie insbesondere `dsolve(dg1,y(0)=0,y(x))` und `dsolve(dg1,y(Pi)=0,y(x))` wie gewünscht die mehrdeutige Lösung $y(x) = c \sin(x)$, aber `dsolve(dg1,y(0)=1,y(x))` kein Ergebnis.

**Abb. 2.182**

Datei *mehrd_loes_02c1.ps*,
 Richtungsfeld zur gDGl (2),
 Lösungen $c \sin(x)$, $c = -2(1)2$,
 zur AB $y(0) = 0$,
 Kommandos `dsolve`, `plot`, `pointplot`

$$(3) \quad y' \sin(2x) = 2y \cos(2x) \quad \text{bzw.} \quad y' \tan(2x) = 2y$$

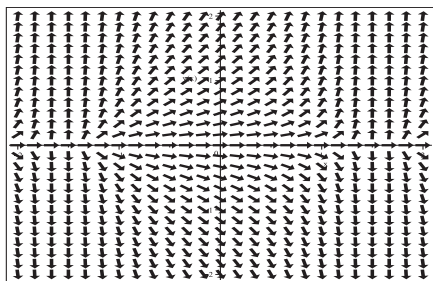
$$y(x) = c \sin(2x)$$

Mit der linearen Transformation $2x = t$ der unabhängigen Variablen erhält man den Fall (2) und damit analoge Ergebnisse "gestaucht" um den Faktor 2.

$$(4) \quad y' \cos^2(x) - y = 0$$

$$y(x) = c e^{\frac{\sin(x)}{\cos(x)}} = c e^{\tan(x)}$$

Zahlreiche Informationen dazu sind schon im Abschnitt 2.4 Beispiel 2.6 Teil (4) zu finden.

**Abb. 2.183**

Datei *mehrd_loes_04a.ps*,
 Richtungsfeld zur gDGl (4),
 Kommando `dfieldplot`

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$ mit Ausnahme der Situation $x_0 = (k + \frac{1}{2})\pi$, k ganzzahlig, und $y_0 \neq 0$.

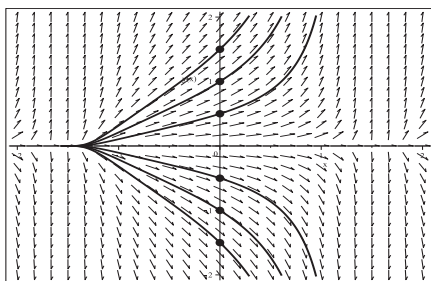
Bei $y(x_0) = 0$ wird numerisch immer und überall die Nulllösung berechnet.

```

> # AB
pts:=[[0,-3/2],[0,-1],[0,-1/2],[0,1/2],[0,1],[0,3/2]];
pts2:[[1/2,0]]; pts3:[[0,0]]; pts4:[[-Pi/2,0]]; # Nullloesung ueberall
pts5:[[-Pi/2,1]]; # keine Naeherungsloesung

> p11:=DEplot(dgl,y(x),x=-2..2,y=-2..2,pts,dirgrid=[25,25],
method=rkf45,stepsize=0.01,color=black,linecolor=black):
p12:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=20):
display(p11,p12);

```

**Abb. 2.184**

Datei *mehrd_Loes_04b1.ps*,
 Richtungsfeld zur gDGI (4),
 Näherungslösungen zu den AB pts
 in $(-\frac{\pi}{2}, \frac{\pi}{2})$, $(x, y) \in [-2, 2] \times [-2, 2]$,
 Kommando `DEplot`,
`method=rkf45, stepsize=0.01`

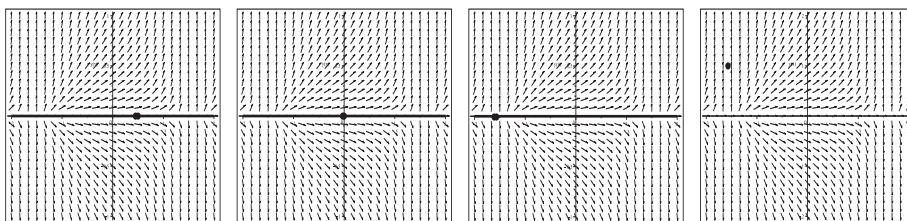


Abb. 2.185 Dateien *mehrd_Loes_04b3..6.ps*, Richtungsfeld zur gDGI (4),
 Näherungslösung zu $y(x)=0$, AB $y(x_0)=0$, $x_0 = \frac{1}{2}, 0, -\frac{\pi}{2}$,
 aber keine Lösung mit AB $y(-\frac{\pi}{2})=1$ (v.l.n.r.),
 Kommando `DEplot`, `method=rkf45, stepsize=0.01`

Beim letzten Fall in dieser Abbildung liefert Maple die folgende Fehlermeldung.

```
Warning, plot may be incomplete, the following error(s) were issued:
cannot evaluate the solution past the initial point,
problem may be complex, initially singular or improperly set up
```

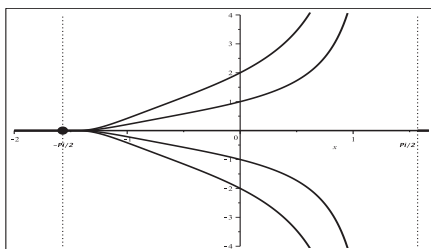
Aber es ist besonders hervorzuheben, dass die numerische Lösung zur AB $y(-\frac{\pi}{2})=0$ die Nulllösung ist, während `dsolve(dgl, y(-Pi/2)=0, y(x))` das Ergebnis

$$y(x) = \begin{cases} 0 & \text{für } x \leq -\frac{\pi}{2}, \\ c e^{\tan(x)} & \text{für } -\frac{\pi}{2} < x < \frac{\pi}{2} \end{cases}$$

liefert.

Das ist der Punkt, wo die Mehrdeutigkeit der Lösung ins Spiel kommt.

Zur AB $y(-\frac{\pi}{2})=0$ gibt es im Intervall $(-\infty, \frac{\pi}{2})$ neben der Nulllösung weitere beliebig viele Lösungen mit reellem Parameter c .

**Abb. 2.186**

Datei *mehrd_Loes_04c1.ps*,
 Nulllösung in $(-\infty, \infty)$,
 Lösungen $c e^{\tan(x)}$, $x \in (-\frac{\pi}{2}, \frac{\pi}{2})$, 0 für $x \leq -\frac{\pi}{2}$,
 zur AB $y(-\frac{\pi}{2})=0$, Parameter $c = -2(1)2$,
 Kommandos `dsolve`, `plot`, `pointplot`

$$(5) \quad y' \cos(x) + y \tan(x) = 0$$

$$y(x) = c e^{-\frac{1}{\cos(x)}}$$

Wegen der Nullstellen der Cosinusfunktion und zugleich Polstellen der Tangensfunktion sind das Richtungsfeld der gDGL und damit die Lösung in den Bereichen (unendliche Streifen) $((k - \frac{1}{2})\pi, (k + \frac{1}{2})\pi) \times (-\infty, \infty)$, k ganzzahlig, definiert. Zu allen AB $y(x_0) = y_0$ mit $x_0 \in ((k - \frac{1}{2})\pi, (k + \frac{1}{2})\pi)$ wird es Lösungen geben, die moderate Verläufe aufweisen. Die Nulllösung ergibt sich zur AB $y(x_0) = 0$ mit freiem x_0 .

Die analytische Lösung mittels `dsolve({dgl,y(x0)=y0},y(x))` liefert die Formel

$$y(x, x_0, y_0) = y_0 e^{-\frac{1}{\cos(x)} + \frac{1}{\cos(x_0)}},$$

speziell $y(x, x_0, 0) = 0$.

Aber das Kommando `dsolve(dgl,y(-Pi/2)=y0,y(x))`, und allgemein mit den AB $y(-\frac{\pi}{2} + k\pi) = y_0$, ergibt keine Lösung, während `dsolve(dgl,y(x0)=0,y(x))` für beliebiges festes x_0 das Ergebnis $y(x) = 0$ liefert.

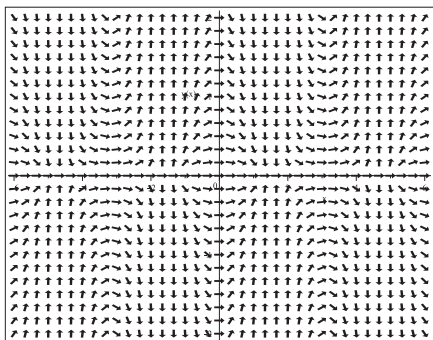


Abb. 2.187

Datei *mehrd_Loes.05a.ps*,
Richtungsfeld zur gDGL (5),
Kommando `dfieldplot`

Numerische Lösungen gibt es zu allen AB $y(x_0) = y_0$ mit Ausnahme der Situation $x_0 = (k + \frac{1}{2})\pi$, k ganzzahlig, und $y_0 \neq 0$.

Bei $y(x_0) = 0$ wird numerisch immer und überall die Nulllösung berechnet.

```
> # AB
pts:=[[0,-3/2],[0,-1],[0,-1/2],[0,1/2],[0,1],[0,3/2]];
pts1:[[Pi,1]];
pts2:[[1/2,0]]; pts3:[[0,0]]; pts4:[[-Pi/2,0]]; # Nullloesung ueberall
pts5:[[-Pi/2,1]]; # keine Naehungsloesung
pts6:[[Pi/2,1]];

> p11:=DEplot(dgl,y(x),x=-2..2,y=-2..2,pts,dirgrid=[25,25],
method=rkf45,color=black,linecolor=black):
p12:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=20):
display(p11,p12);
```

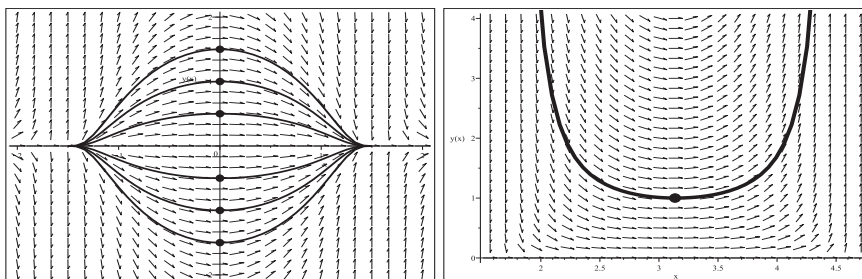


Abb. 2.188 Dateien *mehrd_loes_05b1..2.ps*, Richtungsfeld zur gDGl (5),
 Näherungslösungen zu den AB pts in $[-2, 2] \times [-2, 2]$,
 sowie zur AB $y(\pi) = 1$ in $[\frac{\pi}{2}, \frac{3\pi}{2}] \times [0, 4]$ (v.l.n.r),
 Kommando `DEplot, method=rkf45`

Beim ersten Fall in dieser Abbildung liefert Maple die folgende Warnung, weil die numerischen Verfahren nicht bis an die Sprungstellen $\pm \frac{\pi}{2}$ herankommen.

```
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution further right of 1.5707321,
maxfun limit exceeded (see ?dsolve,maxfun for details)
cannot evaluate the solution further left of -1.5707321,
maxfun limit exceeded (see ?dsolve,maxfun for details)
```

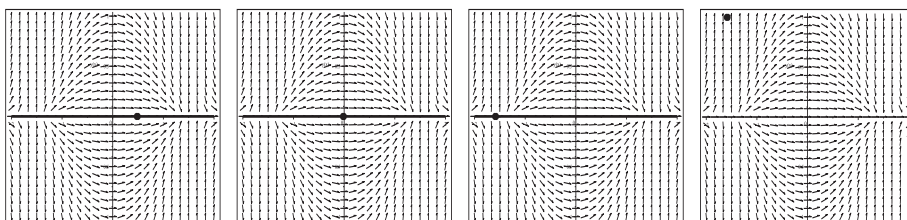


Abb. 2.189 Dateien *mehrd_loes_05b3..6.ps*, Richtungsfeld zur gDGl (5),
 Näherungslösung zu $y(x) = 0$, AB $y(x_0) = 0$, $x_0 = \frac{1}{2}, 0, -\frac{\pi}{2}$,
 aber keine Lösung mit der AB $y(-\frac{\pi}{2}) = 1$ (v.l.n.r),
 Kommando `DEplot, method=rkf45`

Beim letzten Fall in dieser Abbildung liefert Maple die folgende Fehlermeldung.

```
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution past the initial point,
problem may be complex, initially singular or improperly set up
```

Zu den analytischen Lösungen mittels `dsolve` mit ihren Fortsetzungen, die dann 2π -periodisch sind, ist noch Folgendes zu bemerken.

Man unterscheidet z.B. mit $c > 0$ bei den Definitionsbereichen (Streifen) zwischen dem

Gebiet $((2k+1)\frac{\pi}{2}, (2k+3)\frac{\pi}{2}) \times (0, d)$ für ungerade Werte k , wo dann die innenseitigen Grenzwerte an den Intervallenden Null sind, d.h.

$$\lim_{x \rightarrow \mp\pi/2 \pm 0} y(x) = \lim_{x \rightarrow \mp\pi/2 \pm 0} c e^{-1/\cos(x)} = 0,$$

$$\lim_{x \rightarrow \mp\pi/2 \pm 0} y'(x) = \lim_{x \rightarrow \mp\pi/2 \pm 0} -c \tan(x)/\cos(x) e^{-1/\cos(x)} = 0,$$

und dem Gebiet $((2k+1)\frac{\pi}{2}, (2k+3)\frac{\pi}{2}) \times (0, \infty)$ für gerade Werte k , wo dann diese Grenzwerte an den Intervallenden unendlich werden.

Im ersten Fall sind auch die Ableitungen an den Intervallenden Null.

Damit kann man folgern, dass im Intervall $[-\frac{\pi}{2}, \frac{\pi}{2})$ das AWP mit der AB $y(-\frac{\pi}{2}) = 0$, die Lösungsmannigfaltigkeit (mehrdelige Lösung)

$$y(x) = 0 \text{ und } y(x) = \begin{cases} 0 & \text{für } x = -\frac{\pi}{2}, \\ c e^{-\frac{1}{\cos(x)}} & \text{für } -\frac{\pi}{2} < x < \frac{\pi}{2}, \end{cases} \quad c \text{ reell,}$$

besitzt.

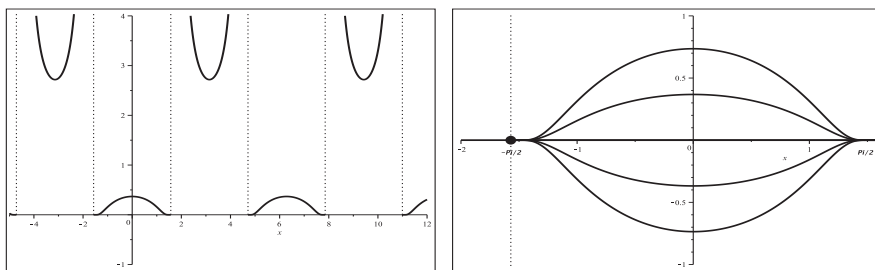


Abb. 2.190 Dateien *mehrdLoes_05c1..2.ps*,

Funktion $e^{-1/\cos(x)}$ in $[-5, 12] \times [-1, 4]$ mit Periode 2π und Sprungstellen, mehrdeutige Lösungen 0 und $ce^{-1/\cos(x)}$, $x \in [-\frac{\pi}{2}, \frac{\pi}{2})$, AB $y(-\frac{\pi}{2}) = 0$, Parameter $c = -2(1)2$ (v.l.n.r), Kommandos `dsolve`, `plot`, `pointplot`

$$(6) \quad y'^2 - 1 = 0$$

$$y_1(x) = x + c, \quad y_2(x) = -x + c$$

Die Nichtlinearität in der gDGL verhindert das Zeichnen des Richtungsfeldes. Zugleich ergeben sich die beiden abgeleiteten gDGL $y' - 1 = 0$ und $y' + 1 = 0$ mit ihren eigenen Richtungsfeldern und verschiedenen allgemeinen Lösungen $y_{1,2}(x)$, die jeweils eine Schar von parallelen Geraden mit dem Anstieg 1 bzw. -1 darstellen.

Nur in diesen Fällen kann eine numerische Lösung mittels `DEplot` erzeugt werden. Für die gDGL (6) hat man die folgende die Fehlermeldung.

Error,

(in `DEtools/DEplot/CheckDE`) unable to isolate DES for required derivatives

```

> # 2 gDGl betrachten
  dglp:=diff(y(x),x)-1; dglm:=diff(y(x),x)+1;
> # AB
  pts:=[[0,-1],[0,0],[0,1]];
> p1:=DEplot(dglp,y(x),x=-1..1,y=-1..1,pts,dirgrid=[25,25], # dglm
  method=rkf45,color=black,linecolor=black):
  p2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=20):
  display(p1,p2);

```

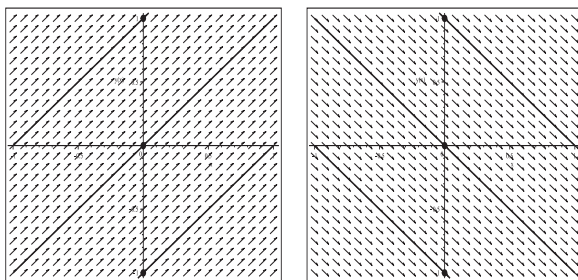


Abb. 2.191 Dateien *mehrd_loes_06b1..2.ps*, Richtungsfeld zu den abgeleiteten gDGl, Näherungslösungen zu den AB pts in $[-1, 1] \times [-1, 1]$, Kommando `DEplot`, `method=rkf45`

Die analytische Lösung mittels `dsolve({dgl,y(x0)=y0},y(x))` liefert die Formeln $y(x, x_0, y_0) = -x + y_0 + x_0$, $y(x) = x + y_0 - x_0$.

Stimmen jedoch beim Befehlsaufruf x_0 und y_0 überein, also `dsolve(dgl,y(z)=z,y(x))`, dann wird nur die ansteigende Lösungskurve berücksichtigt.

Zu den AB $y(x_0) = x_0$, $y(-x_0) = -x_0$, $y(0) = 0$, $y(1) = 1$, $y(-1) = -1$ ergibt sich stets $y(x) = x$. Natürlich liegt die Mehrdeutigkeit mit den beiden Lösungskomponenten zur gDGl (6) vor.

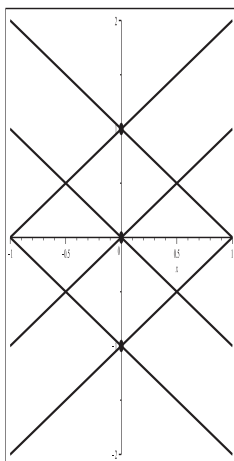


Abb. 2.192

Datei *mehrd_loes_06c1.ps*, Mehrdeutige Lösung zu (6), Lösungskurven $x + c$ und $-x + c$ für $c = -1(1)1$, Kommandos `dsolve`, `plot`, `pointplot`

$$(7) \quad y' = y/(2x) + 1$$

$$y(x) = 2x + c\sqrt{x}$$

Untersuchungen für diese gDGL, insbesondere zur Numerik, findet der Leser ausführlicher im Bsp. 2.4 Teil (16).

Man erkennt, dass $x = 0$ eine kritische Stelle ist, wo zwar die gDGL nicht definiert ist und die Richtungselemente vertikal sind, aber die Lösungsfunktion berechnet werden kann. Mit der AB $y(0) = y_0 \neq 0$ findet `dsolve` keine Lösung, jedoch wird bei $y(0) = 0$ ebenfalls das obige allgemeine parameterabhängige Ergebnis $y(x, c) = 2x + c\sqrt{x}$ geliefert. Damit zeigt sich die Mehrdeutigkeit der Lösung.

So passt die Lösung $2x - \sqrt{x}$ sowohl zur AB $y(1) = 1$ als auch zur AB $y(0) = 0$.

Zur zweiten AB gibt es auch die einfache Lösung $y(x) = 2x$.

Zur AB $y(x_0) = y_0$, $x_0 < 0$, ist die Lösung $y(x) = 2x + \frac{y_0 - 2x_0}{\sqrt{-x_0}} \sqrt{-x}$, $x \in (-\infty, 0)$. Speziell zu $y(-1) = y_0$ folgt $y(x) = 2x + (y_0 + 2)\sqrt{-x}$, $x < 0$.

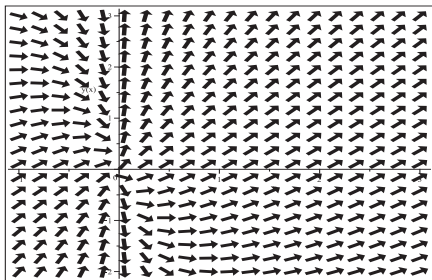


Abb. 2.193

Datei `mehrd_Loes_07a.ps`,
Richtungsfeld zur gDGL (7),
Kommando `dfieldplot`

Mit dem Kommando `DEplot` erhält man ganz verschiedene Ergebnisse.

Dabei beschränken wir uns auf das numerische Verfahren `rkf45` sowie eine entsprechende Wahl von Integrationsbereich und Parameter. Die Auswirkungen der Singularität an der Stelle Null sind in Bsp. 2.4 (16) untersucht worden.

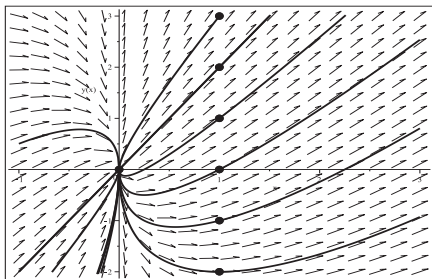


Abb. 2.194 Datei `mehrd_Loes_07b1.ps`,

Richtungsfeld zur gDGL (7),
Näherungslösungen zu den AB
 $[x_0, y_0] = [0, 0]$, $[1, -2(1)3]$
in $[-1, 3] \times [-2, 3]$,
Kommando `DEplot`,
`method=rkf45, stepsize=0.01`

Das Kommando `dsolve(dgl, y(x0)=y0, y(x))` liefert die Lösung

$$y(x, x_0, y_0) = 2x + \frac{y_0 - 2x_0}{\sqrt{x_0}} \sqrt{x}.$$

`dsolve(dgl, y(x0)=0, y(x))` ergibt $y(x) = 2x - 2\sqrt{x_0}x$. Keine Lösungen gibt es zu den AB $y(0) = y_0$ mit $y_0 \neq 0$.

In der parameterabhängigen Darstellung $2x + c\sqrt{x}$ erkennt man beliebig viele Lösungen zum reellen Faktor c mit der AB $y(0) = 0$, was die Mehrdeutigkeit der Lösung charakterisiert.

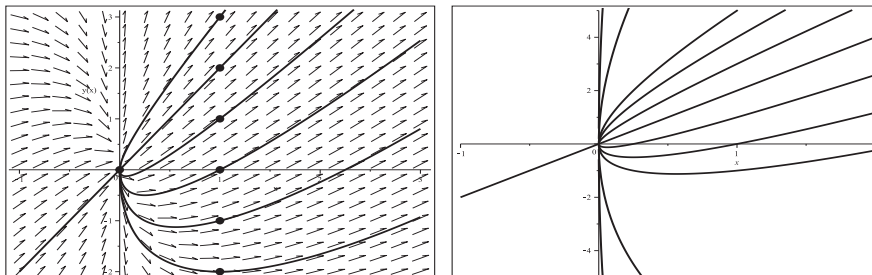


Abb. 2.195 Dateien *mehrd_loes_07c1..2.ps*, mit/ohne Richtungsfeld zur gDGL (7), AWP mit AB $y(0) = 0$, Lösungen $2x + c\sqrt{x}$, $c = -4(1)1$ (links), und $2x + c\sqrt{x}$, $c = -30, -10, -3(1)3, 10, 30$ (rechts), Kommandos `dsolve`, `plot`, `pointplot`

$$(8) \quad y' = y/x + 1$$

$$y(x) = x(c + \ln(x))$$

Untersuchungen für diese gDGL, besonders zur Numerik, findet der Leser ausführlicher im Bsp. 2.4 Teil (17).

Man sieht, dass $x = 0$ eine kritische Stelle ist. Wegen $\lim_{x \rightarrow 0} x \ln(|x|) = 0$ kann man $y(0) = 0$ definieren und damit die Lösung auf das Intervall $[0, \infty)$ fortsetzen.

Mit der AB $y(0) = y_0 \neq 0$ findet `dsolve` keine Lösung, jedoch wird bei $y(0) = 0$ das allgemeine parameterabhängige Ergebnis $y(x, c) = x(c + \ln(x))$ geliefert. Damit zeigt sich die Mehrdeutigkeit der Lösung, wenn man den Punkt $y(0) = 0$ bei den Trajektorien hinzunimmt.

Zur AB $y(x_0) = y_0$, $x_0 < 0$, ist die Lösung $y(x) = x\left(\frac{y_0}{x_0} - \ln(-x_0) + \ln(-x)\right)$, $x \in (-\infty, 0)$.

Speziell folgt zu $y(-1) = y_0$ die Lösung $y(x) = x(-y_0 + \ln(-x))$, $x < 0$.

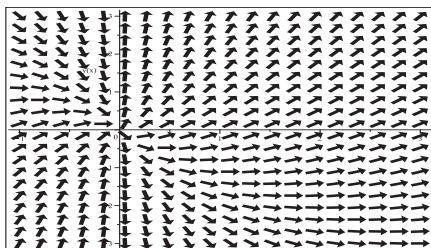


Abb. 2.196

Datei *mehrd_loes_08a.ps*,
Richtungsfeld zur gDGL (8),
Kommando `dfieldplot`

Mit dem Kommando `DEplot` erhält man ganz verschiedene Ergebnisse. Dabei beschränken wir uns auf das numerische Verfahren `rkf45` sowie eine entsprechende Wahl von Integrationsbereich und Parameter. Näherungsverfahren können mit der AB $y(0) = y_0$ nicht starten, aber durchaus über diese Stelle hinwegintegrieren. Die Auswirkungen der Singularität an der Stelle Null sind in Bsp. 2.4 (17) untersucht worden.

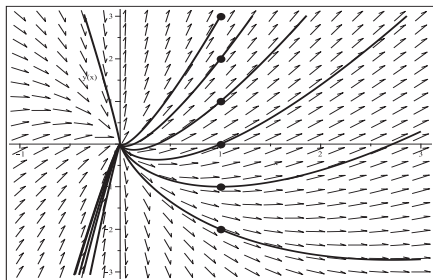


Abb. 2.197 Datei *mehrd_Loes_08b1.ps*,
Richtungsfeld zur gDGI (8),
Näherungslösungen zu den AB
 $[x_0, y_0] = [1, -2(1)3]$
in $[-1, 3] \times [-3, 3]$,
Kommando `DEplot`,
`method=rkf45, stepsize=0.01`

Das Kommando `dsolve(dgl, y(x0)=y0, y(x))` liefert die Lösung

$$y(x, x_0, y_0) = x \left(\frac{y_0}{x_0} - \ln(x_0) + \ln(x) \right),$$

`dsolve(dgl, y(x0)=0, y(x))` ergibt $y(x) = x(-\ln(x_0) + \ln(x))$. Keine Lösungen gibt es zu den AB $y(0) = y_0$ mit $y_0 \neq 0$.

In der parameterabhängigen Darstellung

$$y(x) = \begin{cases} 0 & \text{für } x = 0, \\ x(c + \ln(x)) & \text{für } x > 0 \end{cases}$$

erkennt man beliebig viele Lösungen zum reellen Faktor c mit der AB $y(0) = 0$, was die Mehrdeutigkeit der Lösung charakterisiert.

Die Ableitung dieser Lösung ist $y'(x) = 1 + c + \ln(x)$, sodass $\lim_{x \rightarrow 0^+} y'(x) = -\infty$ gilt. Bei numerischen Verfahren ist letzteres zu beachten.

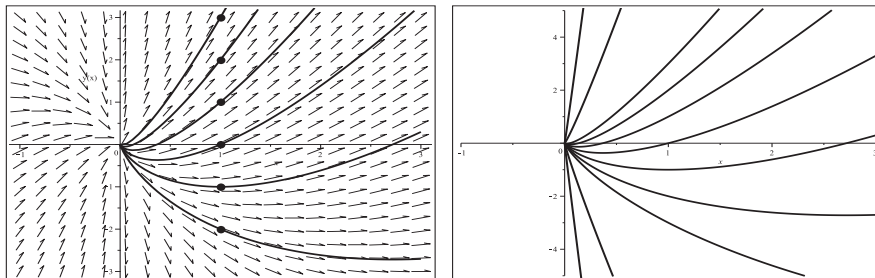


Abb. 2.198 Dateien *mehrd_Loes_08c1..2.ps*, mit/ohne Richtungsfeld zur gDGI (8),
AWP mit AB $y(0) = 0$, Lösungen $x(c + \ln(x))$, $c = -2(1)3$ (links),
sowie $x(c + \ln(x))$, $c = -30, -10, -3(1)3, 10, 30$ (rechts),
Kommandos `dsolve, plot, pointplot`

$$(9) \quad y' = y/x - y^2$$

$$y(x) = 2x/(x^2 + 2c) = 2x/(x^2 + \bar{c})$$

Untersuchungen für diese gDGL, insbesondere zur Numerik, findet der Leser ausführlicher im Bsp. 2.4 Teil (19).

Für die gDGL ist $x = 0$ eine kritische Stelle.

Unter den obigen parameterabhängigen Lösungen gibt die einfache Lösung $y(x) = 2/x$ mit einer Polstelle bei Null (das ist ein Grenzfall), aber auch welche mit zwei Polstellen symmetrisch zum Koordinatenursprung (da ist $c < 0$) und solche, die überall definiert, beliebig glatt sind und durch den Koordinatenursprung gehen (bei $c > 0$). Das ist abhängig von der AB. Des Weiteren gibt es die Nulllösung.

Damit erhält man die Mehrdeutigkeit des AWP zur AB $y(0) = 0$, wo es z.B. die Lösungen $y(x) = 0$ und $y(x) = 2x/(x^2 + 1)$ gibt.

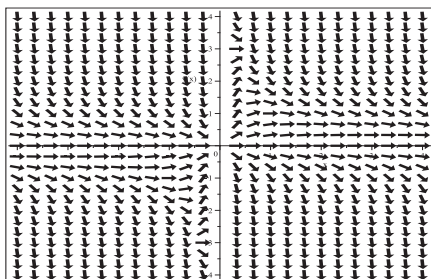


Abb. 2.199

Datei *mehrd_loes_09a.ps*,
Richtungsfeld zur gDGL (9),
Kommando `dfieldplot`

Mit dem Kommando `DEplot` erhält man ganz verschiedene Ergebnisse, die detaillierter im Bsp. 2.4 Teil (19) zu finden sind. Dabei können Näherungsverfahren nicht mit der AB $y(0) = y_0$ starten, aber durchaus über diese Stelle hinweg integrieren.

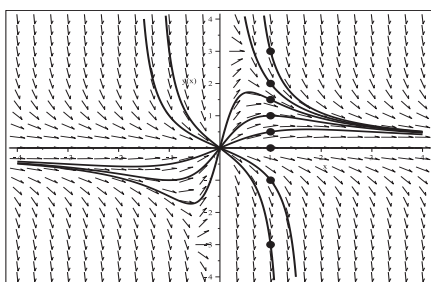


Abb. 2.200 Datei *mehrd_loes_09b1.ps*,
Richtungsfeld zur gDGL (9),
Näherungslösungen zu den AB
 $[x_0, y_0] = [1, -3], [1, -1(1)3], [1, 1/2],$
 $[1, 3/2]$ in $[-4, 4] \times [-4, 4]$,
Kommando `DEplot`,
`method=rkf45, stepsize=0.01`

Mittels `dsolve` ist die Lösung zur AB $y(x_0) = y_0$

$$y(x) = \frac{2x y_0}{y_0(x^2 - x_0^2) + 2x_0}$$

Überraschend ist, dass mit der AB $y(0) = 0$ die vier Lösungen

$$0, \quad \frac{2x}{x^2 + c}, \quad \frac{2x(x^2 \pm \sqrt{-d})}{x^4 + d}$$

erzeugt werden. Die beiden letzten korrespondieren natürlich mit der zweiten bei der Substitution $-d = c > 0$. Zu den AB $y(0) = y_0, 1, \dots$ liefert `dsolve` keine Lösung.

Wenn die AB nicht auf der Grenzkurve $2/x$ liegt, dann gilt für die Lösung $y(x)$, dass sie bzw. der innere Lösungszweig (Lösungsfortsetzung) durch den Koordinatenursprung verläuft. Für $x_0 > 0, y_0 > 2/x_0$ sind die Polstellen $\pm\sqrt{x_0^2 - 2x_0/y_0}$, sodass für die positive Polstelle $0 < \sqrt{x_0^2 - 2x_0/y_0} < x_0$ gilt.

Die Polstellen im Fall der Lösung $2x/(x^2 - c), c > 0$, liegen bei $\pm\sqrt{c}$.

Die Nulllösung allein erhält man mit der AB $y(x_0) = 0, x_0 \neq 0$.

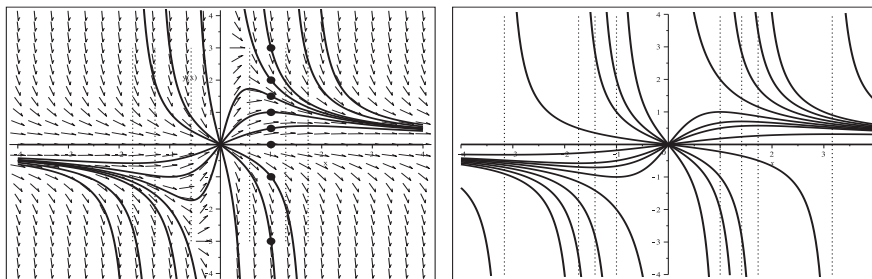


Abb. 2.201 Dateien *mehrd_Loes_09c1..2.ps*, mit/ohne Richtungsfeld zur gDGL (9), AWP mit AB $y(1) = y_0, y_0 = -3, -1, 0, \frac{1}{2}, 1, \frac{3}{2}, 2, 3$, und den Lösungen $2x/(x^2 + c)$ mit $c = -\frac{5}{2}, -3, \infty, 3, 1, \frac{1}{3}, 0, -\frac{1}{3}$ (links), sowie $2x/(x^2 + c), c = -10, -3(1)3, 10$ (rechts), dazu weitere Lösungszweige und Asymptoten an den Polstellen, Kommandos `dsolve, plot, pointplot`

$$(10) \quad y' = y^{(k-1)/k}, \quad k \in \mathbb{R} \setminus \{0\}$$

$$y(x) = \left(\frac{x + \tilde{c}}{k}\right)^k = \left(\frac{x + ck}{k}\right)^k = \left(\frac{x}{k} + c\right)^k$$

Untersuchungen zu dieser gDGL für den Fall $k = 2$ findet man im Bsp. 2.5 Teil (3).

Dazu gibt es das Richtungsfeld für $y \geq 0$. Es existiert die spezielle Lösung $y(x) = 0$.

Zum AWP mit beliebigen AB $y(x_0) = y_0$ ist die Lösung

$$y(x) = \left(\frac{x - x_0 + 2\sqrt{y_0}}{2}\right)^2, \quad x \in [x_0 - 2\sqrt{y_0}, +\infty).$$

Das ist der wachsende Lösungszweig der konvexen Parabel.

Zur AB $y(x_0) = 0$ findet man unendlich viele Lösungen. Dies bedeutet die Mehrdeutigkeit des AWP. Sei z.B. $x_0 = 0$. Dann sind in $[0, \infty)$ die Lösungen

$$y_1(x) = 0, \quad y_2(x) = \left(\frac{x}{2}\right)^2 \quad \text{und} \quad y_3(x) = \begin{cases} 0 & \text{für } x \in [0, c], \quad c > 0 \text{ beliebig,} \\ \left(\frac{x - c}{2}\right)^2 & \text{für } x \in (c, \infty), \end{cases}$$

wobei die Nulllösung natürlich für beliebiges x definiert ist.

Zum Richtungsfeld und der Lösungsmannigfaltigkeit sollen noch einige Untersuchungen angestellt werden. Daraus folgen Erkenntnisse bzgl. der Mehrdeutigkeit der Lösung bei Verwendung von `DEplot` und `dsolve`. Dazu machen wir eine Fallunterscheidung in Bezug auf den Parameter k sowie den Exponenten $\frac{k-1}{k}$.

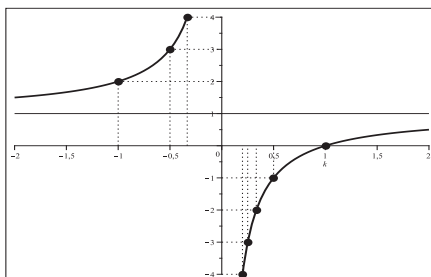


Abb. 2.202

Datei `mehrd.Joes_10.ps`,
gDGI (10),
Funktion des Exponenten
 $\frac{k-1}{k}$, $k \in [-2, 2]$, $k \neq 0$

Das Richtungsfeld ist nicht für $k = 0$ definiert und in Abhängigkeit von k wird es für alle Werte y bzw. für $y > 0$, $y \geq 0$ oder $y \neq 0$ existieren.

Die möglichen Fälle werden in einer Übersicht zusammengestellt.

k	$\frac{k-1}{k}$	Richtungsfeld
$\in (1, \infty)$	$\in (0, 1)$	für $y \geq 0$
$= 1$	$= 0$	für alle y , einfacher Fall
$\in (\frac{1}{2}, 1)$	$\in (-1, 0)$	für $y > 0$
$\in (0, \frac{1}{2}]$	$\in (-\infty, -1]$ $k' = \frac{1-k}{k} \in [1, \infty)$, k' ungerade für $k = \frac{1}{2m}$ k' gerade für $k = \frac{1}{2m+1}$ k' sonst	für $y \neq 0$ nach oben für $y > 0$ nach unten für $y < 0$ nach oben für $y \neq 0$ nach oben für $y > 0$
$= 0$	nicht zulässig	
$\in [-1, 0)$	$\in [2, \infty)$ $k = -\frac{1}{2m-1}$, $m=1, 2, \dots$ ungerader Fall bzgl. k $k = -\frac{1}{2m}$, $m = 1, 2, \dots$ gerader Fall bzgl. k k sonst	horizontal für $y = 0$ nach oben für $y \neq 0$ horizontal für $y = 0$ nach oben für $y > 0$ nach unten für $y < 0$ nach oben für $y \geq 0$
$\in (-\infty, -1)$	$\in (1, 2)$	nach oben für $y \geq 0$

Tab. 2.1 Parameter k , Exponent $\frac{k-1}{k}$ und Eigenschaften des Richtungsfeldes

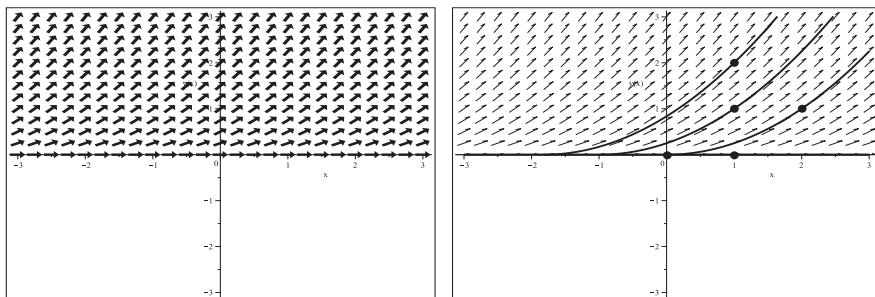


Abb. 2.203 Dateien *mehrd_Loes_10a, b01.ps*, gDGl (10), Fall $k \in (1, \infty)$, Richtungsfeld zur gDGl $k = 2$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(0) = 0, y(1) = 0, 1, 2, y(2) = 1$ (rechts), Kommandos `dfieldplot, DEplot, method=rkf45, stepsize=0.01`

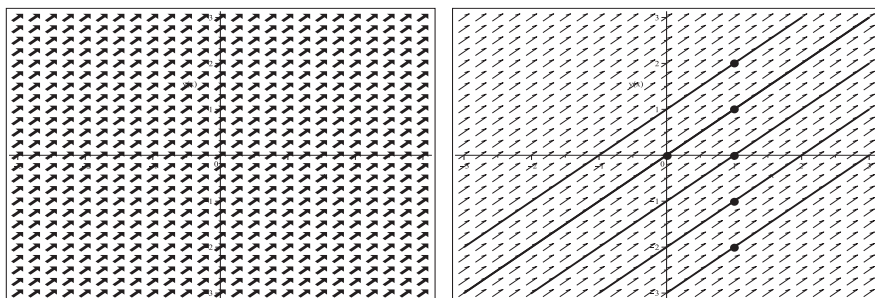


Abb. 2.204 Dateien *mehrd_Loes_10a, b02.ps*, gDGl (10), Fall $k = 1$, Richtungsfeld zur gDGl $k = 1$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(0) = 0, y(1) = -2(1)2$ (rechts), Kommandos `dfieldplot, DEplot, method=rkf45, stepsize=0.01`

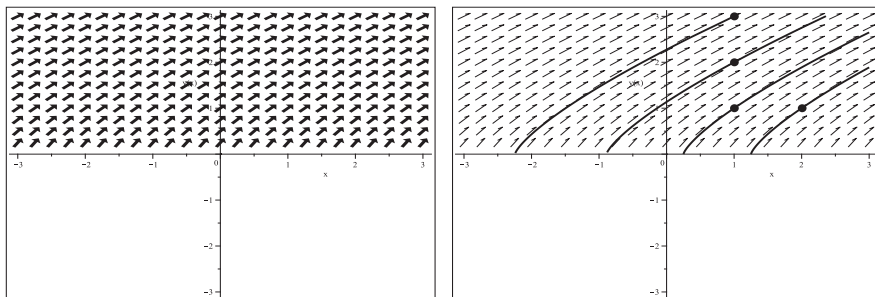


Abb. 2.205 Dateien *mehrd_Loes_10a, b03.ps*, gDGl (10), Fall $k \in (\frac{1}{2}, 1)$, Richtungsfeld zur gDGl $k = \frac{3}{4}$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(1) = 1, 2, 3, y(2) = 1$ (rechts), Kommandos `dfieldplot, DEplot, method=rkf45, stepsize=0.01`

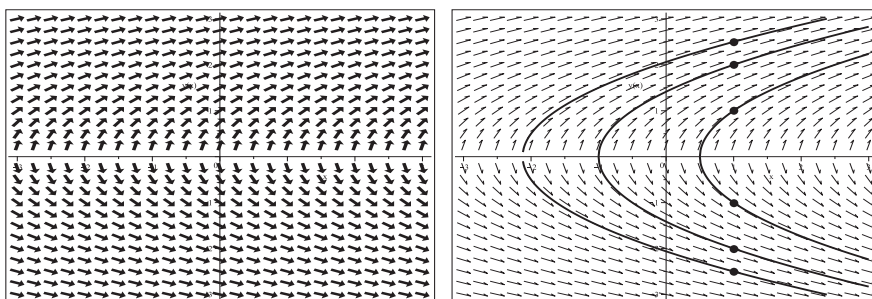


Abb. 2.206 Dateien *mehrd_loes_10a, b04.ps*, gDGl (10), Fall $k \in (0, \frac{1}{2}]$, Richtungsfeld zur gDGl $k = \frac{1}{2}$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(1) = -\frac{5}{2}, -2 - 1, 1, 2, \frac{5}{2}$ (rechts), Kommandos `dfieldplot, DEplot, method=rkf45, stepsize=0.01`

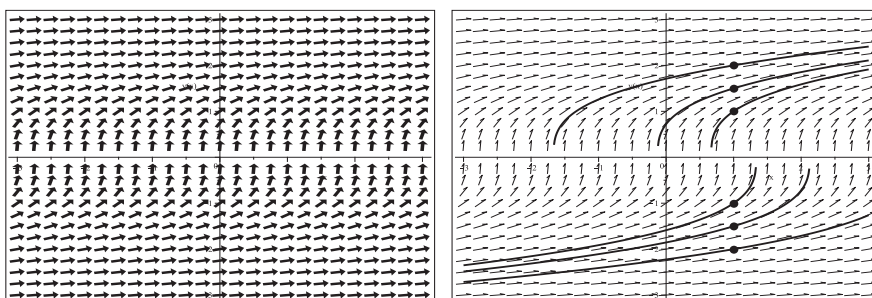


Abb. 2.207 Dateien *mehrd_loes_10a, b05.ps*, gDGl (10), Fall $k \in (0, \frac{1}{2}]$, Richtungsfeld zur gDGl $k = \frac{1}{3}$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(1) = -2, -\frac{3}{2}, -1, 1, \frac{3}{2}, 2$ (rechts), Kommandos `dfieldplot, DEplot, method=rkf45, stepsize=0.01`

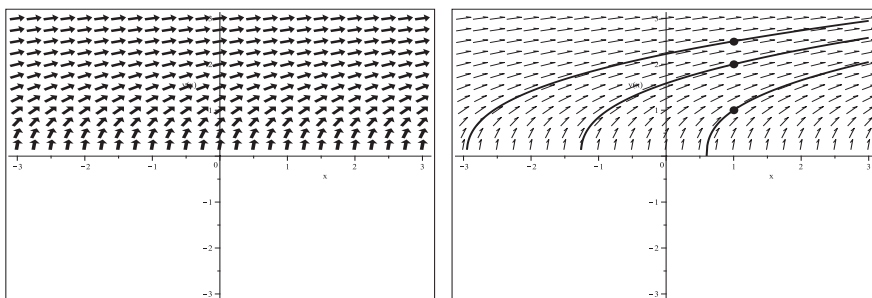


Abb. 2.208 Dateien *mehrd_loes_10a, b06.ps*, gDGl (10), Fall $k \in (0, \frac{1}{2}]$, Richtungsfeld zur gDGl $k = \frac{2}{5}$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(1) = 1, 2, \frac{5}{2}$ (rechts), Kommandos `dfieldplot, DEplot, method=rkf45, stepsize=0.01`

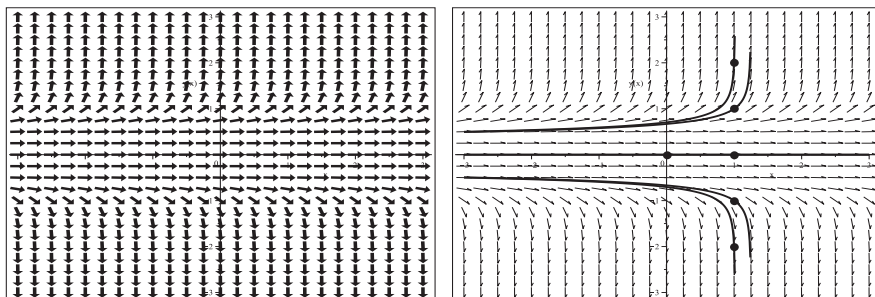


Abb. 2.209 Dateien *mehrd_Loes_10a, b07.ps*, gDGl (10), Fall $k \in [-1, 0)$, Richtungsfeld zur gDGl $k = -\frac{1}{4}$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(0) = 0$, $y(1) = -2$ (rechts), Kommandos `dfieldplot`, `DEplot, method=rkf45, stepsize=0.01`

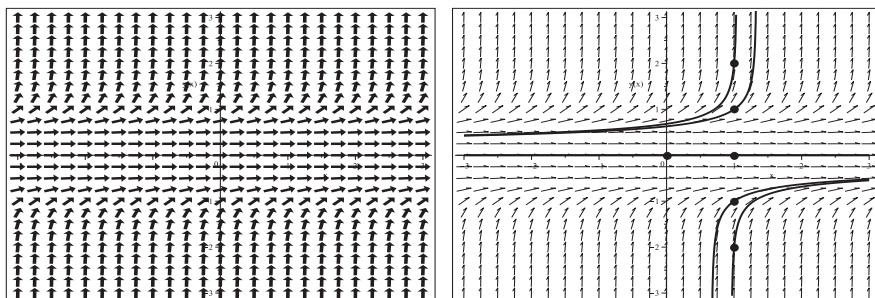


Abb. 2.210 Dateien *mehrd_Loes_10a, b08.ps*, gDGl (10), Fall $k \in [-1, 0)$, Richtungsfeld zur gDGl $k = -\frac{1}{3}$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(0) = 0$, $y(1) = -2$ (rechts), Kommandos `dfieldplot`, `DEplot, method=rkf45, stepsize=0.01`

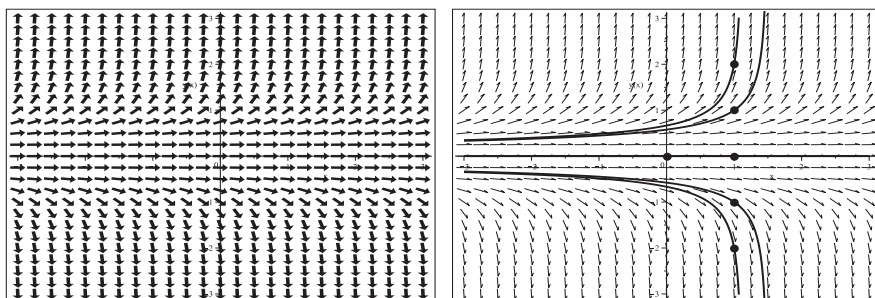


Abb. 2.211 Dateien *mehrd_Loes_10a, b09.ps*, gDGl (10), Fall $k \in [-1, 0)$, Richtungsfeld zur gDGl $k = -\frac{1}{2}$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(0) = 0$, $y(1) = -2$ (rechts), Kommandos `dfieldplot`, `DEplot, method=rkf45, stepsize=0.01`

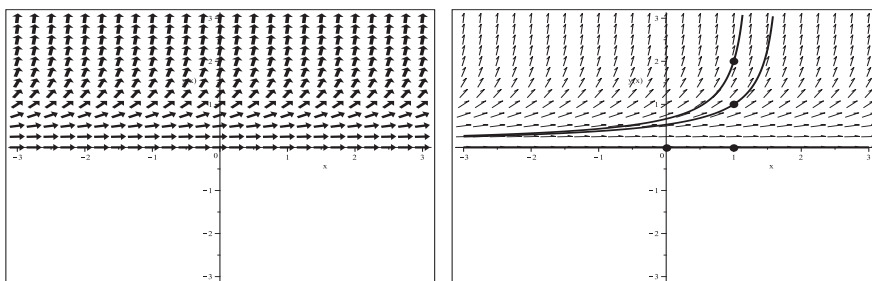


Abb. 2.212 Dateien *mehrd_loes_10a, b10.ps*, gDG1 (10), Fall $k \in [-1, 0)$, Richtungsfeld zur gDG1 $k = -\frac{3}{4}$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(0) = 0$, $y(1) = 0, 1, 2$ (rechts), Kommandos `dfieldplot`, `DEplot,method=rkf45,stepsize=0.01`

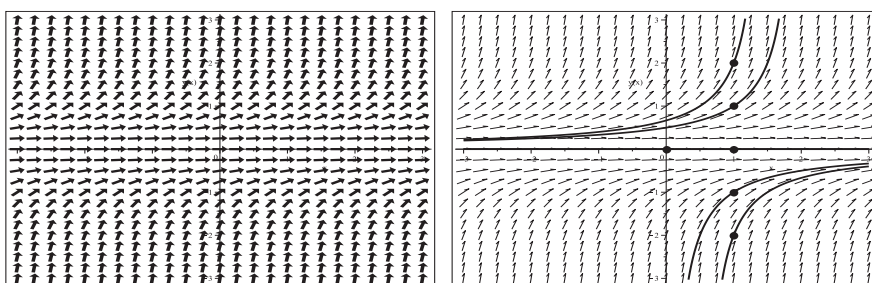


Abb. 2.213 Dateien *mehrd_loes_10a, b11.ps*, gDG1 (10), Fall $k \in [-1, 0)$, Richtungsfeld zur gDG1 $k = -1$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(0) = 0$, $y(1) = -2, 1, 2$ (rechts), Kommandos `dfieldplot`, `DEplot,method=rkf45,stepsize=0.01`

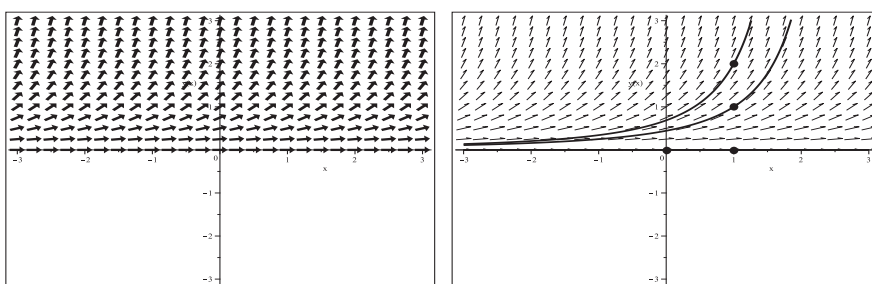


Abb. 2.214 Dateien *mehrd_loes_10a, b12.ps*, gDG1 (10), Fall $k \in (-\infty, -1)$, Richtungsfeld zur gDG1 $k = -2$ in $[-3, 3] \times [-3, 3]$ (links), und Näherungslösungen zu AB $y(0) = 0$, $y(1) = 0, 1, 2$ (rechts), Kommandos `dfieldplot`, `DEplot,method=rkf45,stepsize=0.01`

Näherungslösungen mittels der numerischen Verfahren in `DEplot` werden dort berechnet, wo das Richtungsfeld existiert.

Für die AB $y(x_0) = 0$ erhält man mit `DEplot` die Nulllösung in den Fällen $k > 1$ und $k < 0$. Eine Ausnahme ist der Sonderfall $k = 1$ mit der Lösung $y(x) = x - x_0$.

Für $y_0 \neq 0$ haben die Näherungslösungen einseitig begrenzte Definitionsbereiche, wobei im Fall $k > 0$ die Grenze durch die Positivität des Terms $\frac{x}{k} + c$ bestimmt wird, während für $k < 0$ die Grenze eine Polstelle ist.

Im Fall $k > 1$ hat das AWP mit der AB $y(x_0) = y_0 \geq 0$ die Lösung

$$y(x) = \left(\frac{x - x_0}{k} + \sqrt[k]{y_0} \right)^k, \quad x \in [\tilde{x}, \infty) = [x_0 - k\sqrt[k]{y_0}, \infty).$$

Es handelt sich hierbei um den wachsenden Zweig einer Parabel. Neben dieser Lösung gibt es die überall definierte Nulllösung für $y(x_0) = 0$.

Sei speziell $k = 2$. Zu den beiden AB $y(x_0) = y_0$, $y_0 = 1, \frac{1}{4}$ ist der Definitionsbereich der Lösung $[\tilde{x}, \infty)$, wobei

$$\tilde{x} = x_0 - 2\sqrt{y_0} = \begin{cases} x_0 - 2 & \text{für } y_0 = 1, \\ x_0 - 1 & \text{für } y_0 = \frac{1}{4}, \end{cases}$$

und der Parabelzweig

$$y(x) = \begin{cases} \left(\frac{x - x_0}{2} + 1 \right)^2, & x \in [x_0 - 2, \infty), y_0 = 1, y(x_0 - 2) = 0, \\ \left(\frac{x - x_0}{2} + \frac{1}{2} \right)^2, & x \in [x_0 - 1, \infty), y_0 = \frac{1}{4}, y(x_0 - 1) = 0, \end{cases} \quad y(x_0) = y_0.$$

In der Abbildung 2.203 ist die Mehrdeutigkeit der Lösung im Fall $k > 1$ erkennbar.

Unter Verwendung der Numerik in `DEplot` kann eine vorhandene Nulllösung berechnet werden, die entweder separiert ist und damit sich keine weitere Lösungskurve aus ihr abzweigt ($k < 0$) oder zur mehrdeutigen Situation ($k > 1$) gehört.

Aber bei der Anwendung von `dsolve` kommen neue Aspekte hinzu.

Das Kommando `k:=k':dsolve(dgl,y(x0)=y0,y(x))` liefert für das allgemeine AWP

$$y(x) = \left(\frac{x + k e^{\ln(y_0)/k} - x_0}{k} \right)^k = \left(\frac{x + k y_0^{1/k} - x_0}{k} \right)^k = \left(\frac{x - x_0}{k} + \sqrt[k]{y_0} \right)^k,$$

bei $y(x_0) = 0$ generell die Nulllösung, egal ob x_0 allgemein oder dafür ein konkreter Wert steht.

Aber bei konkretem k taucht die Mehrdeutigkeit der Lösung in Situationen auf, die bei `DEplot` eben nicht eintreten, und man zeigt diese mit einem kleinen Trick.

Fallunterscheidung bezüglich des Parameters k

$k = 2 \in (1, \infty)$, Mehrdeutigkeit der Lösung

```
> k:=k':
  dgl:=diff(y(x),x)-y(x)^((k-1)/k):
> k:=2;
  dsol:=dsolve({dgl,y(x0)=y0},y(x)); # y(x)=((x-x0)/2+sqrt(y0))^2
```

```

> # 2 Loesungen zur AB y(0)=0
subs(x0=0,y0=0,rhs(dsol));          # y(x)= x^2/4
dsolve({dgl,y(0)=0},y(x));          # y(x)=0
# 2 Loesungen zur AB y(1)=0
factor(subs(x0=1,y0=0,rhs(dsol)));  # y(x)=(x-1)^2/4
dsolve({dgl,y(1)=0},y(x));          # y(x)=0

```

$k = 1$, Eindeutigkeit der Lösung

```

> k:=1;
dsolve({dgl,y(x0)=y0},y(x));        # y(x)=x-x0+y0
dsolve({dgl,y(0)=0},y(x));          # y(x)=x

```

$k = \frac{3}{4} \in (\frac{1}{2}, 1)$, Mehrdeutigkeit der Lösung für $y_0 = 0$, obwohl für diesen Parameter k das Richtungsfeld nur für $y > 0$ erzeugt wird und dort mit DEplot eindeutig Näherungen berechnet werden.

```

> k:=3/4;
> dsol:=dsolve({dgl,y(x0)=y0},y(x));
allvalues(%);
# 2 reelle Loesungen zur AB y(0)=0, dazu noch 2 konjugiert komplexe
# fuer andere Werte x0 aehnlich
subs(x0=0,y0=0,rhs(dsol));
allvalues(%);
> # analog
dsol:=dsolve({dgl,y(x0)=0},y(x));
f10:=subs(x0=0,rhs(dsol[1])); f20:=subs(x0=0,rhs(dsol[2]));

```

$$dsol := y(x) = \text{RootOf}(-3_Z^{4/3} + 4x + 3y0^{4/3} - 4x0)$$

$$y(x) = \frac{1}{9}\sqrt{3}(12x + 9y0^{4/3} - 12x0)^{3/4}, \quad y(x) = -\frac{1}{9}\sqrt{3}(12x + 9y0^{4/3} - 12x0)^{3/4},$$

$$y(x) = -\frac{1}{9}I\sqrt{3}(12x + 9y0^{4/3} - 12x0)^{3/4}, \quad y(x) = \frac{1}{9}I\sqrt{3}(12x + 9y0^{4/3} - 12x0)^{3/4}$$

$$\text{RootOf}(-3_Z^{4/3} + 4x)$$

$$\frac{2}{9}\sqrt{6}(\sqrt{3}\sqrt{x})^{3/2}, -\frac{2}{9}\sqrt{6}(\sqrt{3}\sqrt{x})^{3/2}, -\frac{2}{9}I\sqrt{6}(\sqrt{3}\sqrt{x})^{3/2}, \frac{2}{9}I\sqrt{6}(\sqrt{3}\sqrt{x})^{3/2}$$

$$dsol := y(x) = -\frac{2}{9}\sqrt{6}(3x - 3x0)^{3/4}, \quad y(x) = \frac{2}{9}\sqrt{6}(3x - 3x0)^{3/4},$$

$$y(x) = -\frac{2}{9}I\sqrt{6}(3x - 3x0)^{3/4}, \quad y(x) = \frac{2}{9}I\sqrt{6}(3x - 3x0)^{3/4}$$

$$f10 = -\frac{2}{9}\sqrt{6} 3^{3/4} x^{3/4}$$

$$f20 = \frac{2}{9}\sqrt{6} 3^{3/4} x^{3/4}$$

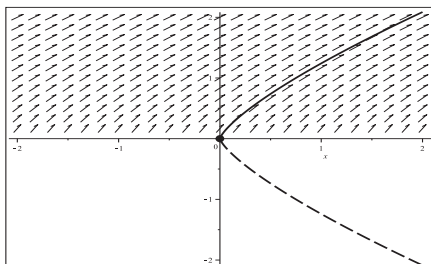


Abb. 2.215

Datei *mehrdLoes_10c01.ps*,

Richtungsfeld zur gDGL (10), $k = \frac{3}{4}$, $y > 0$,
zur AB $y(0) = 0$ zwei Lösungen in $[-2, 2]$

$$y_1(x) = \frac{2}{9}\sqrt{6}(3x)^{3/4}, \quad y_2(x) = -\frac{2}{9}\sqrt{6}(3x)^{3/4},$$

Kommandos *dsolve*, *plot*, *pointplot*,

dfieldplot für Richtungsfeld

$k = \frac{1}{2} \in (0, \frac{1}{2}]$, Mehrdeutigkeit der Lösung für $y_0 = 0$, obwohl für diesen Parameter k das Richtungsfeld nur für $y \neq 0$ erzeugt wird und dort mit DEplot eindeutig Näherungen berechnet werden.

```
> k:=1/2:
> dsolve({dgl,y(x0)=y0},y(x)); # kein Ergebnis
dsol:=dsolve({dgl,y(x0)=y0},y(x),implicit);
# explizite Loesungsdarstellung, 2 reelle Loesungen
dsol12:=solve(dsol,y(x));
> # 2 reelle Loesungen zur AB y(0)=0, fuer andere Werte x0 aehnlich
subs(x0=0,y0=0,dsol);
solve(%,y(x));
> # analog
dsol:=dsolve({dgl,y(x0)=0},y(x));
f10:=subs(x0=0,rhs(dsol[1])); f20:=subs(x0=0,rhs(dsol[2]));
> dsolve({dgl,y(0)=0},y(x));
```

$$\begin{aligned}
 dsol &:= y(x)^2 - 2x - y0^2 + 2x0 = 0 \\
 dsol12 &:= \sqrt{2x + y0^2 - 2x0}, -\sqrt{2x + y0^2 - 2x0} \\
 dsol &:= y(x)^2 - 2x = 0 \\
 &\quad \sqrt{2x}, -\sqrt{2x} \\
 y(x) &= \sqrt{2x - 2x0}, y(x) = -\sqrt{2x - 2x0} \\
 f10 &:= \sqrt{2x} \\
 f20 &:= -\sqrt{2x} \\
 y(x) &= \sqrt{2x}, y(x) = -\sqrt{2x}
 \end{aligned}$$

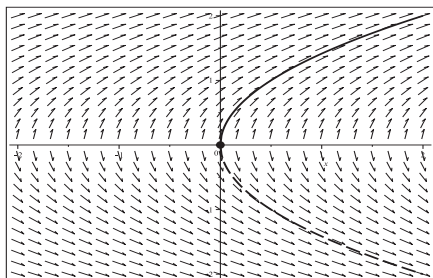


Abb. 2.216

Datei *mehrdLoes_10c02.ps*,
 Richtungsfeld zur gDGl (10), $k = \frac{1}{2}$, $y \neq 0$,
 zur AB $y(0) = 0$ zwei Lösungen in $[-2, 2]$
 $y_1(x) = \sqrt{2x}$, $y_2(x) = -\sqrt{2x}$,
 Kommandos `dsolve`, `plot`, `pointplot`,
`dfieldplot` für Richtungsfeld

$k = \frac{1}{3} \in (0, \frac{1}{2}]$, Mehrdeutigkeit der Lösung für $y_0 = 0$, obwohl für diesen Parameter k das Richtungsfeld nur für $y \neq 0$ erzeugt wird und dort mit DEplot eindeutig Näherungen berechnet werden.

```
> k:=1/3:
> dsolve({dgl,y(x0)=y0},y(x)); # kein Ergebnis
dsol:=dsolve({dgl,y(x0)=y0},y(x),implicit);
# 1 reelle Loesung zur AB y(x0)=y0, dazu noch 2 konjugiert komplexe
dsol123:=solve(dsol,y(x));
```

```

> # 1 reelle Loesung und 2 konjugiert komplexe zur AB y(0)=0
  subs(x0=0,y0=0,dsol);
  solve(%,y(x));
> # analog
  dsol:=dsolve({dgl,y(x0)=0},y(x));
  f10:=subs(x0=0,rhs(dsol[1]));
> dsolve({dgl,y(0)=0},y(x));

```

$$dsol := y(x)^3 - 3x - y0^3 + 3x0 = 0$$

$$dsol123 :=$$

$$(3x + y0^3 - 3x0)^{1/3}, -\frac{1}{2}(1 - I\sqrt{3})(3x + y0^3 - 3x0)^{1/3}, -\frac{1}{2}(1 + I\sqrt{3})(3x + y0^3 - 3x0)^{1/3}$$

$$dsol := y(x)^3 - 3x = 0$$

$$(3x)^{1/3}, -\frac{1}{2}(1 - I\sqrt{3})(3x)^{1/3}, -\frac{1}{2}(1 + I\sqrt{3})(3x)^{1/3}$$

$$dsol := y(x) = (3x-3x0)^{1/3}, y(x) = -\frac{1}{2}(1-I\sqrt{3})(3x-3x0)^{1/3}, y(x) = -\frac{1}{2}(1+I\sqrt{3})(3x-3x0)^{1/3}$$

$$f10 := (3x)^{1/3}$$

$$y(x) = (3x)^{1/3}, y(x) = -\frac{1}{2}(1 - I\sqrt{3})(3x)^{1/3}, y(x) = -\frac{1}{2}(1 + I\sqrt{3})(3x)^{1/3}$$

Die zweite reelle Lösung $y(x) < 0$, die ja existiert, kann mittels Maple nur nach der Substitution $y = -z > 0$ erzeugt werden. Man erhält dann den negativen Zweig

$$y(x) = -\sqrt[3]{3(x_0 - x) - y_0^3}, y(x_0) = y_0 < 0, x \in (-\infty, \tilde{x}] = (-\infty, x_0 - y_0^3/3],$$

speziell $y(x) = -\sqrt[3]{-3x}$, $x \leq 0$, zur AB $y(0) = 0$.

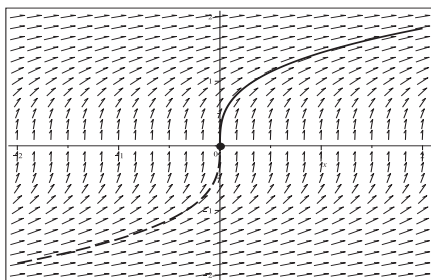


Abb. 2.217

Datei *mehrd_Loes_10c03.ps*,

Richtungsfeld zur gDGL (10), $k = \frac{1}{3}$, $y \neq 0$,
zur AB $y(0) = 0$ zwei Lösungen in $[-2, 2]$

$$y_1(x) = \sqrt[3]{3x}, y_2(x) = -\sqrt[3]{-3x},$$

Kommandos `dsolve`, `plot`, `pointplot`,
`dfieldplot` für Richtungsfeld

$k = \frac{2}{5} \in (0, \frac{1}{2}]$, Eindeutigkeit der Lösung für $y_0 = 0$, wobei für diesen Parameter k das Richtungsfeld nur für $y > 0$ erzeugt wird und dort mit `DEplot` eindeutig Näherungen berechnet werden.

```

> k:=2/5:
> dsolve({dgl,y(x0)=y0},y(x));
  dsol:=dsolve({dgl,y(x0)=y0},y(x),implicit);
  # 1 reelle Loesung zur AB y(x0)=y0, dazu 2 Paare konjugiert komplexer
  dsol12345:=solve(dsol,y(x));
> # 1 reelle Loesung und 2 Paare konjugiert komplexer zur AB y(0)=0

```

```

subs(x0=0,y0=0,dsol);
solve(%,y(x));
> # analog
dsol:=dsolve({dgl,y(x0)=0},y(x));
f10:=subs(x0=0,rhs(dsol[1]));
dsolve({dgl,y(0)=0},y(x)):

```

$$dsol := y(x) = \text{RootOf}(-2 _Z^{5/2} + 5x + 2y0^{5/2} - 5x0)$$

$$dsol := y(x)^{5/2} - \frac{5}{2}x - y0^{5/2} + \frac{5}{2}x0 = 0$$

$$dsol12345 := \frac{1}{4}(80x + 32y0^{5/2} - 80x0)^{2/5}, d_1(80x + 32y0^{5/2} - 80x0)^{2/5}, \\ d_2(80x + 32y0^{5/2} - 80x0)^{2/5}, d_3(80x + 32y0^{5/2} - 80x0)^{2/5}, d_4(80x + 32y0^{5/2} - 80x0)^{2/5}$$

$$dsol := y(x)^{5/2} - \frac{5}{2}x = 0$$

$$\frac{1}{2}5^{2/5}2^{3/5}x^{2/5}, 2d_15^{2/5}2^{3/5}x^{2/5}, 2d_25^{2/5}2^{3/5}x^{2/5}, 2d_35^{2/5}2^{3/5}x^{2/5}, 2d_45^{2/5}2^{3/5}x^{2/5}$$

$$dsol := y(x) = \frac{1}{4}(80x - 80x0)^{2/5}, y(x) = d_1(80x - 80x0)^{2/5},$$

$$y(x) = d_2(80x - 80x0)^{2/5}, y(x) = d_3(80x - 80x0)^{2/5}, y(x) = d_4(80x - 80x0)^{2/5}$$

$$f10 := \frac{1}{4}80^{2/5}x^{2/5}$$

wobei

$$d_1 = \frac{1}{4} \left(-\frac{1}{4} + \frac{1}{4}\sqrt{5} + \frac{1}{4}I\sqrt{2}\sqrt{5+\sqrt{5}} \right)^2, \quad d_2 = \frac{1}{4} \left(-\frac{1}{4} - \frac{1}{4}\sqrt{5} + \frac{1}{4}I\sqrt{2}\sqrt{5-\sqrt{5}} \right)^2,$$

$$d_3 = \frac{1}{4} \left(-\frac{1}{4} - \frac{1}{4}\sqrt{5} - \frac{1}{4}I\sqrt{2}\sqrt{5-\sqrt{5}} \right)^2, \quad d_4 = \frac{1}{4} \left(-\frac{1}{4} + \frac{1}{4}\sqrt{5} - \frac{1}{4}I\sqrt{2}\sqrt{5+\sqrt{5}} \right)^2.$$

Die eindeutige reelle Lösung $y(x) \geq 0$ mit der AB $y(x_0) = 0$ ist also

$$y(x) = \frac{1}{4}(80x - 80x_0)^{2/5} = \frac{1}{4}80^{2/5}(x - x_0)^{2/5} = \left(\frac{5}{2}\right)^{2/5}(x - x_0)^{2/5}.$$

Ein zweite reelle Lösung $y(x) \leq 0$ mit derselben AB existiert nicht. Betrachtet man die verdächtigen Funktionen

$$g_1(x) = -\left(\frac{5}{2}\right)^{2/5}(x_0 - x)^{2/5}, \quad x \in (-\infty, x_0] \quad \text{bzw.} \quad g_2(x) = -\left(\frac{5}{2}\right)^{2/5}(x - x_0)^{2/5}, \quad x \in [x_0, \infty),$$

so sind diese zwar eine "schöne" Fortsetzung des positiven Lösungszweigs nach unten, aber keine davon erfüllt die gDGL.

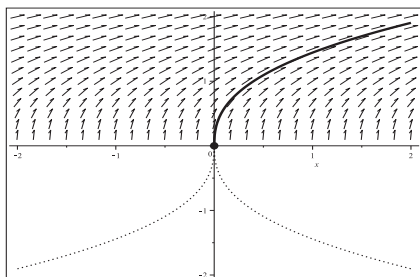


Abb. 2.218

Datei *mehrdLoes_10c04.ps*,

Richtungsfeld zur gDGL (10), $k = \frac{2}{5}$, $y > 0$,

zur AB $y(0) = 0$ die Lösung $y(x) = \left(\frac{5}{2}\right)^{2/5}x^{2/5}$,

$x \in [-2, 2]$, dazu die Funktionen $g_1(x)$ und $g_2(x)$,

Kommandos `dsolve`, `plot`, `pointplot`,

`dfieldplot` für Richtungsfeld

$k \in (-\infty, 0)$

Zu diesen Parametern k wird das Richtungsfeld entweder überall oder nur für $y \geq 0$ erzeugt. Die exakten bzw. numerischen Lösungen der entsprechenden AWP sind eindeutig. Mehrdeutigkeit tritt also nicht auf.

Bzgl. der Darstellung der Lösung sind jedoch Unterschiede zu erkennen.

In der allgemeinen Version oder bei beliebigen AB $y(x_0) = y_0$ mit den Formeln

$$y(x) = \left(\frac{x}{k} + c\right)^k = \frac{1}{(x/k + c)^{-k}}, \quad y(x) = \left(\frac{x - x_0}{k} + \sqrt[k]{y_0}\right)^k$$

fehlt der Fall der Nulllösung, den man immer mit der AB $y(x_0) = 0$ erhalten würde.

In der ersten Darstellung könnte man das mit dem Wert $c = \infty$ deuten.

Auch die zweite Darstellung lässt sich geeignet umformen, um die Nulllösung sichtbar zu machen.

$$\left(\frac{x - x_0}{k} + y_0^{1/k}\right)^k = \frac{y_0}{\left(\frac{x - x_0}{k} + y_0^{1/k}\right)^{-k} y_0} = \frac{y_0}{\left(-\frac{1}{k}(x_0 - x) y_0^{-1/k} + 1\right)^{-k}}$$

Die letzte Gestalt ergibt sich auch, sobald $k < 0$ einen konkreten Wert annimmt.

Als weiteres Ergebnis findet man die Polstelle zur Lösung aus der Nullstelle des Nennerters

$$\tilde{x} = x_0 + \frac{-k}{y_0^{-1/k}}, \quad y_0 \neq 0.$$

Die Lösung ist also ein Lösungsweig, der entweder für $x \in (-\infty, \tilde{x})$ oder $x \in (\tilde{x}, \infty)$ definiert ist. Dazu folgt eine Übersicht mit ausgewählten Parametern $k < 0$.

k	$y(x)$	\tilde{x}	Definitionsbereich
$k < 0$	$y(x) = 0$ für AB $y(x_0) = y_0 = 0$		$(-\infty, +\infty)$
$-\frac{1}{4}$	$\frac{y_0}{(4(x_0 - x)y_0^4 + 1)^{1/4}}$	$x_0 + \frac{1}{4y_0^4}$	$(-\infty, \tilde{x}), y_0 \neq 0$
$-\frac{1}{3}$	$y_0(-1 + \text{I}\sqrt{3}) \frac{[-(3(x - x_0)y_0^3 - 1)^2]^{1/3}}{2(3(x - x_0)y_0^3 - 1)}$	$x_0 + \frac{1}{3y_0^3}$	$(-\infty, \tilde{x})$ für $y_0 > 0$ $(\tilde{x}, +\infty)$ für $y_0 < 0$
$-\frac{1}{2}$	$\frac{y_0}{(2(x_0 - x)y_0^2 + 1)^{1/2}}$	$x_0 + \frac{1}{2y_0^2}$	$(-\infty, \tilde{x}), y_0 \neq 0$
$-\frac{3}{4}$	$\frac{1}{z^2} \sqrt{-z} \sqrt{-3z} y_0^{4/3} \sqrt{-3z} y_0^{4/3} = \frac{3^{3/4} y_0}{\tilde{z}^{3/4}},$ wobei $z = 4(x - x_0)y_0^{4/3} - 3 = -\tilde{z} < 0$	$x_0 + \frac{3}{4y_0^{4/3}}$	$(-\infty, \tilde{x}), y, y_0 > 0$

$k < 0$	$y(x)$	\tilde{x}	Definitionsbereich
-1	$\frac{y_0}{(x_0 - x)y_0 + 1}$	$x_0 + \frac{1}{y_0}$	$(-\infty, \tilde{x})$ für $y_0 > 0$ $(\tilde{x}, +\infty)$ für $y_0 < 0$
-2	$\frac{y_0}{(\frac{1}{2}(x - x_0)y_0^{1/2} - 1)^2}$	$x_0 + \frac{2}{y_0^{1/2}}$	$(-\infty, \tilde{x}), y, y_0 > 0$

Tab. 2.2 Parameter $k < 0$, Lösung $y(x)$ mit evtl. Polstelle \tilde{x} und Definitionsbereich

Löst man zum Parameter $k = -\frac{1}{4}$ das allgemeine AWP implizit, so folgt

$$\frac{1}{y(x)^4} + 4x - \frac{1}{y_0^4} - 4x_0 = 0,$$

woraus sich vier Wurzeln ergeben, von denen eine die gesuchte Lösung darstellt. Einen zweiten Funktionszweig dazu auf der anderen Seite der Polstelle gibt es nicht.

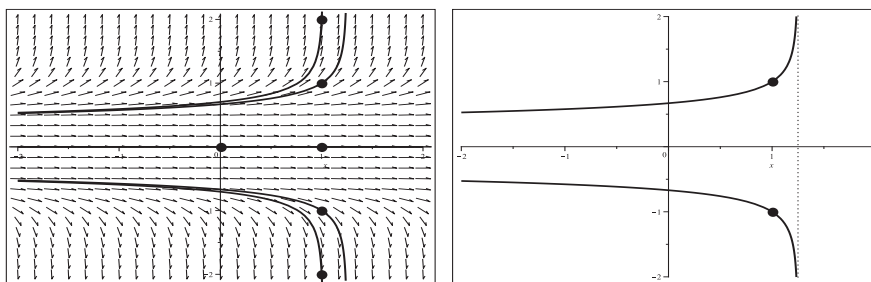


Abb. 2.219 Dateien *mehrd_loes_10c05_05a.ps*, gDGl (10), Fall $k = -\frac{1}{4}$, links Richtungsfeld zur gDGl in $[-2, 2] \times [-2, 2]$ und den Lösungen $y(x) = \frac{y_0}{\sqrt[4]{4(x_0 - x)y_0^4 + 1}}$ zu den AB $y(0) = 0$, $y(1) = -2(1)2$, rechts Lösungen $y(x) = \pm \frac{1}{\sqrt[4]{5 - 4x}}$, AB $y(1) = \pm 1$, Asymptote bei $\tilde{x} = \frac{5}{4}$, Kommandos `dfieldplot`, `plot`, `pointplot`, `dsolve`, `solve`

Für den Parameter $k = -\frac{1}{3}$ erhält man zwar eine komplexe Darstellung, aber die Berechnung ergibt reelle Werte. Das folgt daraus, dass Maple bei Berechnung der Wurzel $\sqrt[3]{-1}$ nicht -1 , sondern die komplexe Zahl $\frac{1}{2}(1 + i\sqrt{3})$ liefert.

Damit wird bei einer grafischen Ausgabe der eigentlichen Lösung auf der anderen Seite der Polstelle \tilde{x} eine zweite Kurve gezeichnet.

Die Lösung des AWP mit `dsol_impl:=dsolve(dgl,y(x0)=y0,y(x),implicit)` ergibt bei $k = -\frac{1}{3}$ die Gestalt

$$dsol_impl := \frac{1}{y(x)^3} + 3x - \frac{1}{y_0^3} - 3x_0 = 0.$$

Mittels `solve(dsol_impl,y(x))` folgen die drei Wurzeln

$$y_0 z, y_0 \left(-\frac{1}{2} - \frac{1}{2}I\sqrt{3}\right) z, y_0 \left(-\frac{1}{2} + \frac{1}{2}I\sqrt{3}\right) z, \text{ wobei } z = \frac{[-(3(x-x_0)y_0^3 - 1)]^{2/3}}{3(x-x_0)y_0^3 - 1}.$$

Es ist der dritte Term, welcher die gesuchte Lösung beinhaltet, denn die beiden anderen liefern in der Tat komplexe Werte.

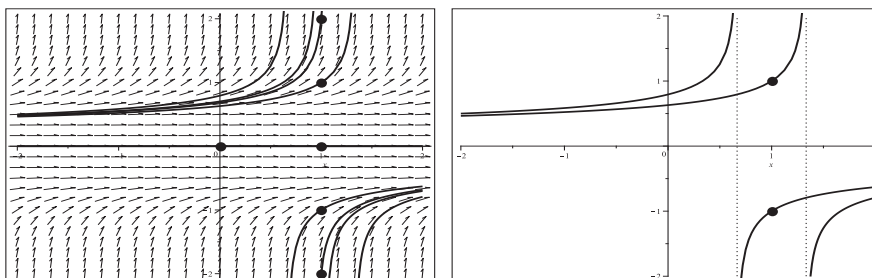


Abb. 2.220 Dateien *mehrd_loes_10c06_06a.ps*, gDGI (10), Fall $k = -\frac{1}{3}$, links Richtungsfeld zur gDGI in $[-2, 2] \times [-2, 2]$ mit Lösungen $y(x)$ zu den AB $y(0) = 0$, $y(1) = -2(1)2$ und evtl. Zweigen, rechts Lösung $y(x) = \left(-\frac{1}{2} + \frac{1}{2}I\sqrt{3}\right) \frac{(-3x-2)^{1/3}}{3x-2}$, $y(1) = 1$, $x \in (-\infty, \tilde{x})$, mit Asymptote bei $\tilde{x} = \frac{4}{3}$, Zusatzzweig rechts, und $y(x) = \left(-\frac{1}{2} + \frac{1}{2}I\sqrt{3}\right) \frac{(-3x-2)^{1/3}}{3x-2}$, $y(1) = -1$, $x \in (\tilde{x}, \infty)$, mit Asymptote bei $\tilde{x} = \frac{2}{3}$, Zusatzzweig links, Kommandos `dfieldplot,plot,pointplot, dsolve,solve`

Die Lösung des AWP zum Parameter $k = -\frac{3}{4}$ mit `dsol_impl:=dsolve(dgl,y(x0)=y0,y(x),implicit)` ergibt die Gestalt

$$dsol_impl := \frac{1}{y(x)^{4/3}} + \frac{4}{3}x - \frac{1}{y_0^{4/3}} - \frac{4}{3}x_0 = 0,$$

aus der mittels `solve(dsol_impl,y(x))` die vier Wurzeln

$$\pm u, \pm u I, \text{ wobei } u = \frac{3^{3/4} \left(-y_0^{4/3} z^3\right)^{3/4}}{z^3}, z = 4(x-x_0)y_0^{4/3} - 3,$$

folgen. Unter diesen ist die gesuchte Lösung $y(x) = -u = \frac{3^{3/4}y_0}{z^{3/4}}$, $\tilde{z} = -z > 0$.

In Maple ist die Vereinfachung der Lösungen auf diese kurze Form äußerst umständlich.

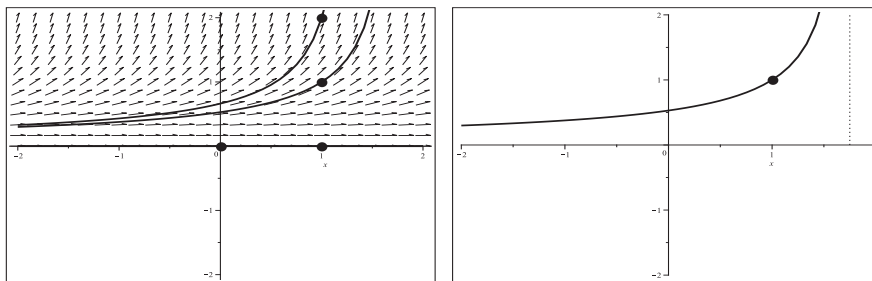


Abb. 2.221 Dateien *mehrd_Loes_10c07, 07a.ps*, gDGl (10), Fall $k = -\frac{3}{4}$,
 links Richtungsfeld zur gDGl in $[-2, 2] \times [-2, 2]$ mit
 Lösungen $y(x)$ zu den AB $y(0) = 0$, $y(1) = 0(1)2$,
 rechts Lösung $y(x) = \frac{3^{3/4}}{(7-4x)^{3/4}}$, $y(1) = 1$, $x \in (-\infty, \tilde{x})$,
 mit Asymptote bei $\tilde{x} = \frac{7}{4}$,
 Kommandos `dfieldplot`, `plot`, `pointplot`, `dsolve`, `solve`

Die Lösung des AWP zum Parameter $k = -1$ mit `dsol:=dsolve(dgl,y(x0)=y0,y(x))`
 ergibt die einfache Form

$$dsol := y(x) = \frac{y_0}{1 - y_0(x - x_0)}.$$

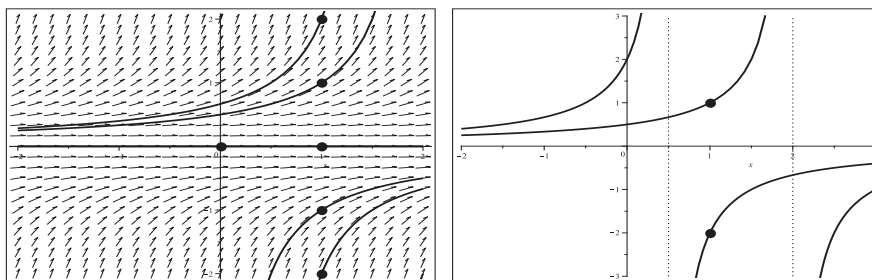


Abb. 2.222 Dateien *mehrd_Loes_10c08, 08a.ps*, gDGl (10), Fall $k = -1$,
 links Richtungsfeld zur gDGl in $[-2, 2] \times [-2, 2]$ mit
 Lösungen $y(x)$ zu den AB $y(0) = 0$, $y(1) = -2(1)2$,
 rechts Lösungen $y(x) = \frac{1}{2-x}$, $y(1) = 1$, $x < 2$, Asymptote bei $\tilde{x} = 2$,
 und $y(x) = -\frac{2}{2x-1}$, $y(1) = -2$, $x > \frac{1}{2}$, Asymptote bei $\tilde{x} = \frac{1}{2}$,
 dazu die jeweils anderen Funktionszweige,
 Kommandos `dfieldplot`, `plot`, `pointplot`, `dsolve`

Für den Parameter $k = -2$ ist die Lösung des AWP $y(x) \geq 0$ (bei der impliziten Lösung
 wird diese Bedingung sichtbar) und lautet

$$y(x) = \frac{4y_0}{(\sqrt{y_0}(x - x_0) - 2)^2}.$$

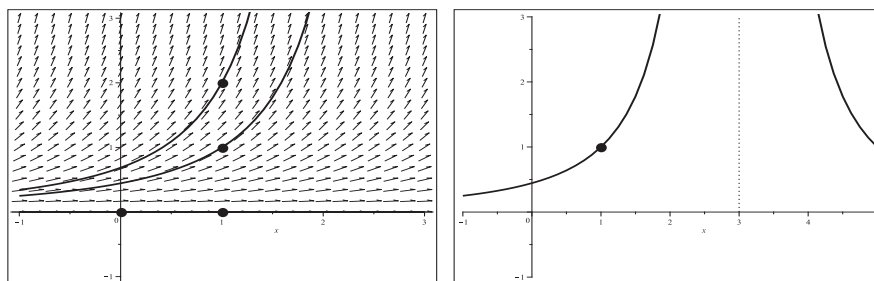


Abb. 2.223 Dateien *mehrd_loes_10c09, 09a.ps*, gDGI (10), Fall $k = -2$, links Richtungsfeld zur gDGI in $[-1, 3] \times [-1, 3]$ mit Lösungen $y(x)$ zu den AB $y(0) = 0$, $y(1) = 0(1)2$, rechts Lösung $y(x) = \frac{4}{(x-3)^2}$, $y(1) = 1$, $x \in (-\infty, \tilde{x})$, mit Asymptote bei $\tilde{x} = 3$, dazu der andere Funktionszweig, Kommandos `dfieldplot, plot, pointplot, dsolve`

Betrachtet man die allgemeine Lösung der gDGI (10) $y(x) = \left(\frac{x+ck}{k}\right)^k$ als Funktion $f(x, c, k)$, so werden bei deren Verwendung einige Lösungskurven nicht grafisch darstellbar sein. Insbesondere können Lösungszweige $y(x) < 0$ fehlen, z.B. $y(x) = -\sqrt{2x}$ mit der AB $y(2) = -2$ bei $k = \frac{1}{2}$.

Die Funktion $f(x, 0, k)$ wird für ausgewählte Werte $k \neq 0$ geplottet, dazu die Nulllösung. Mit der Mehrdeutigkeit der Lösung hat diese Betrachtung natürlich nichts zu tun.

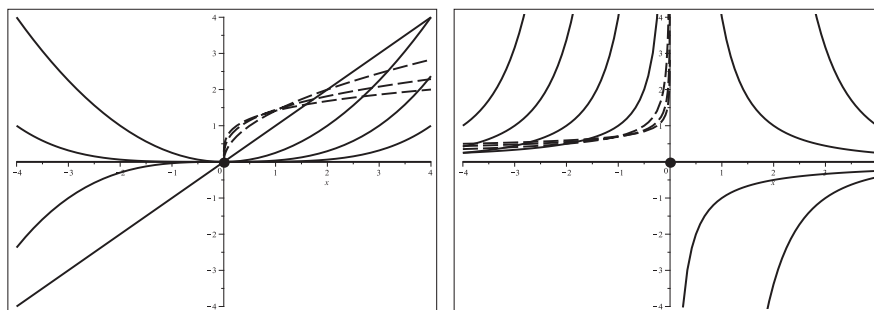


Abb. 2.224 Dateien *mehrd_loes_10d1, 2.ps*, gDGI (10), Funktionen $f(x, 0, k) = \left(\frac{x}{k}\right)^k$, links $f(x, 0, k)$, $k = 1(1)4$, $k = \frac{1}{m}$, $m = 2, 3, 4$, $x \in [-4, 4]$, rechts $f(x, 0, k)$, $k = -4(1) - 1$, $k = -\frac{1}{m}$, $m = 2, 3, 4$, dazu die Funktion $y(x) = 0$ (Nulllösung), Kommandos `plot, pointplot`

Auf ähnliche Weise kann man die gDGI $y' = ay^{(k-1)/k}$, $k \in \mathbb{R} \setminus \{0\}$, mit einem weiteren reellen Parameter $a \neq 0$ untersuchen.

$$(11) \quad y^2(1 + y'^2) - 1 = 0$$

$$y(x) = \pm 1, \pm \sqrt{2xc - c^2 - x^2 + 1} = \pm \sqrt{1 - (x - c)^2}$$

Die Nichtlinearität dieser gDGL führt zunächst auf vier Teillösungen, jeweils paarweise vorzeichenverschieden. Dabei sind die beiden stationären Lösungen. Die anderen haben wegen der Wurzel einen endlichen Definitionsbereich. Die Lösungsmannigfaltigkeit ist zugleich ein Hinweis auf die Mehrdeutigkeit der Lösung.

Das alles wird sich insbesondere auf das Richtungsfeld sowie die numerische Lösung auswirken. Die Nichtlinearität verhindert die Erzeugung eines Richtungsfeldes mittels `dfieldplot`.

Error, (in DEtools/dfieldplot) unable to isolate DEs for required derivatives

Erst in der expliziten Darstellung der gDGL mit der Fallunterscheidung

$$y' = \sqrt{\frac{1}{y^2} - 1}, \quad y' = -\sqrt{\frac{1}{y^2} - 1}, \quad y \neq 0,$$

erzeugt man zwei zueinander "symmetrische Richtungsfelder" im Bereich $|y| \leq 1$, in denen jedoch jeweils die Hälfte der Richtungselemente fehlt. Eigentlich sind diese die Tangenten an den Kreis $y^2 = 1 - (x - c)^2$.

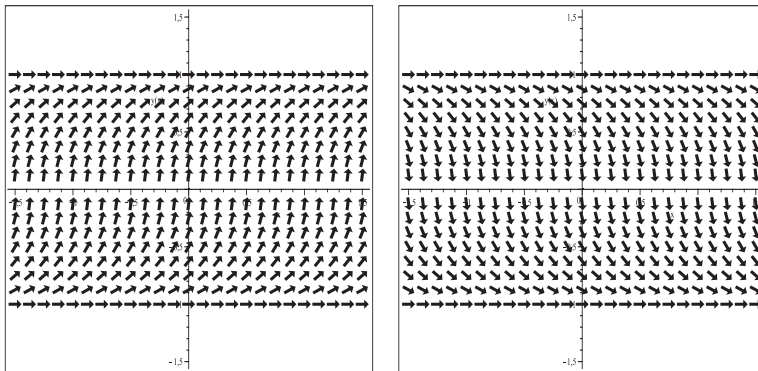


Abb. 2.225 Dateien *mehrd_Loes_11a1, 2.ps*, Richtungsfelder zur gDGL (11) als explizite gDGL in den beiden obigen Formen in $[-\frac{3}{2}, \frac{3}{2}] \times [-1, 0) \cup (0, 1]$, horizontale Richtungselemente bei $y = \pm 1$, Kommando `dfieldplot`

Mit dem Kommando `DEplot` erhält man interessante Ergebnisse.

Der Wertevorrat der Lösung ist $|y| \leq 1$, sodass der Darstellungsbereich (x, y) der Näherungslösungen dem angepasst werden kann. Dazu kommt eine geeignete Verfahrensauswahl mit der Schrittweitenwahl `stepsize`.

DEplot liefert keine numerische Lösung des AWP mit (11) bzw. der expliziten Form sowie der AB $y(x_0) = 0$.

Für $y = 0$ war ja auch kein Richtungsfeld möglich.

```
Error,
  (in DEtools/DEplot/CheckDE) unable to isolate DEs for required derivatives

Warning, plot may be incomplete, the following errors(s) were issued:
  cannot evaluate the solution past the initial point, problem may be complex,
  initially singular or improperly set up
```

Einige gute numerische Verfahren integrieren über die singuläre Stelle $y = 0$ hinweg und erzeugen eine durchgängige Kurve für alle $x \in \mathbb{R}$. Diese besteht aus 4 Teilstücken: v.l.n.r. $y(x) = -1$, unterer und oberer Viertelkreis, $y(x) = 1$ bzw. $y(x) = 1$, oberer und unterer Viertelkreis, $y(x) = -1$. Andere oder insbesondere ungenaue Verfahren integrieren entweder nicht über die singuläre Stelle hinweg oder liefern keine durchgängige Kurve bzw. nur das die AB enthaltende Kurvenstück.

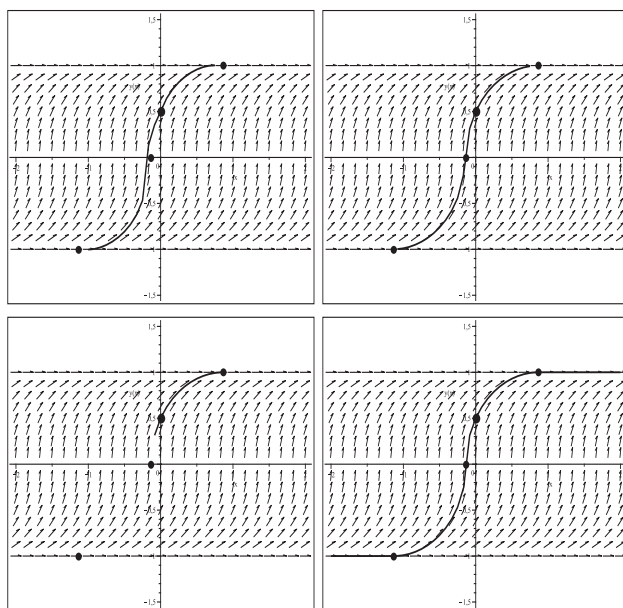


Abb. 2.226

Dateien

mehrd_loes_11b11..4.ps,

Richtungsfeld zur gDGl (11)

mit $y' > 0$,

Näherungslösung zur AB

$y(0) = \frac{1}{2}$ in $[-2, 2] \times [-\frac{3}{2}, \frac{3}{2}]$,

Verfahren v.l.o.n.r.u.

`method=classical[foreuler]`,

`method=classical[rk4]`,

`method=gear`,

`method=rkf45`,

Kommandos

`DEplot, pointplot`

Die Näherungsverfahren liefern die Warnungen

```
zu method=classical[foreuler], method=classical[rk4]:
Warning, plot may be incomplete, the following errors(s) were issued:
  solution becomes complex valued
```

```
zu method=gear:
Warning, plot may be incomplete, the following errors(s) were issued:
  solution becomes complex valued
  step taken with h = hmin but the requested error not achieved
```

Die letzte Näherung in Abb. 2.226 gehört zur Lösung

$$y(x) = \begin{cases} -1 & \text{für } x \in (-\infty, -2 + \sqrt{3}/2), \\ -\sqrt{1 - (x + 2 - \sqrt{3}/2)^2} & \text{für } x \in [-2 + \sqrt{3}/2, -1 + \sqrt{3}/2), \\ \sqrt{1 - (x - \sqrt{3}/2)^2} & \text{für } x \in [-1 + \sqrt{3}/2, \sqrt{3}/2], \\ 1 & \text{für } x \in (\sqrt{3}/2, +\infty), \end{cases} \quad y(0) = 1/2.$$

Für die AB $y(0) = -\frac{1}{2}$ ist es diese Funktion um $2 - \sqrt{3}$ nach rechts verschoben.

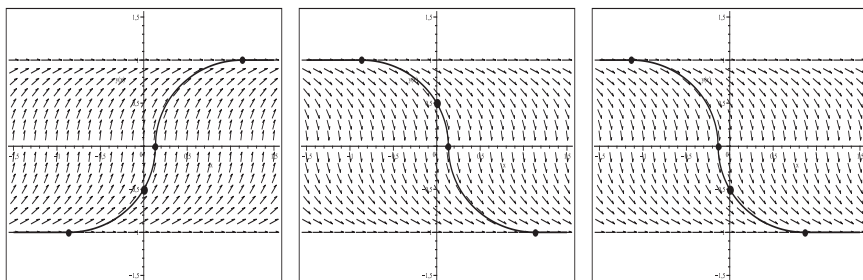


Abb. 2.227 Dateien *mehrd_loes_11b2..4.ps*, Richtungsfeld zur gDGL (11), Näherungslösungen mit AB $y(0) = \pm\frac{1}{2}$ in $[-\frac{3}{2}, \frac{3}{2}] \times [-1, 1]$, v.l.n.r. Richtungsfeld zu $y' > 0$, $y(0) = -\frac{1}{2}$, RF zu $y' < 0$, $y(0) = \pm\frac{1}{2}$, Kommandos `DEplot,method=rkf45, pointplot`

Mit `DEplot,method=rkf45` und der AB $y(x_0) = y_0$, $|y_0| < 1$, $y_0 \neq 0$, treffen die Kurven zu den Näherungslösungen nach links und rechts immer auf die stationären Lösungen. Letztere sind natürlich auch Näherungslösungen zu den AB $y(x_0) = \pm 1$. Die Verzweigung der Kurven kann man nicht als Mehrdeutigkeit deuten. Numerische Verfahren erzeugen zu jeder zulässigen AB eine eindeutige Näherung, mehr oder weniger genau.

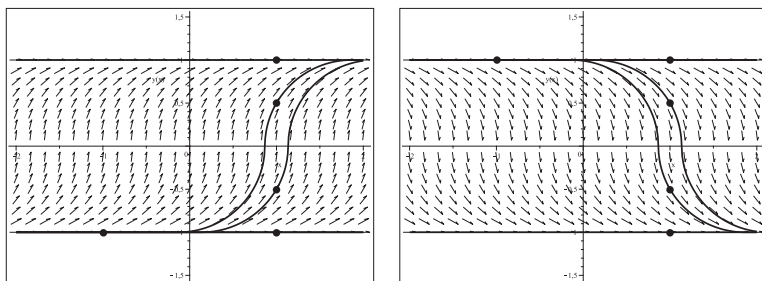


Abb. 2.228 Dateien *mehrd_loes_11b5, 6.ps*, Richtungsfeld zur gDGL (11) Näherungslösungen in $[-2, 2] \times [-1, 1]$, links: Richtungsfeld zu $y' > 0$, $y(-1) = -1$, $y(1) = -1, -\frac{1}{2}, \frac{1}{2}, 1$, rechts: Richtungsfeld zu $y' < 0$, $y(-1) = 1$, $y(1) = -1, -\frac{1}{2}, \frac{1}{2}, 1$, Kommandos `DEplot,method=rkf45, pointplot`

Nun kommen wir zur Anwendung des Kommandos `dsolve`.

Ohne Lösung bleiben die Befehle

```
> dsolve({dgl,y(x0)=y0},y(x));
dsolve({dgl,y(x0)=y0},y(x)) assuming abs(y0)<=1;
dsolve({dgl,y(1)=y0},y(x));
dsolve({dgl,y(1)=y0},y(x)) assuming abs(y0)<=1;
```

Die Einschränkung der AB verbessert die Situation.

```
> dsolve({dgl,y(x0)=y0},y(x)) assuming 0<=y0, y0<=1;
dsolve({dgl,y(x0)=y0},y(x)) assuming -1<=y0, y0<=0;
```

$$y_{1,2}(x) = \sqrt{y_0^2 \pm 2(x-x_0)\sqrt{1-y_0^2} - (x-x_0)^2},$$

$$y_{3,4}(x) = -\sqrt{y_0^2 \pm 2(x-x_0)\sqrt{1-y_0^2} - (x-x_0)^2}.$$

Die Lösungen sind mehrdeutig und paarweise Halbkreise ober- bzw. unterhalb der x -Achse mit Radius Eins und Mittelpunkt auf der x -Achse.

$y_1(x)$ und $y_3(x)$ haben den Definitionsbereich $[x_0 + \sqrt{1-y_0^2} - 1, x_0 + \sqrt{1-y_0^2} + 1]$ sowie den Mittelpunkt $(m_1, 0) = (x_0 + \sqrt{1-y_0^2}, 0)$, $y_2(x)$ und $y_4(x)$ den Definitionsbereich $[x_0 - \sqrt{1-y_0^2} - 1, x_0 - \sqrt{1-y_0^2} + 1]$ sowie $(m_2, 0) = (x_0 - \sqrt{1-y_0^2}, 0)$.

Mit den Anweisungen

```
> dsolve({dgl,y(x0)=1},y(x));
dsolve({dgl,y(x0)=-1},y(x));
```

ergeben sich nicht die beiden stationären Lösungen $y(x) = \pm 1$, sondern die Halbkreise $y(x) = \pm \sqrt{1 - (x-x_0)^2}$. Erst wenn x_0 einen konkreten Wert annimmt, folgt $y(x) = \pm 1$.

Zulässig ist hier auch die AB $y(x_0) = 0$. So folgen aus den Kommandos

```
> dsolve({dgl,y(x0)=0},y(x));
dsolve({dgl,y(1)=0},y(x));
dsolve({dgl,y(0)=0},y(x));
```

jeweils die vier Lösungen

$$y_{1,2}(x) = \sqrt{\pm 2(x-x_0) - (x-x_0)^2}, \quad y_{3,4}(x) = -\sqrt{\pm 2(x-x_0) - (x-x_0)^2},$$

$$y_{1,2,3,4}(x) = \pm \sqrt{\pm 2(x-1) - (x-1)^2} = \begin{cases} \pm \sqrt{-3+4x-x^2}, \\ \pm \sqrt{1-x^2}, \end{cases}$$

$$y_{1,2,3,4}(x) = \pm \sqrt{\pm 2x - x^2} = \pm \sqrt{x(\pm 2 - x)}.$$

Wir notieren noch die beiden Lösungen $y_{1,2}(x)$ zur AB $y(1) = \frac{1}{2}$. Die folgenden Kommandos in Maple 12 über die Auswertung der Wurzel liefern zusätzlich die beiden negativen Funktionszweige $y_{3,4}(x)$, in Maple 14 gibt es nur $y_{1,2}(x)$.

```
> dsolve({dgl,y(1)=1/2},y(x));
allvalues(%);
```

$$y(x) = \frac{1}{2} \text{RootOf}(4 \text{RootOf}(_Z^2 - 3) - 8x + 3 - 4 \text{RootOf}(_Z^2 - 3)x + 4x^2 + _Z^2),$$

$$y_{1,2}(x) = \sqrt{\left(\frac{1}{2}\right)^2 \pm 2(x-1) \sqrt{1 - \left(\frac{1}{2}\right)^2} - (x-1)^2} = \frac{1}{2} \sqrt{1 \pm 4(x-1)(1 + \sqrt{3} - x)},$$

$$y_{3,4}(x) = -\sqrt{\left(\frac{1}{2}\right)^2 \pm 2(x-1) \sqrt{1 - \left(\frac{1}{2}\right)^2} - (x-1)^2} = -\frac{1}{2} \sqrt{1 \pm 4(x-1)(1 + \sqrt{3} - x)}.$$

Komplexe Lösungen ergeben sich für AB $y(x_0) = y_0$, $|y_0| > 1$.

Zu den genannten Situationen mit mehrdeutigen Lösungen folgen einige grafische Darstellungen. Liefert `dsolve` das Ergebnis in der Form `RootOf`, so tauchen bei manchen AB die Zusatzfunktionen auf, die jedoch für diese AB keine Lösungen sind.

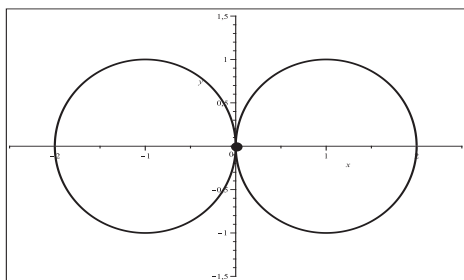


Abb. 2.229

Datei `mehrd_Loes_11c01.ps`,
 gDGl (11) mit AB $y(0) = 0$,
 4 Lösungen (Halbkreise)
 $y(x) = \pm \sqrt{\pm 2x - x^2}$, $x \in [0, 2] \vee [-2, 0]$,
 Kommandos `dsolve, plot, pointplot`

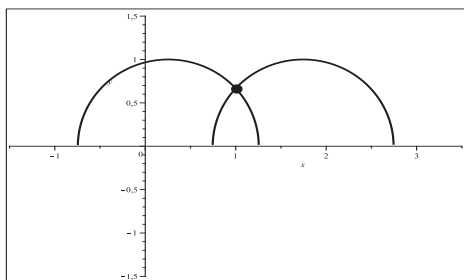


Abb. 2.230

Datei `mehrd_Loes_11c02.ps`,
 gDGl (11) mit AB $y(1) = \frac{2}{3}$,
 2 Lösungen (Halbkreise)
 $y(x) = \sqrt{\frac{4}{9} \pm 2(x-1) \frac{\sqrt{5}}{3} - (x-1)^2}$,
 $x \in [\frac{\sqrt{5}}{3}, 2 + \frac{\sqrt{5}}{3}] \vee [-\frac{\sqrt{5}}{3}, 2 - \frac{\sqrt{5}}{3}]$,
 Kommandos `dsolve, plot, pointplot`

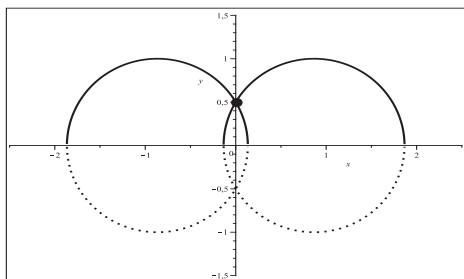
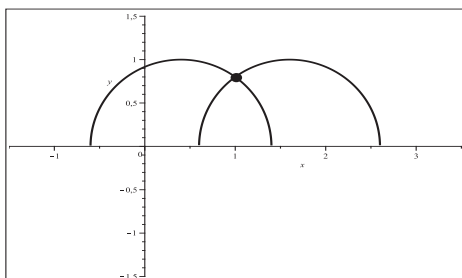


Abb. 2.231

Datei `mehrd_Loes_11c03.ps`,
 gDGl (11) mit AB $y(0) = \frac{1}{2}$,
 2 Lösungen (Halbkreise)
 $y(x) = \sqrt{\frac{1}{4} \pm x\sqrt{3} - x^2}$,
 $x \in [\frac{\sqrt{3}}{2} - 1, \frac{\sqrt{3}}{2} + 1] \vee [-\frac{\sqrt{3}}{2} - 1, -\frac{\sqrt{3}}{2} + 1]$,
 2 Zusatzfunktionen bei Maple 12
 $y(x) = -\sqrt{\frac{1}{4} \pm x\sqrt{3} - x^2}$,
 ohne diese Funktionen bei Maple 14,
 Kommandos `dsolve, (allvalues), plot, pointplot`

Mit der AB $y(1) = \frac{1}{2}$ erhält man eine ähnliche Darstellung wie in der Abbildung 2.231. Die Kurven sind nur um Eins nach rechts verschoben.

**Abb. 2.232**

Datei *mehrd_loes_11c05.ps*,
gDGl (11) mit AB $y(1) = \frac{4}{5}$,

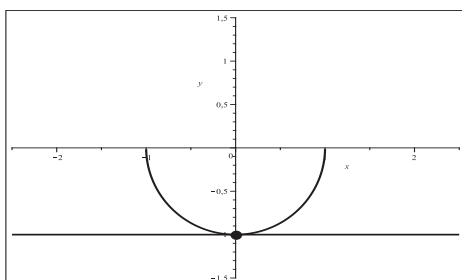
2 Lösungen (Halbkreise)

$$y(x) = \sqrt{\frac{16}{25} \pm (x-1)\frac{6}{5} - (x-1)^2},$$

$$x \in \left[\frac{3}{5}, \frac{13}{5}\right] \vee \left[-\frac{3}{5}, \frac{7}{5}\right],$$

Kommandos `dsolve, plot, pointplot`

Nimmt man die allgemeine parameterabhängige Lösung $\pm\sqrt{1-(x-c)^2}$, so hat man für $c=0$ die Halbkreise $y = \pm\sqrt{1-x^2}$. Zusammen mit den stationären Lösungen folgen zur AB $y(0) = -1$ die beiden Lösungen $y(x) = -1$, $y(x) = -\sqrt{1-x^2}$, zu $y(0) = 1$ die beiden Lösungen $y(x) = 1$, $y(x) = \sqrt{1-x^2}$, also die Mehrdeutigkeit.

**Abb. 2.233**

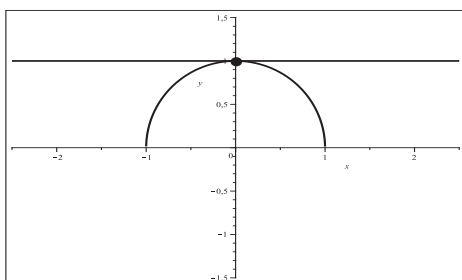
Datei *mehrd_loes_11c06.ps*,

gDGl (11) mit AB $y(0) = -1$,

2 Lösungen $y(0) = -1$, $x \in (-\infty, \infty)$,

und $y(x) = -\sqrt{1-x^2}$, $x \in [-1, 1]$,

Kommandos `dsolve, plot, pointplot`

**Abb. 2.234**

Datei *mehrd_loes_11c07.ps*,

gDGl (11) mit AB $y(0) = 1$,

2 Lösungen $y(0) = 1$, $x \in (-\infty, \infty)$,

und $y(x) = \sqrt{1-x^2}$, $x \in [-1, 1]$,

Kommandos `dsolve, plot, pointplot`

$$(12) \quad y' = \frac{y-2}{x(2x-1)}$$

$$y(x) = c \frac{2x-1}{x} + \frac{1}{x} = 2c + (1-c) \frac{1}{x}$$

Man erkennt die konstante Lösung $y(x) = 2$, die mit dem Parameter $c = 1$ korrespondiert. Sie ist überall definiert, obwohl die rechte Seite der gDGl als auch die allgemeine Lösung an bestimmten Stellen nicht auswertbar sind.

An den Stellen $x = 0, \frac{1}{2}$ gibt es kritische Situationen. $P(\frac{1}{2}, 2)$ ist ein singulärer Punkt, obwohl die Lösungsformel an der Stelle $x = \frac{1}{2}$ ausgewertet werden kann. Somit ist mit mehrdeutigen Lösungen zu rechnen.

Das Richtungsfeld ist nicht für $x = 0, \frac{1}{2}$ definiert, bei $y = 2$ gibt es horizontale Richtungselemente.

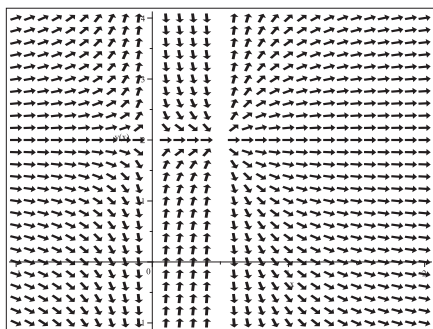


Abb. 2.235

Datei *mehrd_Loes_12a.ps*,

Richtungsfeld zur gDGI (12),

Kommando `dfieldplot`

Mit dem Kommando `DEplot` erhält man zu jeder zulässigen AB eine eindeutige Näherungslösung, aber eben nicht zu allen. Zu den AB $y(x_0) = y_0$ mit $x_0 = 0$ oder $x_0 = \frac{1}{2}$ gibt es keine Ergebnisse, eine Ausnahme bildet die AB $y(0) = 2$ mit der Lösung $y(x) = 2$.

Bei den numerischen Verfahren beschränken wir uns auf die Methode `rkf45`.

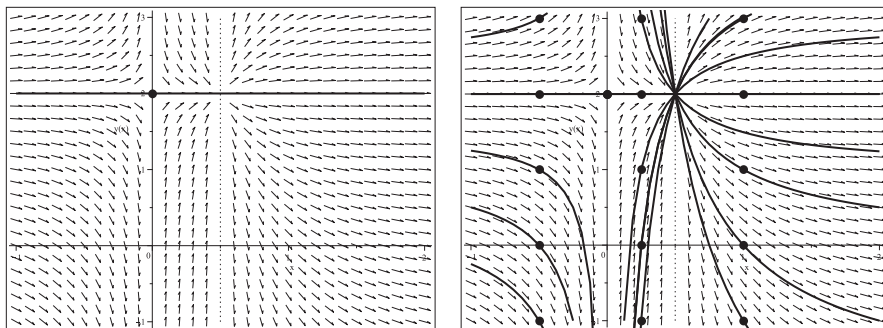


Abb. 2.236 Dateien *mehrd_Loes_12b1, 2.ps*, Richtungsfeld zur gDGI (12)

Näherungslösungen in $[-1, 2] \times [-1, 3]$,

links: zur AB $y(0) = 2$ mit Lösung $y(x) = 2$,

rechts: zu den AB $y(0) = 2$ und $y(x_0) = -1(1)3, x_0 = -\frac{1}{2}, \frac{1}{4}, 1$,

Kommandos `DEplot, method=rkf45, plot, pointplot`

Einige Bemerkungen bei Verwendung des Kommandos `dsolve`.

Aus `dsolve(dg1, y(x))` folgt die obige allgemeine parameterabhängige Darstellung. Darunter ist für $c = 1$ die konstante Lösung $y(x) = 2$. Die anderen Funktionen bestehen jeweils aus zwei Zweigen in den Halbebenen $x < 0$ und $x > 0$. Liegt eine AB $y(x_0) = y_0$ vor, so ist ein Zweig die gesuchte Lösung des entsprechenden AWP.

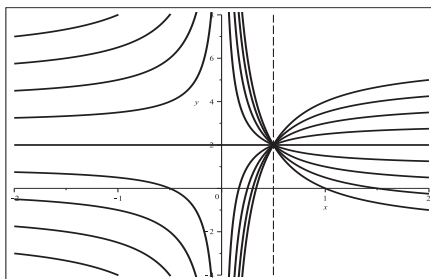


Abb. 2.237

Datei *mehrdJoel_12c1.ps*,
 gDGL (12), Lösungsfunktionen
 $y(x) = 2c + (1 - c)\frac{1}{x}$, $c = -1(\frac{1}{2})3$,
 Kommandos `solve`, `plot`

Wenn für das AWP die AB $y(\frac{1}{2}) = 2$ zulässig wäre, gäbe es damit unendlich viele Lösungen dieses AWP, gehen doch unendlich viele Kurven durch den Punkt $P(\frac{1}{2}, 2)$.

Anders als bei `DEplot` ist im Kommando `dsolve` diese AB möglich.

Wie `dsolve` ohne diese AB liefert `dsolve(dg1, y(1/2)=2, y(x))` die allgemeine Lösung $y(x) = 2c + (1 - c)\frac{1}{x}$. Man hat so mit dem freien Parameter c die unendliche Kurvenschar und damit die Mehrdeutigkeit.

Trotzdem sollte man die Lösung des AWP in Maple mittels `dsolve(dg1, y(x0)=y0, y(x))` etwas genauer untersuchen.

I. Allg. erhält man eindeutige Lösungen. Natürlich können verschiedene AB auf der gleichen Lösungskurve liegen. `dsolve(dg1, y(1)=3, y(x))` und `dsolve(dg1, y(1/4)=0, y(x))` ergeben die Funktion $4 - \frac{1}{x}$, deren Zweig in der Halbebene $x > 0$ die Lösung darstellt.

Das Ergebnis zu `dsolve(dg1, y(1/4)=0, y(x))` ist

$$y(x) = -\frac{(2x-1)\left(-y_0 + \frac{1}{x_0}\right)x_0}{x(2x_0-1)} + \frac{1}{x}$$

In dieser Darstellung wird die weitere Verarbeitung problematisch, insbesondere wenn mit $x_0 = 0$ bzw. $x_0 = \frac{1}{2}$ gerechnet werden soll.

```
> dsol:=dsolve({dg1, y(x0)=y0}, y(x));
> subs(x0=1, y0=2, rhs(dsol)); simplify(%);
subs(y0=2, rhs(dsol)); simplify(%);
> # Error, numeric exception: division by zero
subs(x0=0, rhs(dsol)); subs(x0=0, y0=2, rhs(dsol));
subs(x0=1/2, rhs(dsol)); subs(x0=1/2, y0=2, rhs(dsol));
```

$$\frac{2x-1}{x} + \frac{1}{x}$$

$$2$$

$$-\frac{(2x-1)\left(-2+\frac{1}{x}\right)x_0}{x(2x_0-1)} + \frac{1}{x}$$

$$2$$

Die Substitution von $y_0 = 2$, ohne das x_0 zu belegen, und eine Vereinfachung führt zum Ergebnis 2, d.h. zur konstanten Lösung. Nimmt man jedoch $x_0 = 0$ bzw. $x_0 = \frac{1}{2}$ dazu, folgt die Nulldivision.

Anders bei `simplify(dsolve(dg1,y(x0)=y0,y(x)))` mit der nun vereinfachten Formel

$$y(x) = \frac{2x y_0 x_0 - 2x - y_0 x_0 + 2x_0}{x(2x_0 - 1)} = 2 + \frac{(2x - 1)x_0(y_0 - 1)}{x(2x_0 - 1)}.$$

Die Division durch x_0 ist dabei eliminiert worden und das hat folgende Auswirkungen.

```
> dsols:=simplify(dsolve({dg1,y(x0)=y0},y(x))):
> subs(x0=1,y0=2,rhs(dsols));
  subs(y0=2,rhs(dsols)); simplify(%);
  subs(x0=0,rhs(dsols));
  subs(x0=0,y0=2,rhs(dsols));
> # Error, numeric exception: division by zero
  subs(x0=1/2,rhs(dsols));
  subs(x0=1/2,y0=2,rhs(dsols));
```

2

$$\frac{4x_0 x - 2x}{x(2x_0 - 1)}$$

2

2

2

Hier führen die Substitution von $y_0 = 2$, ohne das x_0 zu belegen, sowie die Substitution von $x_0 = 0$, ohne y_0 zu setzen, bei einer evtl. Vereinfachung zum Ergebnis 2.

Dann gibt es noch einige Sonderfälle.

- Keine Lösung bei $y(\frac{1}{2}) = y_0$, auch wenn y_0 konkrete Werte $\neq 2$ annimmt.
- Lösung $y(x) = 2$ bei $y(0) = y_0$ bzw. wenn hier y_0 einen beliebigen Wert hat.
- Natürlich folgt die Lösung $y(x) = 2$ bei $y(x_0) = 2$, wobei $x_0 \neq 0, \frac{1}{2}$ ist.

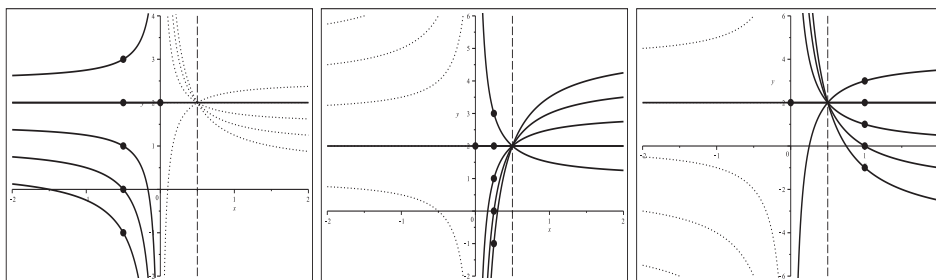


Abb. 2.238 Dateien *mehrd_Loes_12c2.4.ps*, gDGI (12)

Lösungen $y(x)$ zur AB $y(x_0) = y_0$ mit jeweils zweitem Funktionszweig,

links in $[-2, 2] \times [-2, 4]$: AB $y(0) = 2$, $y(-\frac{1}{2}) = -1(1)3$,

Mitte in $[-2, 2] \times [-2, 6]$: AB $y(0) = 2$, $y(\frac{1}{4}) = -1(1)3$,

rechts in $[-2, 2] \times [-6, 6]$: AB $y(0) = 2$, $y(1) = -1(1)3$,

Kommandos `dsolve`, `plot`, `pointplot`

Singuläre oder spezielle Stellen in der gDGL, die zumeist die Mehrdeutigkeit der Lösung verursachen, können bei der numerischen bzw. analytischen Behandlung der gDGL zu unterschiedlichen Ergebnissen führen. Auch die Akzeptanz der AB durch die Lösungsmethoden ist manchmal verschieden. Numerische Verfahren zu AWP finden bei Mehrdeutigkeit oft den "einfacheren" Lösungszweig. Dann macht erst die exakte Lösung der gDGL mögliche weitere Lösungswege sichtbar.

Bei gDGL höherer Ordnung, insbesondere bei nichtlinearen, treten mehrdeutige Situationen öfter auf.

Scharen von Lösungskurven entstehen natürlich, wenn die Menge der AB nicht vollständig angegeben wird.

Beispiel 2.8

$$(1) \quad y''(x + 2y') = 0$$

$$y(x) = c_1 + c_2x, \quad c - \frac{1}{4}x^2$$

Die 2. Ordnung dieser gDGL und ihre Nichtlinearität führen zunächst auf zwei parameterabhängige Teillösungen, die überall definiert sind.

Die Ordnung verhindert die Erzeugung eines Richtungsfelds mittels `dfieldplot`.

`Error, (in DEtools/dfieldplot) cannot produce plot non-autonomous DE(s) require initial conditions.`

Das Kommando `DEplot` verlangt AB der Form $y(x_0) = y_0$, $y'(x_0) = y'_0$. Aber genau diese bestimmen eindeutig die beiden Parameter in der linearen Funktion $c_1 + c_2x$ der Lösungsmannigfaltigkeit, während die quadratische Parabel nur einen Freiheitsgrad hat. Berechnete Näherungslösungen sind im linearen Fall exakt. Ein Richtungsfeld wird mit `DEplot` für die gDGL 2. Ordnung nicht erzeugt.

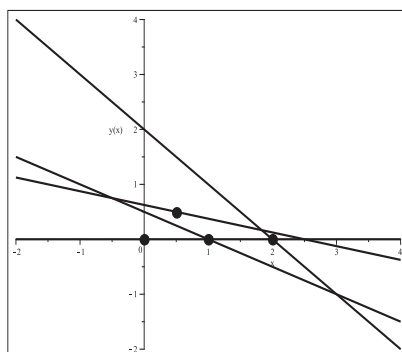


Abb. 2.239

Datei `mehrd_Loes_13b.ps`, gDGL (1),

Näherungslösungen zu den AB

$$\{y(2) = 0, y'(2) = -1\}, \{y(1) = 0, y'(1) = -\frac{1}{2}\}, \\ \{y(0) = 0, y'(0) = 0\}, \{y(\frac{1}{2}) = \frac{1}{2}, y'(\frac{1}{2}) = -\frac{1}{4}\},$$

zu den exakten Lösungen $y(x) = 2 - x$,

$$y(x) = \frac{1}{2} - \frac{x}{2}, \quad y(x) = 0, \quad y(x) = \frac{5}{8} - \frac{x}{4},$$

$$(x, y) \in [-2, 4] \times [-2, 4],$$

Kommando `DEplot, method=rkf45`

Das Kommando `dsolve(dg1, y(x))` liefert die obigen beiden Lösungen.

Bei zwei allgemeinen oder unabhängigen AB gemäß dem Kommando

`dsolve(dg1, y(x0)=y0, D(y)(x0)=y0s, y(x))` folgt immer die Geradengleichung

$$y(x) = y_0 + y'_0(x - x_0).$$

Wird die AB auf die Bedingung $y(x_0) = y_0$ reduziert, also $\text{dsolve}(\text{dgl}, y(x_0)=y_0, y(x))$, bleibt es bei den zwei Lösungen, wobei die Gerade einen freien Parameter hat, aber die Parabel eindeutig bestimmt ist, also

$$y(x) = y_0 + c(x - x_0), \quad y(x) = y_0 - \frac{1}{4}(x^2 - x_0^2).$$

Nun kommen wir zur Mehrdeutigkeit, wo Gerade und Parabel einen gemeinsamen Berührungspunkt haben und dort gleiche Tangentenanstiege besitzen. Genau aus diesem Umstand folgt, dass in der AB die Ableitung $y'(x_0) = -\frac{x_0}{2}$ sein muss.

Mit $\text{dsolve}(\text{dgl}, y(x_0)=y_0, D(y)(x_0)=-x_0/2, y(x))$ ergeben sich die Geradengleichung $y(x) = y_0 - \frac{x_0}{2}(x - x_0)$ und die Parabel $y(x) = y_0 - \frac{1}{4}(x^2 - x_0^2)$.

Bei zwei AB mit $y'(x_0) = y'_0 \neq -\frac{x_0}{2}$ entsteht nur die Geradengleichung, weil der Ableitungswert y'_0 (Tangente) an der Stelle x_0 nicht an die Parabel passt.

Nun kann man beliebige Situationen der Zweideutigkeit aufzeichnen.

```
> dsol12:=dsolve({dgl,y(x0)=y0,D(y)(x0)=-x0/2},y(x)):
> f1:=unapply(rhs(dsol12[1]),x,x0,y0); f2:=unapply(rhs(dsol12[2]),x,x0,y0);
# Kontrolle der AB
f1(x0,x0,y0); D[1](f1)(x0,x0,y0); f2(x0,x0,y0); D[1](f2)(x0,x0,y0);
> pl1:=plot([f1(x,1,0),f2(x,1,0)],x=-1..2,y=-1..1,color=black,thickness=3):
pl2:=pointplot([[1,0]],color=black,symbol=solidcircle,symbolsize=24):
display(pl1,pl2);
```

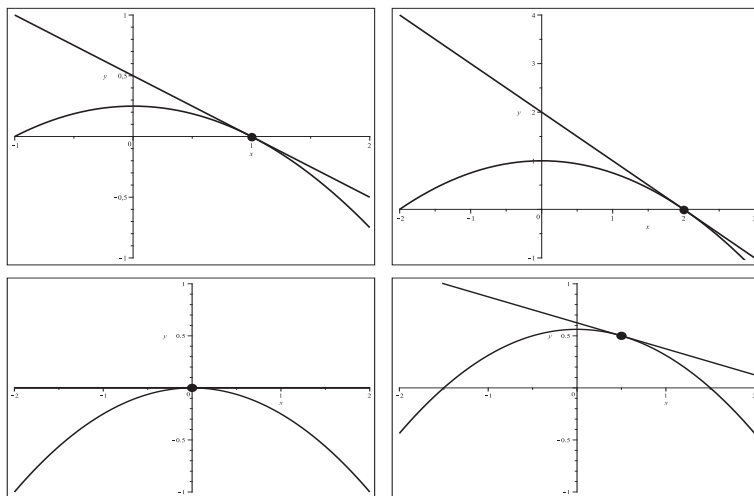


Abb. 2.240 Dateien *mehrdLoes_13c1.4.ps*, gDGl (1), Lösungen v.l.o.n.r.u.

$$\begin{aligned}
 y_1(x) &= \frac{1}{2}(1-x), & y_2(x) &= \frac{1}{4}(1-x^2) & \text{zur AB } & y(1) = 0, & y'(1) &= -\frac{1}{2}, \\
 y_1(x) &= 2-x, & y_2(x) &= 1-\frac{1}{4}x^2 & \text{zur AB } & y(2) = 0, & y'(2) &= -1, \\
 y_1(x) &= 0, & y_2(x) &= -\frac{1}{4}x^2 & \text{zur AB } & y(0) = 0, & y'(0) &= 0, \\
 y_1(x) &= \frac{5}{8} - \frac{1}{4}x, & y_2(x) &= \frac{9}{16} - \frac{1}{4}x^2 & \text{zur AB } & y\left(\frac{1}{2}\right) = \frac{1}{2}, & y'\left(\frac{1}{2}\right) &= -\frac{1}{4},
 \end{aligned}$$

Kommandos `dsolve`, `plot`, `pointplot`

$$(2) \quad y'''' - y = 0$$

$$y(x) = c_1 e^{-x} + c_2 e^x + c_3 \sin(x) + c_4 \cos(x)$$

Die 4. Ordnung dieser linearen gDGI mit konstanten Koeffizienten führt auf eine überall definierte parameterabhängige Lösung mit dem Freiheitsgrad 4 und dem Fundamentalsystem $\{e^{-x}, e^x, \sin(x), \cos(x)\}$.

Die Ordnung verhindert die Erzeugung eines Richtungsfelds mittels `dfieldplot`.

Das Kommando `DEplot` verlangt AB der Form $y(x_0) = y_0$, $y'(x_0) = y'_0$, $y''(x_0) = y''_0$, $y'''(x_0) = y'''_0$. Genau diese bestimmen eindeutig die vier Parameter in der allgemeinen Lösung. Näherungslösungen können mit guter Genauigkeit berechnet werden.

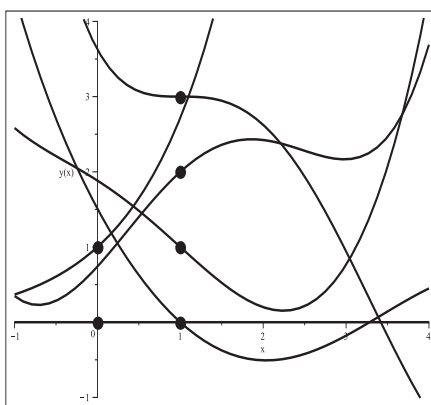


Abb. 2.241

Datei `mehrd_Loes_14b.ps`, gDGI (2),

Näherungslösungen zu den AB

$$\{y(1) = 0, y'(1) = -1, y''(1) = 1, y'''(1) = 0\},$$

$$\{y(1) = 1, y'(1) = -1, y''(1) = 0, y'''(1) = 1\},$$

$$\{y(1) = 2, y'(1) = 1, y''(1) = -1, y'''(1) = -1\},$$

$$\{y(1) = 3, y'(1) = 0, y''(1) = 0, y'''(1) = -3\},$$

$$\{y(0) = 0, y'(0) = 0, y''(0) = 0, y'''(0) = 0\},$$

$$\{y(0) = 1, y'(0) = 1, y''(0) = 1, y'''(0) = 1\},$$

$$(x, y) \in [-1, 4] \times [-1, 4],$$

Kommandos `DEplot, method=rkf45, pointplot`

Die exakten Lösungen dazu sind in der Reihenfolge der obigen AB

$$y(x) = \frac{1}{2} e^{1-x} - \frac{1}{2} (d_1 \sin(x) + d_2 \cos(x)),$$

$$y(x) = \frac{1}{4} (e^{1-x} + e^{-1+x}) + \frac{1}{2} (d_2 \sin(x) + d_1 \cos(x)),$$

$$y(x) = \frac{1}{4} (e^{1-x} + e^{-1+x}) + \frac{3}{2} (d_1 \sin(x) - d_2 \cos(x)),$$

$$y(x) = \frac{3}{2} e^{1-x} + \frac{3}{2} (d_1 \sin(x) - d_2 \cos(x)),$$

$$y(x) = 0, \quad y(x) = e^x,$$

wobei $d_1 = \sin(1) + \cos(1)$, $d_2 = \sin(1) - \cos(1)$.

Die exakte Lösung des AWP mit dem vollständigen System von AB gemäß

`dsolve(dg1, y(x0)=y0, D(y)(x0)=y0s, (D@@2)(y)(x0)=y02s, (D@@3)(y)(x0)=y03s, y(x))`
ergibt

$$\begin{aligned} y(x) = & \frac{1}{4} [(y_0 - y'_0 + y''_0 - y'''_0) e^{x_0-x} + (y_0 + y'_0 + y''_0 + y'''_0) e^{-x_0+x}] \\ & + \frac{1}{2} [(y_0 - y''_0) \sin(x_0) + (y'_0 - y'''_0) \cos(x_0)] \sin(x) \\ & + \frac{1}{2} [(-y'_0 + y'''_0) \sin(x_0) + (y_0 - y''_0) \cos(x_0)] \cos(x). \end{aligned}$$

Sehr einfache Lösungen gibt es zu den beiden AB $y^{(k)}(0) = 0$, $y^{(k)}(0) = 1$, $k = 0(1)3$, nämlich $y_1(x) = 0$ bzw. $y_2(x) = e^x$.

AB und zugehörige Lösungen sind

$$y(0) = 1, y'(0) = 0, y''(0) = 0, y'''(0) = 0, \quad y_3(x) = \frac{1}{2} (\cosh(x) + \cos(x)),$$

$$y(0) = 0, y'(0) = 1, y''(0) = 0, y'''(0) = 0, \quad y_4(x) = \frac{1}{2} (\sinh(x) + \sin(x)),$$

$$y(0) = 0, y'(0) = 0, y''(0) = 1, y'''(0) = 0, \quad y_5(x) = \frac{1}{2} (\cosh(x) - \cos(x)),$$

$$y(0) = 0, y'(0) = 0, y''(0) = 0, y'''(0) = 1, \quad y_6(x) = \frac{1}{2} (\sinh(x) - \sin(x)),$$

$$\text{wobei } \cosh(x) = \frac{1}{2}(e^x + e^{-x}), \quad \sinh(x) = \frac{1}{2}(e^x - e^{-x}).$$

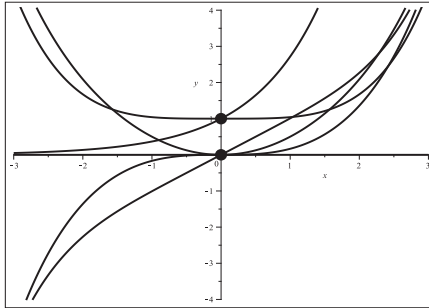


Abb. 2.242

Datei *mehrd_Loes.14c.ps*, gDGI (2),

Lösungen

$y_m(x)$, $m = 1(1)6$, in $[-3, 3] \times [-4, 4]$,

Kommandos `dsolve, plot, pointplot`

Ist das System der AB nicht vollständig, bleiben freie Parameter in der Lösungsformel und es entsteht eine unendliche Schar von Kurven. Das hat natürlich mit der Mehrdeutigkeit der Lösung eines AWP nichts zu tun.

AB und zugehörige Lösungen sind dann z.B.

$$y(0) = 1, \quad y(x) = (1 - c_1 - c_2)e^x + c_1 e^{-x} + c_3 \sin(x) + c_2 \cos(x),$$

$$y(0) = y'(1) = 1, \quad y(x) = (1 - c_1 - c_2)e^x + c_1 e^{-x} + (2c_1 + c_2) \sin(x) + c_2 \cos(x),$$

$$y(0) = y'(0) = y''(0) = 1, \quad y(x) = (1 - c_1)e^x + c_1 e^{-x} + 2c_1 \sin(x),$$

$$y(0) = y'(0) = y''(0) = y'''(0) = 1, \quad y(x) = e^x.$$

2.6 Systeme gewöhnlicher Differentialgleichungen

Bei der Untersuchung und Lösung von SysgDGI verweisen wir zunächst auf den Abschnitt 2.2.3 und die Erläuterungen zum Beispiel mit dem SysgDGI (2.3).

In den Kommandos `dfieldplot`, `DEplot` und `dsolve` spielen u.a. Dimension, Ordnung, Autonomie und Richtungsfeld zum Phasenraum des SysgDGI eine Rolle. Neben den Teillösungen (Koordinatenfunktionen) können auch Phasenkurven betrachtet werden, welche mit der Option `scene` ausgewählt werden. Das übliche Richtungsfeld wie bei der expliziten gDGI 1. Ordnung tritt nicht auf.

Am einfachsten ist die Handhabung eines Systems zweier autonomer gDGI 1. Ordnung. Hier wird mittels `dfieldplot` oder `DEplot` das Richtungsfeld zum Phasenraum angezeigt, bei `DEplot` auch die Phasenkurve (`scene=[x(t),y(t)]`) im Richtungsfeld, wenn dazu das entsprechende AWP gewählt wurde. Sind in `DEplot` z.B. die Optionen `scene=[t,x(t)]` oder `scene=[t,y(t)]` eingestellt, so werden nur die Kurven $x(t)$ bzw. $y(t)$ gezeigt. Bei Systemen zweier nicht autonomer gDGI 1. Ordnung sowie mit drei und mehr (autonomen) gDGI 1. Ordnung wird kein Richtungsfeld angegeben.

Beispiel 2.9

(1) Würfe (Wurfparabeln) unter vereinfachten Bedingungen

(1.1) Horizontaler Wurf mit Anfangsgeschwindigkeit v_0 , Masse m und Erdbeschleunigung g

$$mx'' = 0, \quad my'' = -mg$$

$$x(t) = c_1 t + c_2, \quad y(t) = -\frac{1}{2}gt^2 + c_3 t + c_4$$

bzw. als AWP

$$x'' = 0, \quad x(0) = 0, \quad x'(0) = v_0,$$

$$y'' = -g, \quad y(0) = 0, \quad y'(0) = 0$$

$$x(t) = v_0 t, \quad y(t) = -\frac{1}{2}gt^2$$

Für $g = 9.81$, $m = 1$ und $v_0 = 20$ erhält man $x(t) = 20t$, $y(t) = -\frac{981}{200}t^2 = -4.905t^2$ und die Phasenkurve $(x(t), y(t))$.

Diese kann auch notiert werden als Funktion $y(x) = -\frac{981}{80000}x^2 = -0.0122625x^2$.

Ein Richtungsfeld wird mit `dfieldplot` bzw. `DEplot` für dieses SysgDGI nicht erzeugt.

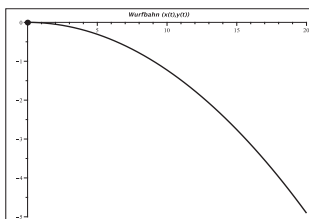


Abb. 2.243 Datei `sys_Loes_01a.ps`, gDGI (1.1),

Lösung $x(t) = 20t$, $y(t) = -\frac{981}{200}t^2$

als Phasenkurve (Wurfbahn) $(x(t), y(t))$, $t \in [0, 1]$,

mit der AB $x(0) = 0$, $x'(0) = 20$, $y(0) = 0$, $y'(0) = 0$,

$(x, y) \in [0, 20] \times [-5, 0]$,

Kommandos `dsolve`, `plot`, `pointplot`

(1.2) Schräger Wurf mit der Anfangsgeschwindigkeit v_0 , Masse m , Wurfwinkel α und Abwurfhöhe h als nicht autonomes System 1. Ordnung

$$\begin{aligned} m x' &= v_0 \cos(\alpha), & x(0) &= 0, \\ m y' &= m(v_0 \sin(\alpha) - gt), & y(0) &= h \\ x(t) &= \frac{v_0 \cos(\alpha) t}{m}, & y(t) &= -\frac{1}{2}gt^2 + v_0 \sin(\alpha) t + h \end{aligned}$$

Das Auftreffen des Wurfes, d.h. $y(t) = 0$, ergibt die Wurfzeit und Wurfweite

$$T = \frac{1}{g}(v_0 \sin(\alpha) + \sqrt{v_0^2 \sin^2(\alpha) + 2gh}), \quad X = x(T).$$

Für $m = 1$, $h = 4$, $v_0 = 20$, $\alpha = \frac{\pi}{6}$ sind das die Werte $T = 2.381$, $X = 41.243$.

```
> g:='g': m:='m': h:='h': v0:='v0': alpha:='alpha':
  inits2:=x(0)=0,y(0)=h:
  dgls2:=diff(x(t),t)*m=v0*cos(alpha),
          diff(y(t),t)*m=(v0*sin(alpha)-g*t)*m:
  vars:=x(t),y(t):

> r2:=dsolve({dgls2,inits2},{vars});
  xt:=rhs(op(1,r2));
  yt:=rhs(op(2,r2));
  T2:=solve(yt,t); # 2 Wurzeln
  TT:=T2[1];
  XX:=subs(t=TT,xt);
> T:=evalf(subs(g=9.81,m=1,h=4,v0=20,alpha=Pi/6,TT));
  X:=evalf(subs(g=9.81,m=1,h=4,v0=20,alpha=Pi/6,XX));
```

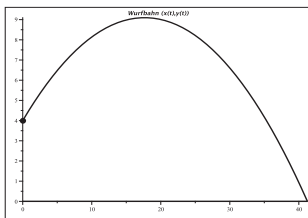


Abb. 2.244 Datei *sys_Loes_01b.ps*, gDGI (1.2), Lösung $x(t) = 10\sqrt{3}t$, $y(t) = -\frac{981}{200}t^2 + 10t + 4$ als Phasenkurve (Wurfbahn) $(x(t), y(t))$, $t \in [0, T]$, mit der AB $x(0) = 0$, $y(0) = 4$, $(x, y) \in [0, X] \times [0, 10]$, Kommandos `dsolve`, `plot`, `pointplot`

Die berechneten Näherungslösungen sind sehr genau.

```
> g:=9.81: m:=1: h:=4: v0:=20: alpha:=Pi/6: # Digits=16
> pts:=[[0,0,4]]:
  pl3:=DEplot([dgls2],[vars],t=0..T,x=0..X,y=0..10,pts,
              scene=[x(t),y(t)], # Standard bei [vars]
              # scene=[t,x(t)]
              # scene=[t,y(t)]
              linecolor=black):
  pl3a:=pointplot([0,4],color=black,symbol=solidcircle,symbolsize=24):
  display(pl3,pl3a);
```

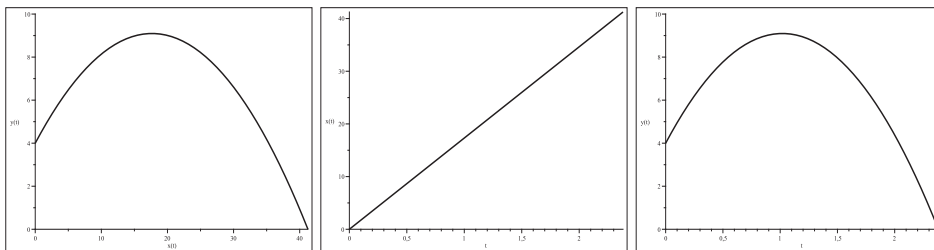


Abb. 2.245 Datei *sys_Loes_01c1..3.ps*, gDGL (1.2),
 Näherungslösung zu $x(t) = 10\sqrt{3}t$, $y(t) = -\frac{981}{200}t^2 + 10t + 4$
 als Phasenkurven $(x(t), y(t))$, $(t, x(t))$, $(t, y(t))$, $t \in [0, T]$,
 mit der AB $x(0) = 0$, $y(0) = 4$,
 $(x, y) \in [0, X] \times [0, 10]$, $[0, T] \times [0, X]$, bzw. $[0, T] \times [0, 10]$,
 Kommandos `DEplot`, `pointplot`

Der schräge Wurf mit negativer Anfangsgeschwindigkeit v_0 bei gleichem Wurfwinkel α und Abwurfhöhe h wird auch erhalten, wenn man $v_0 > 0$ nimmt und die Kurve nach links von $t = 0$ verfolgt.

Das Auftreffen des Wurfes “auf der linken Seite“ liefert die Wurfzeit und Wurfweite

$$T_l = \frac{1}{g}(v_0 \sin(\alpha) - \sqrt{v_0^2 \sin^2(\alpha) + 2gh}), \quad X_l = x(T_l).$$

Es ist die andere Wurzel der allgemeinen Wurfparabel (quadratische Parabel) aus der Bedingung $y(t) = 0$.

```
> T2:=solve(yt,t); # 2 Wurzeln
TT1:=T2[2];
XX1:=subs(t=TT1,xt);
> Tl:=evalf(subs(g=9.81,m=1,h=4,v0=20,alpha=Pi/6,TT1));
Xl:=evalf(subs(g=9.81,m=1,h=4,v0=20,alpha=Pi/6,XX1));
```

Für $m = 1$, $h = 4$, $v_0 = 20$, $\alpha = \frac{\pi}{6}$ sind das die Werte $T_l = -0.342$, $X_l = -5.931$.

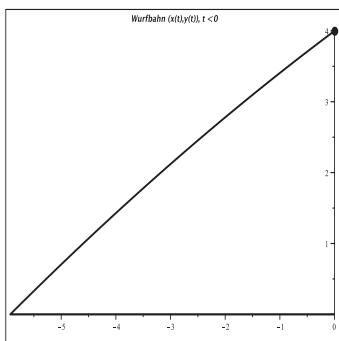


Abb. 2.246
 Datei *sys_Loes_01d.ps*, gDGL (1.2),
 Näherungslösung zu
 $x(t) = 10\sqrt{3}t$, $y(t) = -\frac{981}{200}t^2 + 10t + 4$
 als Phasenkurve (Wurfbahn) $(x(t), y(t))$, $t \in [T_l, 0]$,
 mit der AB $x(0) = 0$, $y(0) = 4$,
 $(x, y) \in [X_l, 0] \times [0, 4]$,
 Kommandos `DEplot`, `pointplot`

(2) Nicht autonome lineare SysgDGI

$$(2.1) \quad x' = y, \quad y' = -2x/t^2 + 2y/t$$

$$x(t) = c_1 t^2/2 + c_2 t, \quad y(t) = c_1 t + c_2$$

bzw. als AWP

$$x' = y, \quad x(1) = 1,$$

$$y' = -2x/t^2 + 2y/t, \quad y(1) = 0$$

$$x(t) = -t^2 + 2t, \quad y(t) = -2t + 2$$

Die Darstellung der Phasenkurve $(x(t), y(t))$ kann wegen der quadratischen Parabel für $x(t)$ in Abhängigkeit von x durch die beiden Zweige $y_{1,2}(x) = \mp 2\sqrt{1-x}$ oder in impliziter Form $f(x, y) = |y| - 2\sqrt{1-x} = 0$ erfolgen. Wegen der Linearität von $y(t)$ kann in Abhängigkeit von y die Darstellung $x(y) = 1 - y^2/4$ gewählt werden.

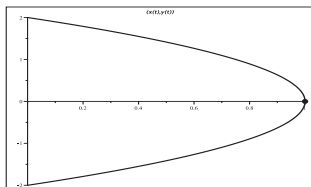


Abb. 2.247 Datei *sys_loes_021a.ps*, gDGI (2.1),
Lösung $x(t) = -t^2 + 2t$, $y(t) = -2t + 2$
als Phasenkurve $(x(t), y(t))$, $t \in [0, 2]$,
mit der AB $x(1) = 1$, $y(1) = 0$, $(x, y) \in [0, 1] \times [-2, 2]$,
Kommandos `dsolve`, `plot`, `pointplot`

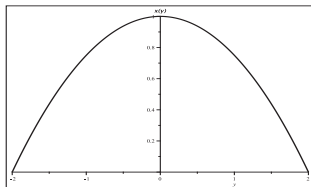


Abb. 2.248 Datei *sys_loes_021b.ps*, gDGI (2.1),
Lösung $x(t) = -t^2 + 2t$, $y(t) = -2t + 2$
als Kurve $x = 1 - y^2/4$, $x \in [-2, 2]$,
mit der AB $x(0) = 1$, $(y, x) \in [-2, 2] \times [0, 1]$,
Kommandos `dsolve`, `plot`, `pointplot`

Näherungslösungen berechnet man mit dem Kommando `DEplot`. Unter Verwendung der Option `scene` sind drei Kurven darstellbar.

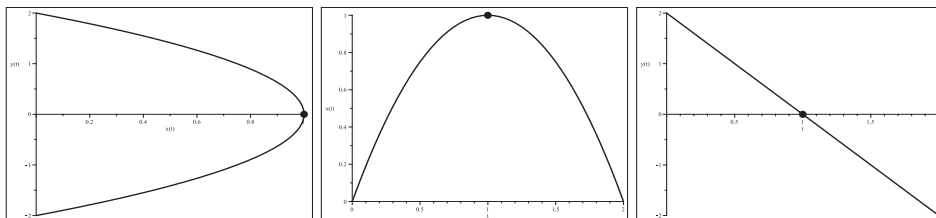


Abb. 2.249 Datei *sys_loes_021b1..3.ps*, gDGI (2.1),
Näherungslösung zu $x(t) = -t^2 + 2t$, $y(t) = -2t + 2$
als Phasenkurven $(x(t), y(t))$, $(t, x(t))$, $(t, y(t))$, $t \in [0, 2]$,
mit der AB $x(1) = 1$, $y(1) = 0$,
 $(x, y) \in [0, 1] \times [-2, 2]$, $[0, 2] \times [0, 1]$ bzw. $[0, 2] \times [-2, 2]$,
Kommandos `DEplot`, `pointplot`

$$(2.2) \quad x' = y + t, \quad y' = -2x/t^2 + 2y/t - 2$$

$$x(t) = -3t^2 \ln(t) + (c_1/2 + 5)t^2 + c_2 t, \quad y(t) = -6t(\ln(t) - 1) + c_1 t + c_2$$

bzw. als AWP

$$x' = y + t, \quad x(1) = 1,$$

$$y' = -2x/t^2 + 2y/t - 2, \quad y(1) = 0$$

$$x(t) = -3t^2(\ln(t) - 1) - 2t, \quad y(t) = -2t(3\ln(t) - 1) - 2$$

Die Darstellung der Phasenkurve $(x(t), y(t))$ ist nur für $t \geq 0$ möglich.

```
> dgls2:=diff(x(t),t)=y(t)+t,
      diff(y(t),t)=-2*x(t)/t^2+2*y(t)/t-2:
vars:=x(t),y(t):
> inits2:=x(1)=1,y(1)=0;
r1:=dsolve([dgls2,inits2],[vars]);
xt:=rhs(op(1,r1)); yt:=rhs(op(2,r1));
> pl1:=plot([xt,yt,t=0..2],color=black,thickness=3,title='(x(t),y(t))',
            titlefont=[HELVETICA,bold,10]):
pl2:=pointplot([[1,0]],color=black,symbol=solidcircle,symbolsize=20):
display(pl1,pl2);
```

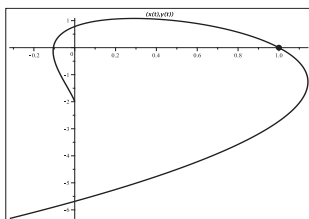


Abb. 2.250 Datei *sys_Loes_022a.ps*, gDGl (2.2),

Lösung

$$x(t) = -3t^2(\ln(t) - 1) - 2t, \quad y(t) = -2t(3\ln(t) - 1) - 2$$

als Phasenkurve $(x(t), y(t))$, $t \in [0, 2]$,

mit der AB $x(1) = 1$, $y(1) = 0$, $(x, y) \in [-0.4, 1.2] \times [-7, 2]$,

Kommandos `dsolve`, `plot`, `pointplot`

Näherungslösungen berechnet man wieder mit dem Kommando `DEplot`. Unter Verwendung der Option `scene` werden drei Kurven erzeugt.

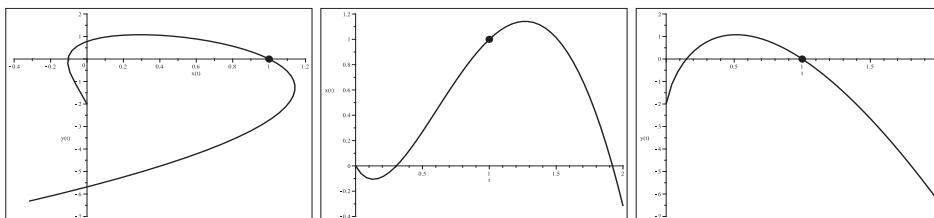


Abb. 2.251 Datei *sys_Loes_022b1..3.ps*, gDGl (2.2),

Näherungslösung zu $x(t) = -3t^2(\ln(t) - 1) - 2t$, $y(t) = -2t(3\ln(t) - 1) - 2$
als Phasenkurven $(x(t), y(t))$, $(t, x(t))$, $(t, y(t))$, $t \in [0, 2]$,

mit der AB $x(1) = 1$, $y(1) = 0$,

$(x, y) \in [-0.4, 1.2] \times [-7, 2]$, $[0, 2] \times [-0.4, 1.2]$ bzw. $[0, 2] \times [-7, 2]$,

Kommandos `DEplot`, `pointplot`

(3) Räuber-Beute-Modelle, nichtlineare autonome SysgDGI

Die folgenden Modelle basieren auf dem gleichen SysgDGI mit den Funktionen $y(t)$ als Beute und $z(t)$ als Räuber. Es ändern sich nur die AB in den AWP.

$$y' = ay - bzy = y(a - bz), \quad z' = -cz + dzy = -z(c - dy)$$

Für Populationen betrachtet man den Fall $y(t), z(t) \geq 0$, aber auch andere Situationen sind theoretisch möglich.

Maple findet mittels `dsolve` die spezielle einfache Lösung $y(t) = c_1 e^{at}$, $z(t) = 0$, dazu gehört die Nulllösung, sowie eine zweite Formel, die wegen einem Integral, `RootOf` und `LambertW` nicht explizit notiert werden kann. Es gibt aber die zweite einfache Lösung $y(t) = 0$, $z(t) = c_2 e^{-ct}$. Eine weitere stationäre Lösung ist $y(t) = \frac{c}{d}$, $z(t) = \frac{a}{b}$.

$$(3.1) \quad \begin{aligned} y' &= y(a - bz), & y(t_0) &= y_0, & \text{spez. } y(0) &= 6000, \\ z' &= -z(c - dy), & z(t_0) &= z_0, & z(0) &= 30 \end{aligned}$$

Für das AWP ergibt sich in Maple eine sehr komplizierte Lösungsdarstellung. Nur für einige spezielle Parametersituationen und AB wird eine Lösung berechnet. In den meisten Fällen wird keine explizite Lösung erzeugt.

Das Richtungsfeld zum Phasenraum des SysgDGI mit den Parametern

$$a = \frac{1}{5}, \quad b = \frac{1}{500}, \quad c = \frac{1}{10}, \quad d = \frac{1}{100000}$$

macht die Vielfalt der möglichen Phasenkurven deutlich.

So darf man stationäre, einfache, fallende, wachsende, asymptotisch verlaufende, gegen Null tendierende, von Null wegwärts laufende oder oszillierende Lösungen vermuten und auch erwarten.

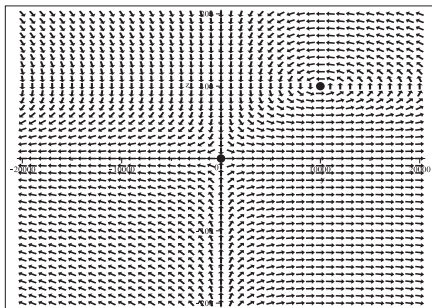


Abb. 2.252

Datei `sys_Loes_031a.ps`,

Richtungsfeld zum Phasenraum

vom SysgDGI (3.1) mit den Parametern

$$a = \frac{1}{5}, \quad b = \frac{1}{500}, \quad c = \frac{1}{10}, \quad d = \frac{1}{100000}$$

in $[-20000, 20000] \times [-200, 200]$,

Kommandos `dfieldplot`, `pointplot`

Im Richtungsfeld (y', z') des Phasenraums (y, z) erkennt man horizontale Richtungselemente bei $z = 0$, vertikale bei $y = 0$, stationäre Punkte bei $(0, 0)$ und $(10000, 100)$, Phasenkurven $(y(t), z(t))$ mit asymptotisch gegen $z = 0$ fallenden Verläufen in der Halbebene $z < 0$ sowie als geschlossene Kurven im ersten Quadranten, die auf periodisches und oszillierendes Verhalten der Koordinatenfunktionen verweisen.

Welche der AWP können mit dem Kommando `dsolve` gelöst werden?

Wir zeigen einige Fälle.

```
> a:='a': b:='b': c:='c': d:='d':
  sys1:=diff(y(t),t)=a*y(t)-b*y(t)*z(t),
        diff(z(t),t)=-c*z(t)+d*y(t)*z(t):
  vars:=y(t),z(t):
> # die spezielle Loesung {y(t)=c1*exp(at), z(t)=0} und
  # eine andere (nicht auswertbar)
  dsolve([sys1],[vars]);
> # sehr umfangreiche und komplizierte Loesungsdarstellung
  inits1:=y(t0)=y0,z(t0)=z0:
  dsolve([sys1,inits1],[vars]);
> # Loesung {y(t)=2000*exp(at), z(t)=0}
  inits13:=y(0)=2000,z(0)=0:
  dsolve([sys1,inits13],[vars]);
> # keine Loesung
  a:=1/5:
  dsolve([sys1,inits13],[vars]);
  a:='a':
```

Jetzt definieren wir die Parameter und lösen das SysgDGI bzw. das AWP.

```
> # Parameter
  a:=1/5: b:=1/500: c:=1/10: d:=1/100000:

> # die spezielle Loesung {y(t)=c1*exp(t/5), z(t)=0} und
  # eine andere (nicht auswertbar)
  dsolve([sys1],[vars]);
> # sehr umfangreiche und komplizierte Loesungsdarstellung
  inits1:=y(t0)=y0,z(t0)=z0:
  dsolve([sys1,inits1],[vars]);

> # keine Loesung, zu kompliziert
  inits10:=y(0)=y0,z(0)=z0:
  dsolve([sys1,inits10],[vars]);
> # keine Loesung
  inits13:=y(0)=2000,z(0)=0:      # inits13:=y(0)=-2000,z(0)=0:
  dsolve([sys1,inits13],[vars]);
> # keine Loesung
  inits14:=y(0)=0,z(0)=100:      # inits14:=y(0)=0,z(0)=-100:
  dsolve([sys1,inits14],[vars]);

> # stationaere Loesung {y(t)=10000, z(t)=100}
  inits12:=y(0)=10000,z(0)=100:
  dsolve([sys1,inits12],[vars]);
> # stationaere Loesung {y(t)=0, z(t)=0}
  inits15:=y(0)=0,z(0)=0:
  dsolve([sys1,inits15],[vars]);
```

Also bleibt nur die numerische Lösung von AWP mittels `dsolve/numeric` oder `DEplot` mit dem Verweis auf die Erläuterungen im Abschnitt 2.2.

Mit der Option `output=listprocedure` wird eine Liste von Prozeduren erzeugt, deren Elemente als einzelne Prozeduren weiter verarbeitet werden können.

Verschiedene Möglichkeiten der Ergebniserzeugung und -verarbeitung

`subs`, `eval`, `assign`

```
> Digits:=16:
  inits11:=y(0)=6000,z(0)=30:
  lsg1:=dsolve([sys1,inits11],[vars],type=numeric,output=listprocedure);

> tf:=titlefont=[COURIER,bold,10]:
  pl2:=pointplot([[0,60],[0,30]],color=black,symbol=solidcircle,
                 symbolsize=24):

> # Koordinatenfunktionen
> ts:=subs(lsg1,t): ys:=subs(lsg1,y(t)): zs:=subs(lsg1,z(t)):
  pl1:=plot([ys(t)/100,zs(t)],t=0..102,color=black,linestyle=[solid,dash],
            thickness=3,tf,
            title='solid: ys(t)/100 = Beute/100\n dash: zs(t) = Raeuber '):
  display(pl1,pl2);

> te:=eval(t,lsg1): ye:=eval(y(t),lsg1): ze:=eval(z(t),lsg1):
  pl1:=plot([ye(t)/100,ze(t)],t=0..102,color=black,linestyle=[solid,dash],
            thickness=3,tf,
            title='solid: ye(t)/100 = Beute/100\n dash: ze(t) = Raeuber '):
  display(pl1,pl2);

> assign(subs(y(t)=ya,z(t)=za,lsg1));
  pl1:=plot([ya/100,za],0..102,color=black,linestyle=[solid,dash],
            thickness=3,tf,
            title='solid: y(t)/100 = Beute/100\n dash: z(t) = Raeuber '):
  display(pl1,pl2);

> # Phasenkurve
  pl1:=plot([ya/100,za,0..51],color=black,labels=['y(t)/100','z(t)'],
            thickness=3,
            view=[0..350,0..250]):
  pl2:=pointplot([60,30],color=black,symbol=solidcircle,symbolsize=24):
  display(pl1,pl2);
```

Die drei Kurvenplots stimmen überein.

Mit der letzten Version erfolgt die Ausgabe der Näherungen der einzelnen Koordinatenfunktionen $y(t)$, $z(t)$ sowie der Phasenkurve $(y(t), z(t))$. Man erkennt das oszillierende Verhalten der Funktionen und die Periodenlänge $T \approx 51$. Zur besseren Darstellung wurde für die Funktion $y(t)$ eine Skalierung durchgeführt.

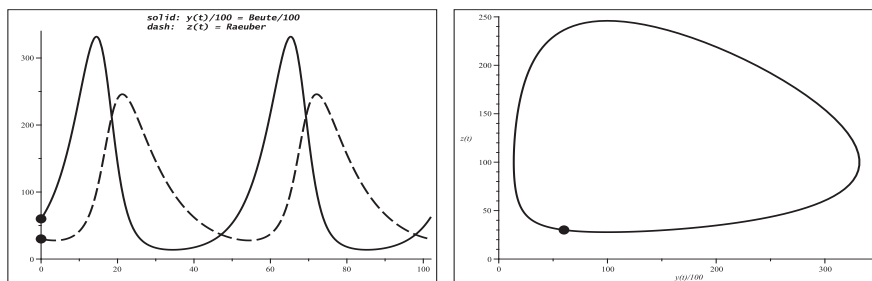


Abb. 2.253 Dateien *sys_loes_031b1, 2.ps*,

SysgDgl (3.1) mit den Parametern $a = \frac{1}{5}$, $b = \frac{1}{500}$, $c = \frac{1}{10}$, $d = \frac{1}{100000}$,
 Näherungen der Koordinatenfunkt. $y(t)/100$, $z(t)$ in $[0, 102] \times [0, 340]$,
 und der Phasenkurve $(y(t)/100, z(t))$ in $[0, 350] \times [0, 250]$,
 Kommandos `dsolve/numeric`, `plot`, `pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie mehrere verschiedene Typen von Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. Falls die Näherungskurven während der Berechnung auf die Koordinatenachsen “treffen“, d.h. sehr nahe sind, dann werden sie im weiteren Verlauf auch dort bleiben.

```
> pts:=[ [0,6000,30], [0,6000,-30], [0,2000,0], [0,-2000,0],
          [0,0,100], [0,0,-100], [0,0,0], [0,10000,100], [0,500,-200],
          [0,8000,60], [0,-4000,200], [0,-4000,-200] ]:
p14:=DEplot([sys1],[vars],t=0..51,y=-25000..35000,z=-250..250,pts,
            dirgrid=[61,51],color=black,method=rkf45,stepsize=0.05,
            scene=[y(t),z(t)],linecolor=black):
p14a:=pointplot([ [6000,30], [6000,-30], [2000,0], [-2000,0],
                  [0,100], [0,-100], [0,0], [10000,100], [500,-200],
                  [8000,60], [-4000,200], [-4000,-200] ],
               color=black,symbol=solidcircle,symbolsize=20):
display(p14,p14a);
```

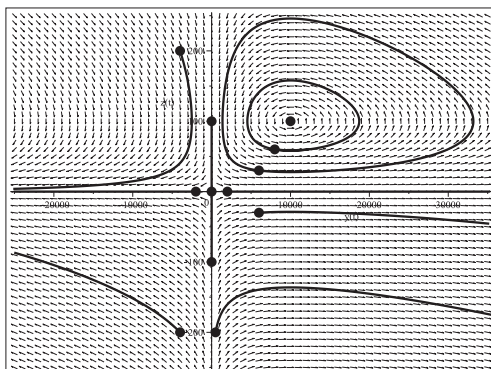


Abb. 2.254

Datei *sys_loes_031c.ps*,
 Richtungsfeld zum Phasenraum vom
 SysgDgl (3.1) mit den Parametern
 $a = \frac{1}{5}$, $b = \frac{1}{500}$, $c = \frac{1}{10}$, $d = \frac{1}{100000}$,
 Näherungslösungen zu $(y(t), z(t))$
 mit verschiedenen AB
 in $[-25000, 35000] \times [-250, 250]$,
 Kommandos `DEplot`, `pointplot`

$$(3.2) \quad \begin{aligned} y' &= y(a - bz), & y(t_0) &= y_0, & \text{spez. } y(0) &= 10000, \\ z' &= -z(c - dy), & z(t_0) &= z_0, & z(0) &= 20 \end{aligned}$$

Die Parameter sind

$$a = \frac{1}{7}, \quad b = \frac{1}{600}, \quad c = \frac{1}{20}, \quad d = \frac{1}{10000}.$$

Es gibt viele Analogien zum SysgDGI (3.1).

Die stationären Punkte sind bei $(0, 0)$ und $(500, 600/7)$. Erst mit einer feineren Auflösung des Richtungsfeldes lässt sich der zweite erkennen.

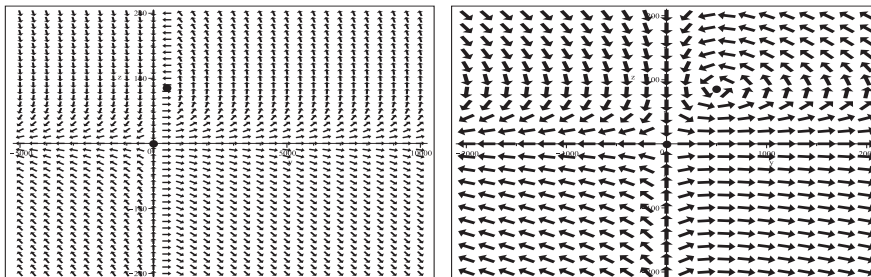


Abb. 2.255 Dateien *sys_Loes_032a1, 2.ps*,

Richtungsfeld zum Phasenraum vom SysgDGI (3.2) mit den Parametern

$$a = \frac{1}{7}, \quad b = \frac{1}{600}, \quad c = \frac{1}{20}, \quad d = \frac{1}{10000}$$

in $[-5000, 10000] \times [-200, 200]$ bzw. $[-2000, 2000] \times [-200, 200]$,

Kommandos `dfieldplot`, `pointplot`

Die mit der AB $y(0) = 10000, z(0) = 20$ ausgewählte Näherung zur Phasenkurve $(y(t), z(t))$ zeigt nach dem anfänglichen “Einschwingen“ einen asymptotischen Verlauf und tendiert mit wachsender Zeit t gegen Null. Sie liegt dann also auf der z -Achse.

Zur besseren Darstellung wurde für die Funktion $y(t)$ eine Skalierung durchgeführt.

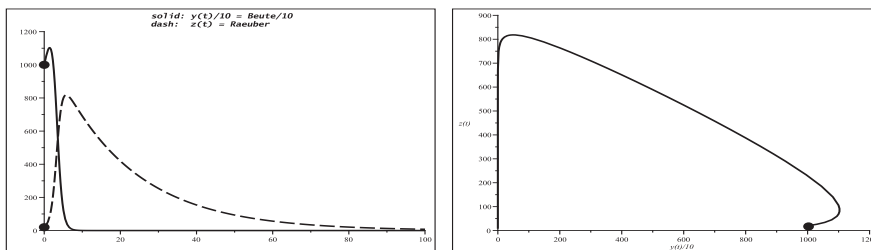


Abb. 2.256 Dateien *sys_Loes_032b1, 2.ps*,

SysgDGI (3.2) mit den Parametern $a = \frac{1}{7}, b = \frac{1}{600}, c = \frac{1}{20}, d = \frac{1}{10000}$,

Näherungen der Koordinatenfunkt. $y(t)/10, z(t)$ in $[0, 100] \times [0, 1200]$,

und der Phasenkurve $(y(t)/10, z(t))$ in $[0, 1200] \times [0, 900]$,

Kommandos `dsolve/numeric`, `plot`, `pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie mehrere verschiedene Typen von Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. Falls die Näherungskurven während der Berechnung auf die z -Achse "treffen", d.h. dieser hinreichend nahe kommen, dann werden sie im weiteren Verlauf auch dort bleiben.

```
> pts:=[ [0,10000,20], [0,10000,-20], [0,2000,0], [0,-2000,0],
          [0,0,500], [0,0,-500], [0,0,0], [0,500,600/7], [0,500,-600/7],
          [0,2000,80], [0,-5000,500], [0,-5000,-500], [0,4000,80], [0,4000,-80]] :
> pts1:=[ [0,10000,20], [0,4000,80], [0,2000,80], [0,500,600/7]] :
```

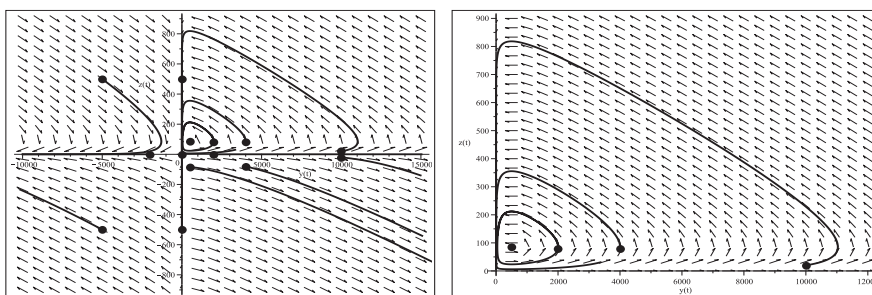


Abb. 2.257 Dateien *sys_Loes_032c1, 2.ps*,

Richtungsfeld zum Phasenraum vom SysgDGI (3.2) mit den Parametern

$$a = \frac{1}{7}, \quad b = \frac{1}{600}, \quad c = \frac{1}{20}, \quad d = \frac{1}{10000},$$

Näherungslösungen (Phasenkurven) zu obigen AB mit $t \in [0, 115]$
in $[-10000, 15000] \times [-900, 900]$ bzw. $[0, 12000] \times [0, 900]$,

Kommandos `DEplot`, `pointplot`

$$(3.3) \quad \begin{aligned} y' &= y(a - bz), & y(t_0) &= y_0, & \text{spez. } y(0) &= 6000, \\ z' &= -z(c - dy), & z(t_0) &= z_0, & z(0) &= 30 \end{aligned}$$

Die Parameter sind

$$a = \frac{1}{6}, \quad b = \frac{1}{600}, \quad c = \frac{1}{60}, \quad d = \frac{1}{600000}.$$

Das Beispiel ist wiederum ähnlich zum SysgDGI (3.1).

Die stationären Punkte sind bei $(0, 0)$ und $(10000, 100)$.

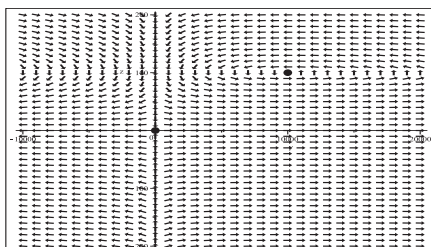


Abb. 2.258

Datei *sys_Loes_033a.ps*,

Richtungsfeld zum Phasenraum vom SysgDGI (3.3) mit den Parametern

$$a = \frac{1}{6}, \quad b = \frac{1}{600}, \quad c = \frac{1}{60}, \quad d = \frac{1}{600000}$$

in $[-10000, 20000] \times [-200, 200]$,

Kommandos `dfieldplot`, `pointplot`

Die mit der AB $y(0) = 6000$, $z(0) = 30$ gewählte Näherung zur Phasenkurve $(y(t), z(t))$ zeigt einen periodischen Verlauf und hat die Periode $T \approx 177$.

Zur besseren Darstellung wurde für die Funktion $y(t)$ eine Skalierung durchgeführt.

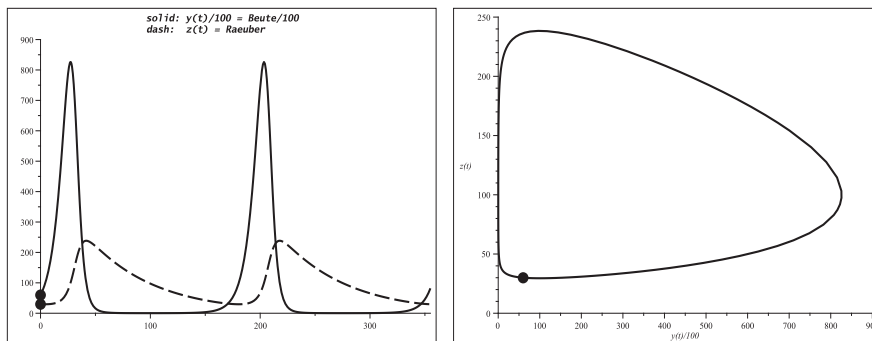


Abb. 2.259 Dateien *sys_Loes_033b1, 2.ps*,

SysgDGl (3.3) mit den Parametern $a = \frac{1}{6}$, $b = \frac{1}{600}$, $c = \frac{1}{60}$, $d = \frac{1}{600000}$,
 Näherungen der Koordinatenfunkt. $y(t)/100$, $z(t)$ in $[0, 355] \times [0, 900]$,
 und der Phasenkurve $(y(t)/100, z(t))$ in $[0, 900] \times [0, 250]$,
 Kommandos `dsolve/numeric`, `plot`, `pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie mehrere verschiedene Typen von Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. Obwohl einige Näherungskurven während der Berechnung der z -Achse sehr nahe kommen, werden sie im weiteren Verlauf nicht von ihr "geschluckt".

```
> pts:= [[0, 6000, 30], [0, 6000, -30], [0, 10000, 0], [0, -10000, 0], [0, 0, 100],
         [0, 0, -100], [0, 0, 0], [0, 10000, 100], [0, 10000, -100], [0, 5000, 100],
         [0, -5000, 100], [0, -1000, -140], [0, -10000, 180], [0, 60000, 100]]:
```

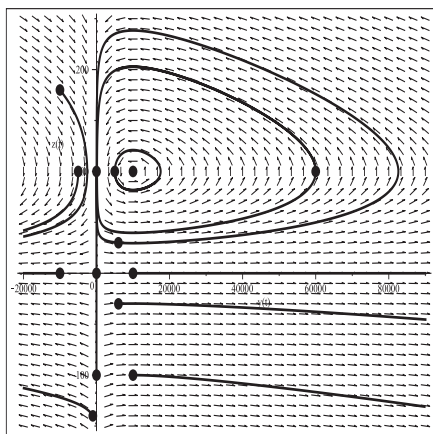


Abb. 2.260

Datei *sys_Loes_033c.ps*,
 Richtungsfeld zum Phasenraum vom
 SysgDGl (3.3) mit den Parametern
 $a = \frac{1}{6}$, $b = \frac{1}{600}$, $c = \frac{1}{60}$, $d = \frac{1}{600000}$,
 Näherungslösungen (Phasenkurven)
 zu obigen AB mit $t \in [0, 175]$
 in $[-20000, 90000] \times [-150, 250]$,
 Kommandos `DEplot`, `pointplot`

$$(3.4) \quad \begin{aligned} y' &= y(a - bz), & y(t_0) &= y_0, & \text{spez. } y(0) &= 50, \\ z' &= -z(c - dy), & z(t_0) &= z_0, & z(0) &= 20 \end{aligned}$$

Im Vergleich zu den bisherigen Beispielen sind hier die Parameter moderat.

$$a = \frac{4}{5}, \quad b = \frac{1}{50}, \quad c = \frac{1}{50}, \quad d = \frac{1}{250}.$$

Das Beispiel ist wiederum ähnlich zum SysgDGL (3.1). Eine Skalierung der Funktionen ist jedoch hier nicht erforderlich. Die stationären Punkte sind bei $(0, 0)$ und $(5, 40)$.

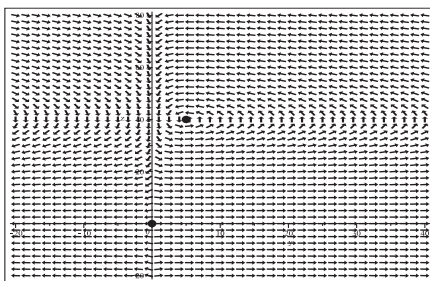


Abb. 2.261

Datei `sys_Loes_034a.ps`,

Richtungsfeld zum Phasenraum vom SysgDGL (3.4) mit den Parametern

$$a = \frac{4}{5}, \quad b = \frac{1}{50}, \quad c = \frac{1}{50}, \quad d = \frac{1}{250}$$

in $[-20, 40] \times [-20, 80]$,

Kommandos `dfieldplot`, `pointplot`

Die mit der AB $y(0) = 50, z(0) = 20$ gewählte Näherung zur Phasenkurve $(y(t), z(t))$ zeigt einen periodischen Verlauf und hat die Periode $T \approx 102$.

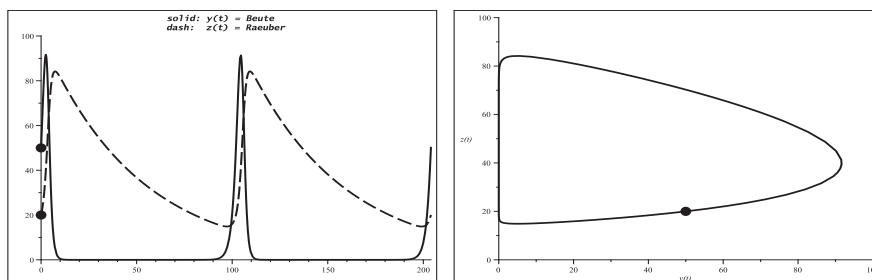


Abb. 2.262 Dateien `sys_Loes_034b1, 2.ps`,

SysgDGL (3.4) mit den Parametern $a = \frac{4}{5}, b = \frac{1}{50}, c = \frac{1}{50}, d = \frac{1}{250}$,

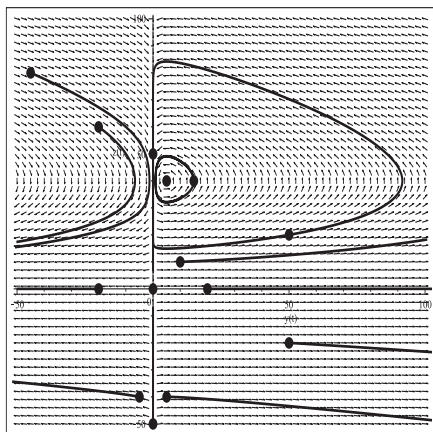
Näherungen der Koordinatenfunkt. $y(t), z(t)$ in $[0, 204] \times [0, 100]$,

und der Phasenkurve $(y(t), z(t))$ in $[0, 100] \times [0, 100]$,

Kommandos `dsolve/numeric`, `plot`, `pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie mehrere verschiedene Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. Obwohl einige Näherungskurven während der Berechnung der z -Achse sehr nahe kommen, werden sie im weiteren Verlauf nicht von ihr "geschluckt".

```
> pts:=[[0,50,20],[0,50,-20],[0,20,0],[0,-20,0],
        [0,0,50],[0,0,-50],[0,0,0],[0,5,40],[0,15,40],
        [0,10,10],[0,-20,60],[0,-45,80],[0,-5,-40],[0,5,-40]]:
```

**Abb. 2.263**

Datei *sys_Joes_034c.ps*,

Richtungsfeld zum Phasenraum vom SysgDGI (3.4) mit den Parametern

$$a = \frac{4}{5}, \quad b = \frac{1}{50}, \quad c = \frac{1}{50}, \quad d = \frac{1}{250},$$

Näherungslösungen (Phasenkurven)

zu obigen AB mit $t \in [0, 102]$

in $[-50, 100] \times [-50, 100]$,

Kommandos `DEplot`, `pointplot`

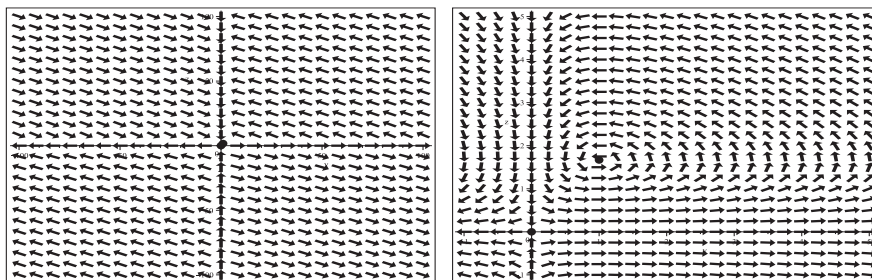
$$(3.5) \quad \begin{aligned} y' &= y(a - bz), & y(t_0) &= y_0, & \text{spez. } y(0) &= 6000, \\ z' &= -z(c - dy), & z(t_0) &= z_0, & z(0) &= 30 \end{aligned}$$

Auch hier sind die Parameter moderat.

$$a = \frac{1}{3}, \quad b = \frac{1}{5}, \quad c = \frac{1}{10}, \quad d = \frac{1}{10}.$$

Für dieses Beispiel sind die stationären Punkte bei $(0, 0)$ und $(1, \frac{5}{3})$.

Das Verhalten in der Nähe dieser Punkte kann nur mit der Auswahl eines entsprechenden Bereiches demonstriert werden. Bei Wahl der AB $y(0) = 6000$, $z(0) = 30$ kann beim Zeichnen der Phasenkurve die Situation um den Nullpunkt selber nicht aufgelöst und dargestellt werden.

**Abb. 2.264** Dateien *sys_Joes_035a1, 2.ps*,

Richtungsfeld zum Phasenraum vom SysgDGI (3.5) mit den Parametern

$$a = \frac{1}{3}, \quad b = \frac{1}{5}, \quad c = \frac{1}{10}, \quad d = \frac{1}{10}$$

in $[-100, 100] \times [-100, 100]$ bzw. $[-1, 5] \times [-1, 5]$,

Kommandos `dfieldplot`, `pointplot`

Die mit der AB $y(0) = 6000$, $z(0) = 30$ berechnete Näherung zur Phasenkurve $(y(t), z(t))$ zeigt einen asymptotischen Verlauf gegen den Nullpunkt.

Die Koordinatenfunktionen $y(t)$, $z(t)$ unterliegen in einem sehr kurzen anfänglichem Abschnitt $[0, 0.02]$ starken, man kann sagen sprunghaften Veränderungen. Man nennt diesen Abschnitt auch transiente Phase. Danach kommt die lange asymptotische Phase. Erst die Zoomtechnik verdeutlicht dieses Anfangsverhalten.

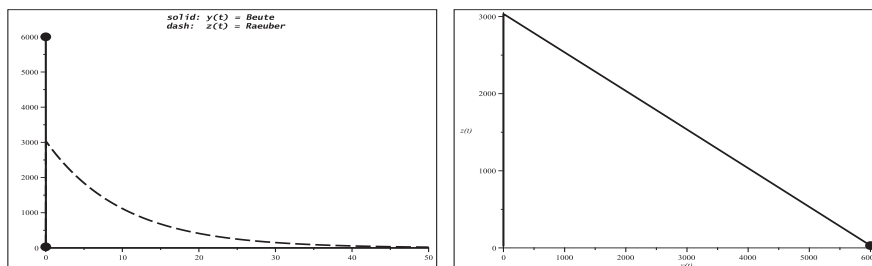


Abb. 2.265 Dateien *sys_Loes_035b1*, 3.ps,

SysgDGI (3.5) mit den Parametern $a = \frac{1}{3}$, $b = \frac{1}{5}$, $c = \frac{1}{10}$, $d = \frac{1}{10}$,
 Näherungen der Koordinatenfunkt. $y(t)$, $z(t)$ in $[0, 50] \times [0, 6000]$,
 und der Phasenkurve $(y(t), z(t))$ in $[0, 6000] \times [0, 3050]$,
 Kommandos `dsolve/numeric`, `plot`, `pointplot`

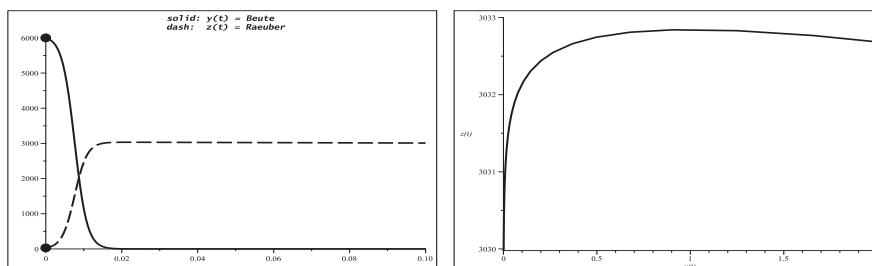


Abb. 2.266 Dateien *sys_Loes_035b2*, 4.ps,

SysgDGI (3.5) mit den Parametern $a = \frac{1}{3}$, $b = \frac{1}{5}$, $c = \frac{1}{10}$, $d = \frac{1}{10}$,
 Näherungen der Koordinatenfunkt. $y(t)$, $z(t)$ in $[0, 0.1] \times [0, 6000]$,
 und der Phasenkurve $(y(t), z(t))$ in $[0, 2] \times [3030, 3033]$,
 Kommandos `dsolve/numeric`, `plot`, `pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie mehrere verschiedene Typen von Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. Wenn einige Näherungskurven während der Berechnung der z-Achse sehr nahe kommen, werden sie im weiteren Verlauf von ihr “geschluckt“.

```
> # Phasenkurven um den stationaeren Punkt (1,5/3) bei pts1 nicht sichtbar
pts1:=[[0,6000,30],[0,1,5/3],[0,100,1],[0,1000,1000]]:
> pts:=[[0,1,1],[0,1,-1],[0,-1,1],[0,-1,-1],[0,3,1],
        [0,-0.2,-4],[0,0.2,-4],[0,-3,4],[0,-0.2,-4],
        [0,0,0],[0,1,5/3],[0,2,0],[0,-2,0],[0,0,2],[0,0,-2]]:
```

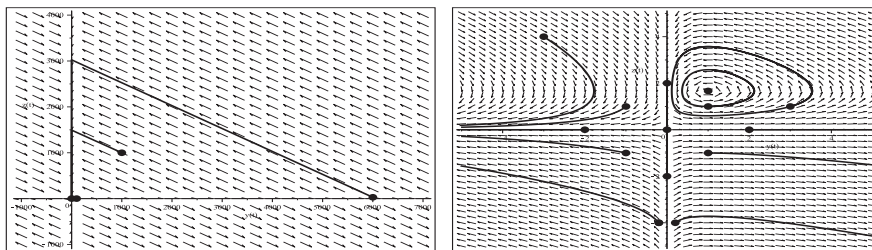


Abb. 2.267 Dateien *sys_Loes_035c1, 2.ps*,
 Richtungsfeld zum Phasenraum vom SysgDGI (3.5) mit den Parametern
 $a = \frac{1}{7}$, $b = \frac{1}{600}$, $c = \frac{1}{20}$, $d = \frac{1}{10000}$,
 Näherungslösungen (Phasenkurven) zu obigen AB mit $t \in [0, 50]$,
 pts1 in $[-1000, 7000] \times [-1000, 4000]$ bzw. pts $[-5, 5] \times [-5, 5]$,
 Kommandos `Dfieldplot`, `pointplot`

(4) Räuber-Beute-Modell, nichtlineares autonomes SysgDGI

$$y' = a(1 - dy)y - byz = y[a(1 - dy) - bz], \quad z' = -cz + dyz = -z(c - dy)$$

Daraus erkennt man die stationären Lösungen $(0, 0)$, $(\frac{1}{d}, 0)$ und $(\frac{c}{d}, \frac{a(1-c)}{b})$. Maple findet mittels `dsolve` die spezielle Lösung $y(t) = \frac{1}{d+c_1e^{-at}}$, $z(t) = 0$, dazu gehört nicht die Nulllösung, sowie eine zweite Formel, die nicht explizit aufgelöst werden kann. Es gibt aber die zweite einfache Lösung $y(t) = 0$, $z(t) = c_2 e^{-ct}$. Die Notation als AWP ist

$$\begin{aligned} y' &= y[a(1 - dy) - bz], & y(t_0) &= y_0, & \text{spez. } y(0) &= 20000, \\ z' &= -z(c - dy), & z(t_0) &= z_0, & z(0) &= 30. \end{aligned}$$

Die Parameter sind $a = \frac{1}{5}$, $b = \frac{1}{500}$, $c = \frac{1}{10}$, $d = \frac{1}{100000}$.

Die stationären Punkte bei $(0, 0)$ und $(10000, 90)$ liegen in der ‘‘Nachbarschaft‘‘, während $(100000, 0)$ erst mit einem entsprechenden Bereich des Richtungsfeldes erfassbar ist.

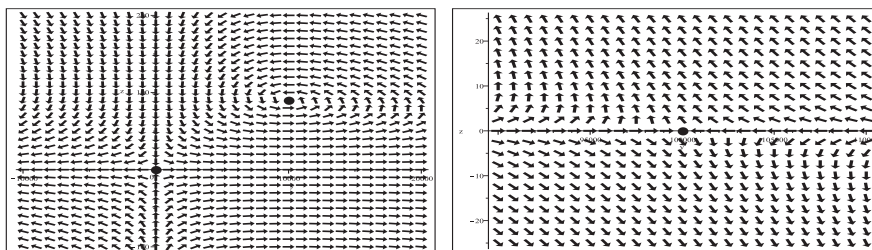


Abb. 2.268 Dateien *sys_Loes_04a1, 2.ps*,
 Richtungsfeld zum Phasenraum vom SysgDGI (4) mit den Parametern
 $a = \frac{1}{5}$, $b = \frac{1}{500}$, $c = \frac{1}{10}$, $d = \frac{1}{100000}$
 in $[-10000, 20000] \times [-100, 200]$ bzw. $[90000, 120000] \times [-25, 25]$,
 Kommandos `dfieldplot`, `pointplot`

Die mit der AB $y(0) = 20000, z(0) = 30$ ausgewählte Näherung zur Phasenkurve $(y(t), z(t))$ zeigt eine gedämpfte Schwingung und tendiert mit wachsender Zeit t gegen den stationären Punkt $(10000, 90)$.

Zur besseren Darstellung wurde für die Funktion $y(t)$ eine Skalierung durchgeführt.

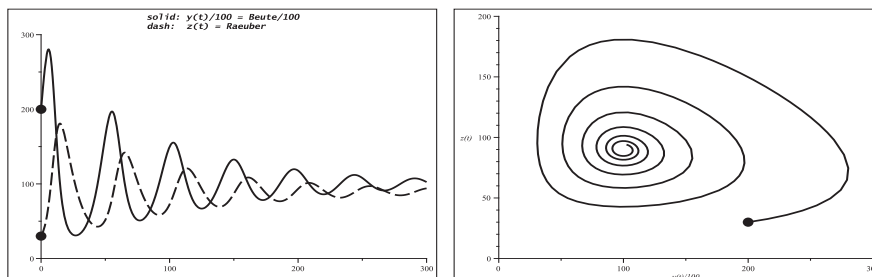


Abb. 2.269 Dateien *sys_Loes_04b1, 2.ps*,

SysgDGL (4) mit den Parametern $a = \frac{1}{5}, b = \frac{1}{500}, c = \frac{1}{10}, d = \frac{1}{100000}$,

Näherungen der Koordinatenfunkt. $y(t)/100, z(t)$ in $[0, 300] \times [0, 300]$,
und der Phasenkurve $(y(t)/100, z(t))$ in $[0, 300] \times [0, 200]$,

Kommandos `dsolve/numeric, plot, pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie mehrere verschiedene Typen von Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. Falls die Näherungskurven während der Berechnung auf die z -Achse "treffen", d.h. dieser hinreichend nahe kommen, dann werden sie im weiteren Verlauf auch dort bleiben.

```
> pts:=[ [0,20000,30], [0,0,0], [0,10000,90], [0,100000,0], [0,40000,20],
          [0,40000,-20], [0,110000,20], [0,110000,-20], [0,1000,200],
          [0,1000,-100], [0,-5000,150], [0,-5000,-100], [0,60000,10],
          [0,10000,0], [0,-10000,0], [0,0,50], [0,0,-50] ]:
```

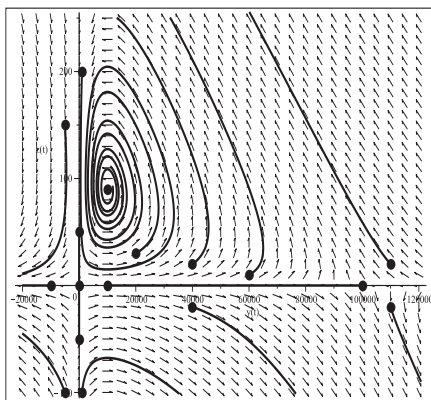


Abb. 2.270

Datei *sys_Loes_04c.ps*,

Richtungsfeld zum Phasenraum vom
SysgDGL (4) mit den Parametern

$a = \frac{1}{5}, b = \frac{1}{500}, c = \frac{1}{10}, d = \frac{1}{100000}$,

Näherungslösungen (Phasenkurven)

zu obigen AB mit $t \in [0, 200]$

in $[-20000, 120000] \times [-100, 250]$,

Kommandos `DEplot, method=rkf45,`

`stepsize=0.05, pointplot`

(5) Räuber-Beute-Modell, nichtlineares autonomes SysgDGI

$$\begin{aligned} y' &= a y (1 - z), & y(t_0) &= y_0, & \text{spez. } y(0) &= 6000, \\ z' &= -c z (1 - y), & z(t_0) &= z_0, & z(0) &= 30 \end{aligned}$$

Die Parameter sind $a = 1$, $c = 3$.

Die stationären Punkte sind bei $(0, 0)$ und $(1, 1)$. Maple findet mittels `dsolve` die spezielle Lösung $y(t) = c_1 e^{at}$, $z(t) = 0$, dazu gehört die Nulllösung, sowie eine zweite Formel, die wegen einem Integral, `RootOf` und `LambertW` nicht explizit notiert werden kann.

Es gibt aber die zweite einfache Lösung $y(t) = 0$, $z(t) = c_2 e^{-ct}$.

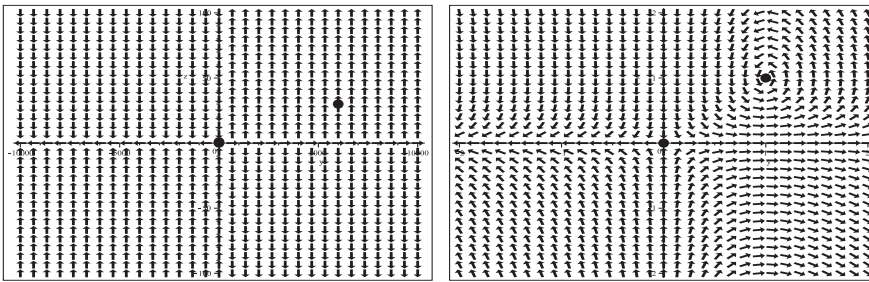


Abb. 2.271 Dateien *sys_Loes_05a1, 2.ps*,

Richtungsfeld zum Phasenraum vom SysgDGI (5) mit den Parametern $a = 1$, $c = 3$ in $[-10000, 10000] \times [-100, 100]$ bzw. $[-2, 2] \times [-2, 2]$,
Kommandos `dfieldplot`, `pointplot`

Die mit der AB $y(0) = 6000$, $z(0) = 30$ ausgewählte Näherung zur Phasenkurve $(y(t), z(t))$ zeigt nach dem anfänglichen “Einschwingen“ einen asymptotischen Verlauf und tendiert mit wachsender Zeit t gegen Null. Sie liegt dann also auf der z -Achse.

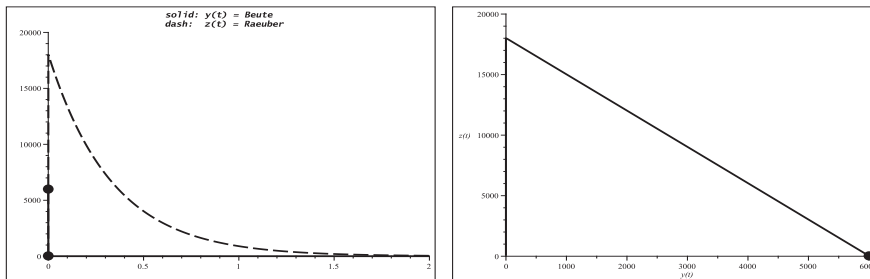


Abb. 2.272 Dateien *sys_Loes_05b1, 3.ps*,

SysgDGI (5) mit den Parametern $a = 1$, $c = 3$,
Näherungen der Koordinatenfunkt. $y(t)$, $z(t)$ in $[0, 2] \times [0, 20000]$,
und der Phasenkurve $(y(t), z(t))$ in $[0, 6000] \times [0, 20000]$,
Kommandos `dsolve/numeric`, `plot`, `pointplot`

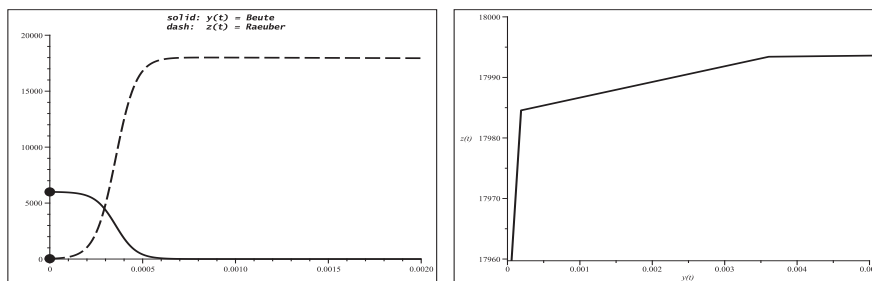


Abb. 2.273 Dateien *sys_loes_05b2, 4.ps*,
 SysgDGL (5) mit den Parametern $a = 1$, $c = 3$,
 Näherungen der Koordinatenfunkt. $y(t)$, $z(t)$ in $[0, 0.002] \times [0, 20000]$,
 und der Phasenkurve $(y(t), z(t))$ in $[0, 0.005] \times [17960, 18000]$,
 Kommandos `dsolve/numeric`, `plot`, `pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie mehrere verschiedene Typen von Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. Falls die Näherungskurven während der Berechnung auf die z -Achse “treffen“, d.h. dieser hinreichend nahe kommen, dann werden sie im weiteren Verlauf auch dort bleiben.

```
> # Phasenkurven um den stationaeren Punkt (1,1) bei pts1 nicht sichtbar
pts1:=[[0,6000,30],[0,3000,1000],[0,2000,0],[0,0,0],[0,1,1]]:
> pts:=[[0,4,1],[0,4,-1],[0,2,0],[0,-2,0],[0,2,1],
        [0,0,5],[0,0,-5],[0,0,0],[0,1,1],[0,0.5,-8],[0,-0.5,-8],
        [0,7,5],[0,-4,8],[0,-4,-8]]:
```

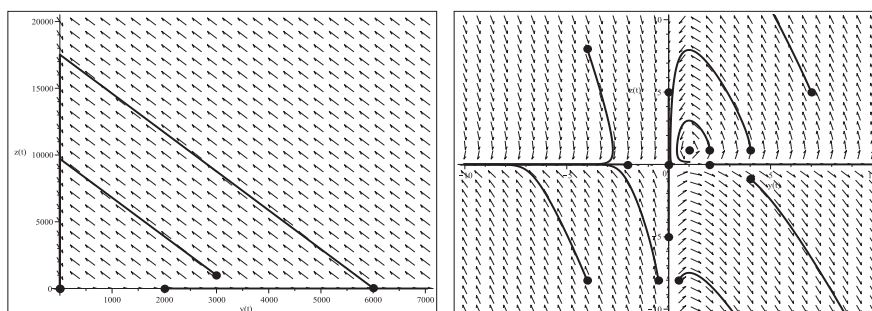


Abb. 2.274 Dateien *sys_loes_05c1, 2.ps*,
 Richtungsfeld zum Phasenraum vom SysgDGL (5)
 mit den Parametern $a = 1$, $c = 3$,
 Näherungslösungen (Phasenkurven) zu obigen AB `pts1` bzw. `pts`
 mit $t \in [0, 3]$ in $[0, 7000] \times [0, 20000]$ bzw. $[-10, 10] \times [-10, 10]$,
 Kommandos `DEplot`, `method=rkf45`, `stepsize=0.01`, `pointplot`

(6) Ressourcen-Modell

Es beschreibt die Konkurrenz zweier Arten $y(t)$ und $z(t)$, die sich aus derselben beschränkten Ressource ernähren.

$$\begin{aligned}y' &= y[-a_1 + c_1(1 - b_1y - b_2z)] = y[c_1 - a_1 - c_1(b_1y + b_2z)] = y[k_1 - c_1(b_1y + b_2z)], \\z' &= y[-a_2 + c_2(1 - b_1y - b_2z)] = y[c_2 - a_2 - c_2(b_1y + b_2z)] = y[k_2 - c_2(b_1y + b_2z)].\end{aligned}$$

Die Parameter sind positiv und es gilt i. Allg. $c_1 > a_1$, $c_2 > a_2$.

Ein stationärer Punkt ist bei $(0, 0)$. Weitere spezielle Punkte sind möglicherweise mit der Auflösung der Ausdrücke in den eckigen Klammern verbunden. Maple findet mittels `dsolve` keine Lösung.

Wir betrachten drei AWP mit folgenden Parameterkonstellationen.

$$(6.1) \quad b_1 = b_2 = 1, \quad d = 1 - \frac{a_1}{c_1} = 1 - \frac{a_2}{c_2}$$

Damit erhält man die Geradengleichung $d - (y + z) = 0$ und für alle Punkte (y, z) auf dieser Geraden ist $y' = z' = 0$, so dass dort keine Richtungselemente definiert werden können ("0/0").

$$\begin{aligned}y' &= y[-a_1 + c_1(1 - b_1y - b_2z)] = y[c_1 - a_1 - c_1(y + z)], \quad y(0) = 0.1, \\z' &= y[-a_2 + c_2(1 - b_1y - b_2z)] = y[c_2 - a_2 - c_2(y + z)], \quad z(0) = 0.2.\end{aligned}$$

Die anderen Parameter sind $c_1 = \frac{1}{5}$, $c_2 = \frac{1}{50}$, $a_1 = \frac{1}{10}$, $a_2 = \frac{1}{100}$.

`dsolve` liefert ohne die AB die spezielle Lösung $y(t) = \frac{1}{2+ce^{-t/10}}$, $z(t) = 0$, sowie eine zweite Formel, die wegen einem Integral und `RootOf` nicht explizit notiert werden kann.

Die Geradengleichung ohne Richtungselemente ist $z = \frac{1}{2} - y$.

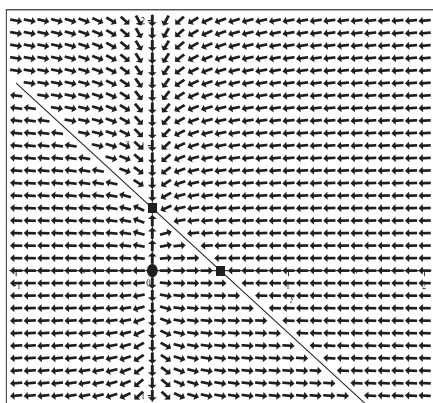


Abb. 2.275

Datei `sys_Joes_06a1.ps`,

Richtungsfeld zum Phasenraum vom

SysgDGl (6.1) mit den Parametern

$b_1 = b_2 = 1$, $c_1 = \frac{1}{5}$, $c_2 = \frac{1}{50}$, $a_1 = \frac{1}{10}$, $a_2 = \frac{1}{100}$
in $[-1, 2] \times [-1, 2]$,

Kommandos `dfieldplot`, `plot`, `pointplot`

Die mit der AB $y(0) = 0.1$, $z(0) = 0.2$ gewählte Näherung zur Phasenkurve $(y(t), z(t))$ zeigt einen asymptotischen Verlauf und "trifft" mit wachsendem t auf einen Punkt der genannten Gerade $y + z = \frac{1}{2}$, hier nahe $(0.278, 0.222)$. Lösungen zu AB, die sich auf der Geraden befinden, bleiben dort stehen.

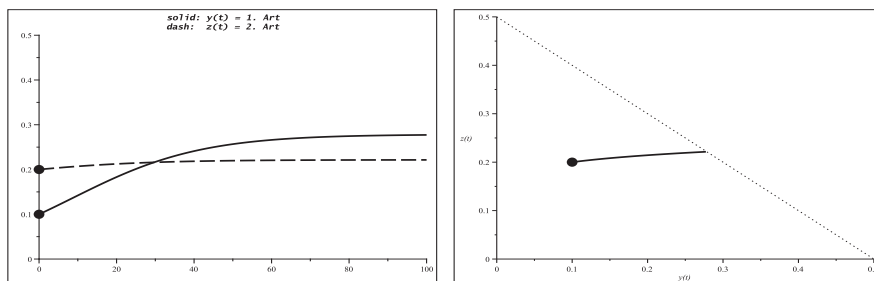


Abb. 2.276 Dateien *sys_Loes_06b11, 12.ps*, SysgDGl (6.1)

mit den Parametern $b_1 = b_2 = 1$, $c_1 = \frac{1}{5}$, $c_2 = \frac{1}{50}$, $a_1 = \frac{1}{10}$, $a_2 = \frac{1}{100}$,

Näherungen der Koordinatenfunkt. $y(t)$, $z(t)$ in $[0, 100] \times [0, 0.5]$,

und der Phasenkurve $(y(t), z(t))$ in $[0, 0.5] \times [0, 0.5]$,

Kommandos `dsolve/numeric`, `plot`, `pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie verschiedene Kurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. Kurven können sich von den Koordinatenachsen entfernen oder auf sie zulaufen. Viele Kurven “stoßen“ auf die Gerade, andere gehen “wegwärts“. AB auf der Geraden können zwar angegeben werden, aber auch die numerische Lösung bleibt in diesem Punkt.

```
> pts1:=[[0,0.1,0.2], [0,0,0], [0,0.5,0], [0,0,0.5]]:
> pts:=[[0,0.1,0.2], [0,0,0], [0,0.3,0], [0,0,0.3], [0,-0.05,0.5], [0,-0.2,0.9],
        [0,-0.05,0.2], [0,-0.05,-0.2], [0,0.1,-0.2], [0,0.9,0.5], [0,0.9,-0.2],
        [0,0.5,0.9], [0,0.5,-0.4], [0,0,-0.3], [0,0.8,0], [0,-0.3,0]]:
```

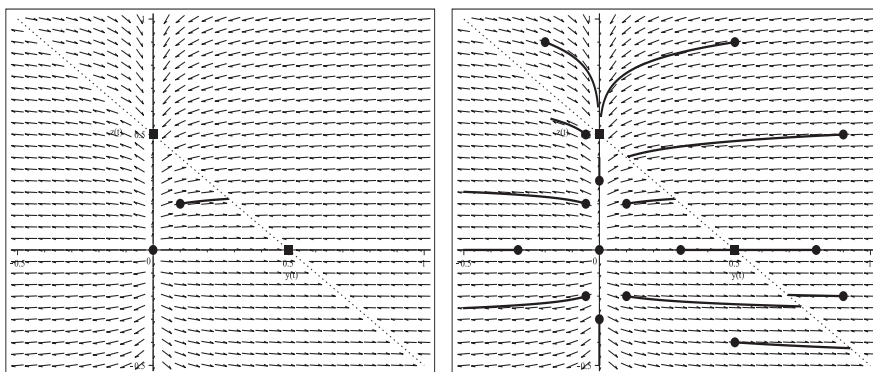


Abb. 2.277 Dateien *sys_Loes_06c11, 12.ps*,

Richtungsfeld zum Phasenraum vom SysgDGl (6.1) mit Parametern

$b_1 = b_2 = 1$, $c_1 = \frac{1}{5}$, $c_2 = \frac{1}{50}$, $a_1 = \frac{1}{10}$, $a_2 = \frac{1}{100}$,

Näherungslösungen (Phasenkurven) zu obigen AB mit $t \in [0, 100]$,

`pts1` bzw. `pts` in $[-0.5, 1] \times [-0.5, 1]$,

Kommandos `DEplot`, `method=rkf45`, `stepsize=0.05`, `plot`, `pointplot`

$$(6.2) \quad b_1 = b_2 = 1, \quad d_1 = 1 - \frac{a_1}{c_1} \neq 1 - \frac{a_2}{c_2} = d_2$$

Damit erhält man zwei Geradengleichungen $d_1 - (y + z) = 0$, $d_2 - (y + z) = 0$.

Für die Punkte (y, z) auf diesen Geraden ist $y' = 0$ oder $z' = 0$, so dass in den beiden speziellen Fällen $y = d_1$, $z = 0$ und $z = d_2$, $y = 0$ dort keine Richtungselemente definiert werden können ("0/0").

Für $y = 0$, $z = d_1$ ergibt sich ein vertikales Richtungselement, allgemein bis auf den Spezialfall vertikale auf der Geraden $d_1 - (y + z) = 0$, wo $y' = 0$ und $z' \neq 0$ werden (" $z'/0 = \pm\infty$ "). Für $y = d_2$, $z = 0$ ergibt sich ein horizontales Richtungselement, allgemein bis auf den Spezialfall horizontale auf der Geraden $d_2 - (y + z) = 0$, wo $y' \neq 0$ und $z' = 0$ sind (" $0/y' = 0$ ").

$$\begin{aligned} y' &= y[-a_1 + c_1(1 - b_1y - b_2z)] = c_1y[1 - \frac{a_1}{c_1} - (y + z)], & y(0) &= 0.1, \\ z' &= y[-a_2 + c_2(1 - b_1y - b_2z)] = c_2y[1 - \frac{a_2}{c_2} - (y + z)], & z(0) &= 0.2. \end{aligned}$$

Die anderen Parameter sind $c_1 = \frac{1}{5}$, $c_2 = \frac{1}{50}$, $a_1 = \frac{1}{10}$, $a_2 = \frac{1}{1000}$.

`dsolve` liefert ohne die AB die spezielle Lösung $y(t) = \frac{1}{2+ce^{-t/10}}$, $z(t) = 0$, sowie eine zweite Formel, die keine explizite Lösung darstellt.

Die Geradengleichungen sind $z = \frac{1}{2} - y$ und $z = \frac{19}{20} - y$.

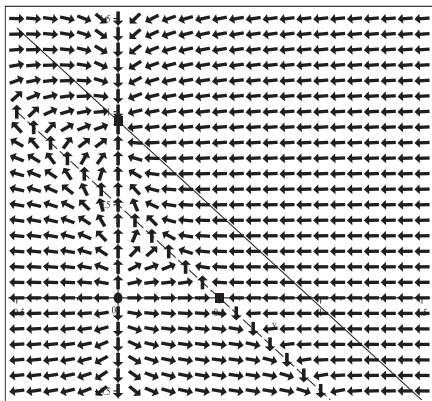


Abb. 2.278

Datei `sys_Joes_06a2.ps`,

Richtungsfeld zum Phasenraum vom

SysgDGL (6.2) mit den Parametern

$$b_1 = b_2 = 1, \quad c_1 = \frac{1}{5}, \quad c_2 = \frac{1}{50}, \quad a_1 = \frac{1}{10}, \quad a_2 = \frac{1}{1000}$$

in $[-0.5, 1.5] \times [-0.5, 1.5]$,

Kommandos `dfieldplot`, `plot`, `pointplot`

Die mit der AB $y(0) = 0.1$, $z(0) = 0.2$ gewählte Näherung zur Phasenkurve $(y(t), z(t))$ zeigt einen asymptotischen Verlauf und "trifft" mit wachsendem t bei $(0, d_2) = (0, 0.95)$ auf die zweite Gerade.

Es folgen weiterhin das Richtungsfeld zum Phasenraum sowie mehrere Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. AB liefern bis auf die Spezialfälle $y = d_1$, $z = 0$ und $z = d_2$, $y = 0$, wo keine Lösung herausführt, überall moderate Kurvenverläufe. Viele Phasenkurven "stoßen" auf den speziellen Punkt $(0, d_2)$.

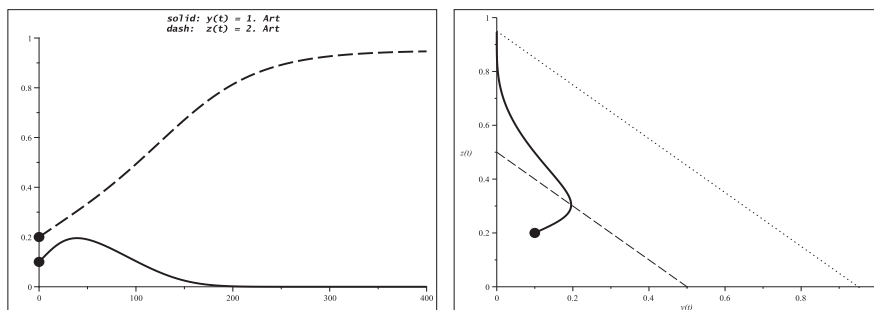


Abb. 2.279 Dateien *sys_loes_06b21, 22.ps*, SysgDGI (6.2)

mit den Parametern $b_1 = b_2 = 1$, $c_1 = \frac{1}{5}$, $c_2 = \frac{1}{50}$, $a_1 = \frac{1}{10}$, $a_2 = \frac{1}{1000}$,
 Näherungen der Koordinatenfunkt. $y(t), z(t)$ in $[0, 400] \times [0, 1]$,
 und der Phasenkurve $(y(t), z(t))$ in $[0, 1] \times [0, 1]$,
 Kommandos `dsolve/numeric`, `plot`, `pointplot`

```
> pts1:=[[0,0.1,0.2],[0,0,0],[0,0.5,0],[0,0,0.5],[0,0.95,0],[0,0,0.95]]:
> pts:=[[0,0.1,0.2],[0,0,0],[0,0.3,0],[0,0,0.3],[0,-0.1,0.5],
        [0,-0.05,0.2],[0,-0.05,-0.2],[0,0.1,-0.2],
        [0,1.2,0.5],[0,1.2,-0.2],[0,0.5,0.9],[0,0.5,-0.4],
        [0,0,-0.3],[0,0.8,0],[0,-0.3,0],[0,-0.2,0.9],[0,-0.5,1],
        [0,0,1.4],[0,-0.4,1.3],[0,-0.1,0.4],[0,1.3,0.2],[0,1.3,1.3]]:
```

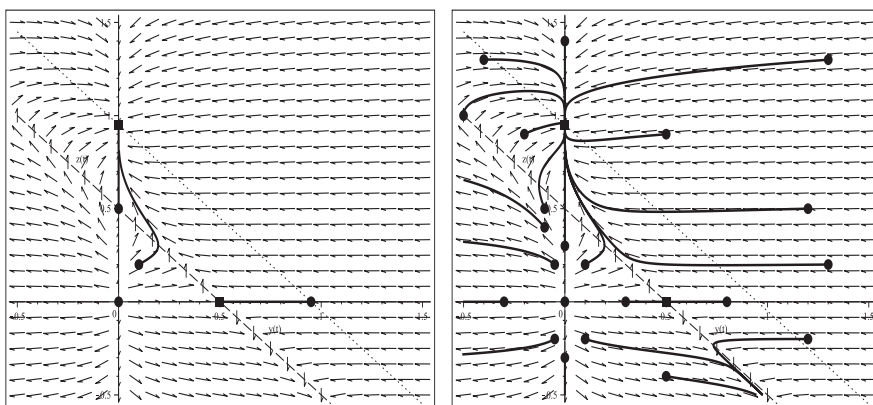


Abb. 2.280 Dateien *sys_loes_06c21, 22.ps*,

Richtungsfeld zum Phasenraum vom SysgDGI (6.2) mit Parametern
 $b_1 = b_2 = 1$, $c_1 = \frac{1}{5}$, $c_2 = \frac{1}{50}$, $a_1 = \frac{1}{10}$, $a_2 = \frac{1}{1000}$,
 Näherungslösungen (Phasenkurven) zu obigen AB mit $t \in [0, 200]$,
`pts1` bzw. `pts` in $[-0.5, 1.5] \times [-0.5, 1.5]$,

Kommandos `DEplot`, `method=rkf45`, `stepsize=0.05`, `plot`, `pointplot`

$$(6.3) \quad b_1 \neq b_2, \quad c_1 < a_1, \quad c_2 < a_2, \quad d_1 = 1 - \frac{a_1}{c_1} \neq 1 - \frac{a_2}{c_2} = d_2$$

Damit erhält man zwei Geradengleichungen $d_1 - (b_1y + b_2z) = 0$, $d_2 - (b_1y + b_2z) = 0$. Für die Punkte (y, z) auf diesen Geraden ist $y' = 0$ oder $z' = 0$, sodass in den beiden speziellen Fällen $y = \frac{d_1}{b_1}$, $z = 0$ und $z = \frac{d_2}{b_2}$, $y = 0$ dort keine Richtungselemente definiert werden können ("0/0").

Für $y = 0$, $z = \frac{d_1}{b_2}$ ergibt sich ein vertikales Richtungselement, allgemein bis auf den Spezialfall vertikale auf der Geraden $d_1 - (b_1y + b_2z) = 0$, wo $y' = 0$ und $z' \neq 0$ werden (" $z'/0 = \pm\infty$ "). Für $y = \frac{d_2}{b_1}$, $z = 0$ ergibt sich ein horizontales Richtungselement, allgemein bis auf den Spezialfall horizontale auf der Geraden $d_2 - (b_1y + b_2z) = 0$, wo $y' \neq 0$ und $z' = 0$ sind (" $0/y' = 0$ ").

$$\begin{aligned} y' &= y[-a_1 + c_1(1 - b_1y - b_2z)] = c_1 y [1 - \frac{a_1}{c_1} - (b_1y + b_2z)], & y(0) &= 0.5, \\ z' &= y[-a_2 + c_2(1 - b_1y - b_2z)] = c_2 y [1 - \frac{a_2}{c_2} - (b_1y + b_2z)], & z(0) &= -1. \end{aligned}$$

Die Parameter sind $b_1 = 2$, $b_2 = 3$, $c_1 = \frac{1}{10}$, $c_2 = \frac{1}{2000}$, $a_1 = \frac{1}{5}$, $a_2 = \frac{1}{500}$.

`dsolve` liefert keine Lösung. Die Geradengleichungen sind $z = -\frac{2}{3}y - \frac{1}{3}$ und $z = -\frac{2}{3}y - 1$.

Das Richtungsfeld hat wegen $c_2 = \frac{1}{2000}$ relativ flache Elemente.

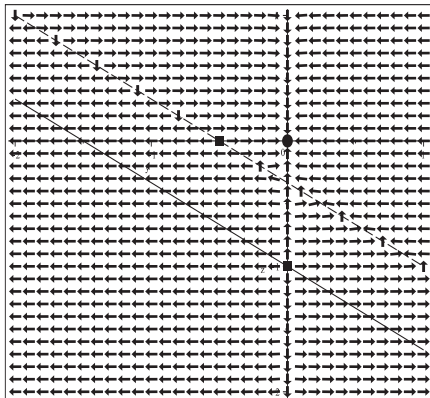


Abb. 2.281

Datei `sys_loes_06a3.ps`,

Richtungsfeld zum Phasenraum vom
SysgDgl (6.3) mit den Parametern

$$b_1 = 2, \quad b_2 = 3, \quad c_1 = \frac{1}{10}, \quad c_2 = \frac{1}{2000}, \quad a_1 = \frac{1}{5},$$

$$a_2 = \frac{1}{500} \text{ in } [-2, 1] \times [-2, 1],$$

Kommandos `dfieldplot`, `plot`, `pointplot`

Die mit der AB $y(0) = 0.5$, $z(0) = -1$ gewählte Näherung zur Phasenkurve $(y(t), z(t))$ zeigt zunächst einen flachen Verlauf und verläuft mit wachsendem t nahe $(0, \frac{d_1}{b_2}) = (0, -0.333)$, um schließlich auf $(0, 0)$ zu "treffen".

Einige Kurvenwerte sind

$$\begin{aligned} y(0) &= 0.5, & z(0) &= -1, \\ y(100) &= 0.8688, & z(100) &= -0.9073, \\ y(400) &= 0.5153, & z(400) &= -0.6704, \\ y(1000) &= 0.0765, & z(1000) &= -0.3644, \\ y(1100) &= 0.0370, & z(1100) &= -0.3285, \\ y(2000) &= 5.8e-9, & z(2000) &= -0.1121, \\ y(3000) &= 1.3e-8, & z(3000) &= -0.0274. \end{aligned}$$

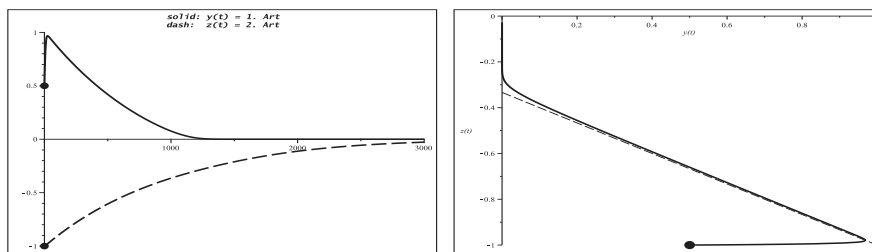


Abb. 2.282 Dateien *sys_loes_06b31*, *32.ps*, SysgDGI (6.3)

mit den Parametern $b_1 = 2$, $b_2 = 3$, $c_1 = \frac{1}{10}$, $c_2 = \frac{1}{2000}$, $a_1 = \frac{1}{5}$, $a_2 = \frac{1}{500}$,
 Näherungen der Koordinatenfunkt. $y(t)$, $z(t)$ in $[0, 3000] \times [-1, 1]$,
 und der Phasenkurve $(y(t), z(t))$ in $[0, 1] \times [-1, 0]$,
 Kommandos `dsolve/numeric`, `plot`, `pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie mehrere Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. AB liefern bis auf die Spezialfälle $y = \frac{d_1}{b_1}$, $z = 0$ und $z = \frac{d_2}{b_2}$, $y = 0$, wo keine Lösung herausführt, überall moderate Kurvenverläufe. Einige Phasenkurven “stoßen“ auf die stationäre Lösung $(0, 0)$.

```
> pts1:=[[0,0.5,-1],[0,0,0],[0,0,-1/3],[0,-3/2,0]]:
> pts:=[[0,0.5,-1],[0,0,0],[0,0,-1/3],[0,-1.5,0],[0,0.4,0],[0,-0.4,0],
        [0,0,1/3],[0,-1,0],[0,-1,1/3],[0,1,1/3],[0,-1,-1],[0,0.5,-1],
        [0,-0.5,-1.5],[0,0.5,-1/6],[0,-0.5,-0.5],[0,1/6,-1.5],
        [0,-1.5,0.7],[0,0,-1.5],[0,0.5,-0.6]]:
```

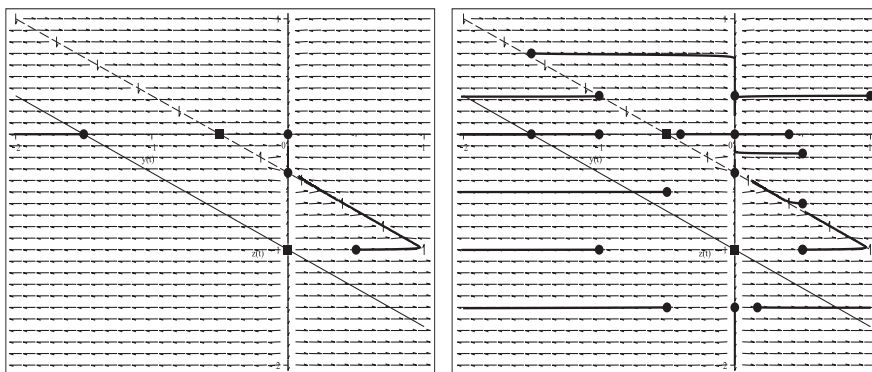


Abb. 2.283 Dateien *sys_loes_06c31*, *32.ps*,

Richtungsfeld zum Phasenraum vom SysgDGI (6.3) mit Parametern
 $b_1 = 2$, $b_2 = 3$, $c_1 = \frac{1}{10}$, $c_2 = \frac{1}{2000}$, $a_1 = \frac{1}{5}$, $a_2 = \frac{1}{500}$,
 Näherungslösungen (Phasenkurven) zu obigen AB mit
 $t \in [0, 1000]$ und `pts1` bzw. $t \in [0, 400]$ und `pts` in $[-2, 1] \times [-2, 1]$,
 Kommandos `DEplot`, `method=rkf45`, `stepsize=0.05`, `plot`, `pointplot`

(7) Lotka-Volterra-Modell

Das SysgDGI beschreibt die Konkurrenz zweier Arten $y(t)$ und $z(t)$ und ist ein Sonderfall des Räuber-Beute-Modells aus Teil (3) mit allen dortigen Informationen.

$$\begin{aligned} y' &= y(1-z), & y(t_0) &= y_0, & \text{spez. } y(0) &= 1.2, \\ z' &= -\frac{3}{10}z(1-y), & z(t_0) &= z_0, & z(0) &= 0.7 \end{aligned}$$

Die Parameter (vergl. Teil (3)) sind $a = b = 1$, $c = d = \frac{3}{10}$.

Die stationären Punkte liegen damit in $(0, 0)$ und $(1, 1)$.

Das Richtungsfeld zum Phasenraum hat i. Allg. moderate Anstiege.

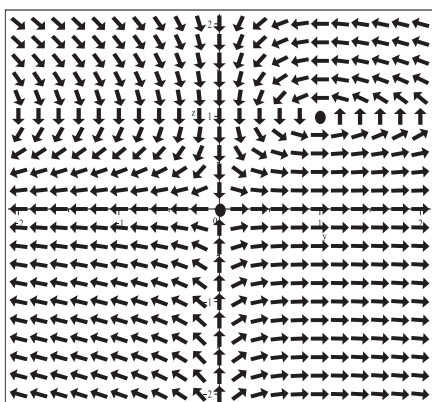


Abb. 2.284

Datei *sysLoes_07a.ps*,

Richtungsfeld zum Phasenraum vom

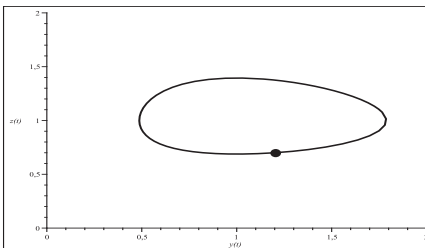
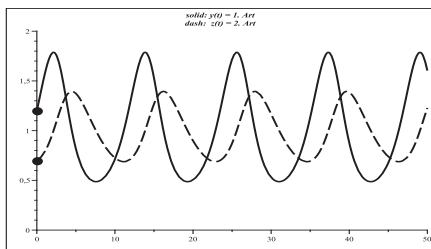
SysgDGI (7) mit den Parametern

$a = b = 1$, $c = d = \frac{3}{10}$

in $[-2, 2] \times [-2, 2]$,

Kommandos `dfieldplot`, `pointplot`

Die mit der AB $y(0) = 1.2$, $z(0) = 0.7$ gewählte Näherung zur Phasenkurve $(y(t), z(t))$ zeigt einen periodischen Verlauf mit der Periode $T \approx 11.73$.

Abb. 2.285 Dateien *sysLoes_07b1, 2.ps*, SysgDGI (7)

mit den Parametern $a = b = 1$, $c = d = \frac{3}{10}$,

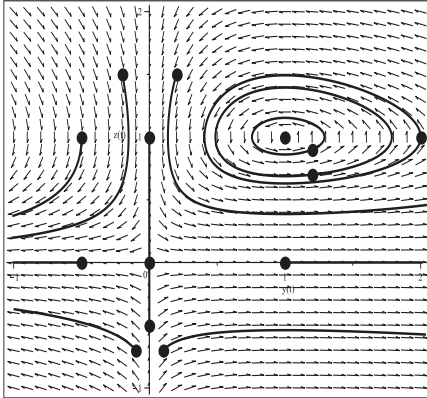
Näherungen der Koordinatenfunkt. $y(t)$, $z(t)$ in $[0, 50] \times [0, 2]$,

und der Phasenkurve $(y(t), z(t))$ in $[0, 2] \times [0, 2]$,

Kommandos `dsolve/numeric`, `plot`, `pointplot`

Es folgen das Richtungsfeld zum Phasenraum sowie mehrere Phasenkurven mittels `DEplot`. Wenn die AB auf den Koordinatenachsen liegen, dann auch die Näherungskurven. Aus den AB $(0, 0)$ und $(1, 1)$ führen keine Lösungen heraus.

```
> pts:=[[0,0,0],[0,1,1],[0,1.2,0.7],[0,1.2,0.9],[0,2,1],
        [0,1,0],[0,-0.5,0],[0,0,1],[0,0,-0.5],[0,-0.1,-0.7],
        [0,-0.5,1],[0,0.1,-0.7],[0,0.2,1.5],[0,-0.2,1.5]]:
```

**Abb. 2.286**

Datei *sys_Loes_07c.ps*,
 Richtungsfeld zum Phasenraum vom
 SysgDGl (7) mit den Parametern
 $a = b = 1$, $c = d = \frac{3}{10}$,
 Näherungslösungen (Phasenkurven)
 zu obigen AB mit $t \in [0, 12]$
 in $[-1, 2] \times [-1, 2]$,
 Kommandos `DEplot,method=rkf45,`
`stepsize=0.05, pointplot`

Phasenkurven als Phasenportraits um den stationären Punkt $(1, 1)$ sowie die Koordinatenfunktionen $y(t)$ und $z(t)$ werden mit dem Kommando `phaseportrait` erzeugt.

```
> lotka_volterra:=diff(y(t),t)=y(t)*(1-z(t)),
                    diff(z(t),t)=-3/10*z(t)*(1-y(t));
vars:=y(t),z(t):
tf:=titlefont=[COURIER,bold,11]:

> pts1:=[[y(0)=1.2,z(0)=0.7],[y(0)=1.0,z(0)=0.9],
        [y(0)=1.0,z(0)=0.6],[y(0)=1.0,z(0)=0.3]]:
pts2:=[[y(0)=1.0,z(0)=0.99],[y(0)=1.0,z(0)=0.9],[y(0)=1.0,z(0)=0.75],
        [y(0)=1.0,z(0)=0.5],[y(0)=1.0,z(0)=0.3],[y(0)=1.0,z(0)=0.2]]:

> T:=11.73:
p1:=phaseportrait([lotka_volterra],[vars],t=0..T,pts1, # pts2
                 method=rkf45,stepsize=0.1,view=[0..4.5,0..2.5],
                 linecolor=black,color=black,dirgrid=[30,30],
                 title='Lotka-Volterra-Modell - Phasenkurven',tf):
p2:=pointplot([[1,1],[1.2,0.7],[1.0,0.9],[1.0,0.6],[1.0,0.3]],
              color=black,symbol=solidcircle,symbolsize=20):
display(p1,p2);

> # Szenemenu fuer y(t) und z(t), 0<=t<=15, mit pts2
p3:=phaseportrait([lotka_volterra],[vars],t=0..15,pts2,
                 method=rkf45,stepsize=0.1,linecolor=black,
                 scene=[t,y(t)],view=[0..15,0..6],
                 # scene=[t,z(t)],view=[0..15,0..3],
                 title='Lotka-Volterra-Modell - Koordinatenkurven y(t)',tf):
p4:=pointplot([[0,1]],color=black,symbol=solidcircle,symbolsize=24):
display(p3,p4);
```

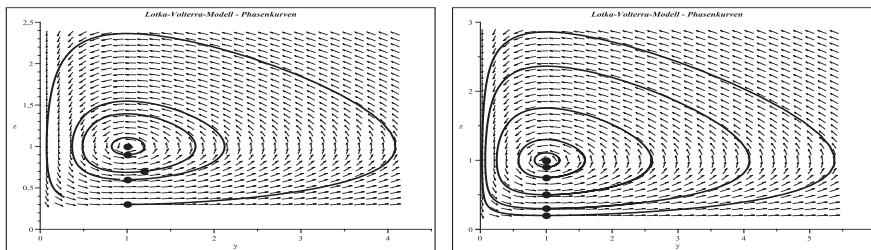


Abb. 2.287 Dateien *sysJoel_07d1, 2.ps*, SysgDGI (7),
 Richtungsfeld zum Phasenraum mit Parametern $a = b = 1$, $c = d = \frac{3}{10}$,
 Näherungslösungen (Phasenkurven) zu den AB
 pts1 für $t \in [0, T]$ in $[0, 4.5] \times [0, 2.5]$ bzw. pts2 für $t \in [0, 15]$ in $[0, 6] \times [0, 3]$,
 Kommandos `phaseportrait`, `pointplot`

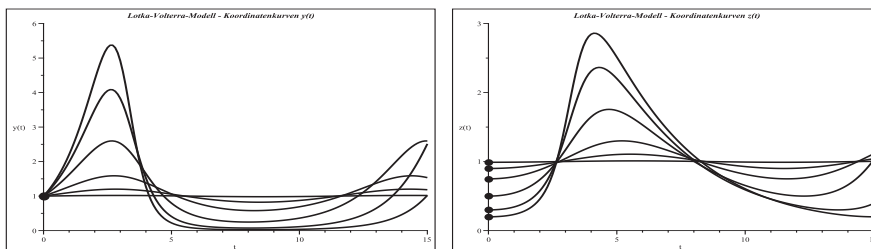


Abb. 2.288 Dateien *sysJoel_07d3, 4.ps*, SysgDGI (7),
 Parameter $a = b = 1$, $c = d = \frac{3}{10}$ und Szenemenü `scene`,
 Näherungslösungen (Koordinatenfunktionen) zu den AB pts2
 und $y(t)$ bzw. $z(t)$ für $t \in [0, 15]$,
 Kommandos `phaseportrait`, `pointplot`

(8) System mit drei gDGI 1. Ordnung

$$\begin{aligned} x' &= y - z, & x(t_0) &= x_0, & \text{spez. } y(0) &= 1, \\ y' &= z - x, & y(t_0) &= y_0, & y(0) &= 0, \\ z' &= x - 2y, & z(t_0) &= z_0, & z(0) &= 2 \end{aligned}$$

Mit dem Kommando `dsolve` ergeben sich für das SysgDGI bzw. AWP sehr lange Lösungsformeln. Ein Richtungsfeld kann nicht erzeugt werden.

Deshalb werden zunächst mit `dsolve/numeric` die Koordinatenfunktionen $x(t)$, $y(t)$, $z(t)$ näherungsweise ermittelt, anschließend mit `phaseportrait` verschiedene Phasenkurven und die Koordinatenfunktionen. Periodizität liegt nicht vor.

```
> dgls3:=D(x)(t)=y(t)-z(t), D(y)(t)=z(t)-x(t), D(z)(t)=x(t)-2*y(t);
vars:=x(t),y(t),z(t):
> inits3:=x(0)=1,y(0)=0,z(0)=2:
lsg3:=dsolve([dgls3,inits3],[vars],type=numeric,output=listprocedure);
```

```

> tf:=titlefont=[COURIER,bold,11]:
pts:=[[0,1],[0,0],[0,2]]:
ptsxy:=[1,0]: ptsxz:=[1,2]: ptsyz:=[0,2]:
pl2:=pointplot(pts,color=black,symbol=solidcircle,symbolsize=24):

> assign(subs(x(t)=xa,y(t)=ya,z(t)=za,lsg3));
> pl1:=plot([xa,ya,za],0..15,color=black,linestyle=[solid,dash,dashdot],
  thickness=3,view=[0..15,-8..8],
  title='solid: x(t), dash: y(t), dashdot: z(t)',tf):
display(pl1,pl2);

```

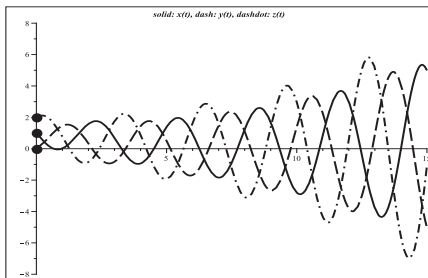


Abb. 2.289

Datei *sys_Loes_08b.ps*, SysgDGI (8),
 Näherungen der Koordinatenfunktionen
 $x(t)$, $y(t)$, $z(t)$ in $[0, 15] \times [-8, 8]$,
 AB $x(0) = 1$, $y(0) = 0$, $z(0) = 2$,
 Kommandos
`dsolve/numeric`, `plot`, `pointplot`

Vorgehensweise bei Erzeugung der Phasenkurven und Koordinatenfunktionen

```

> tt:=-4..4:
> pl1:=phaseportrait([dgl3],[vars],tt,[[inits3]],
  scene=[x(t),y(t)], # scene=[x(t),z(t)], scene=[y(t),z(t)],
  linecolor=black,method=rkf45,stepsize=0.05):
pl2:=pointplot(ptsxy,color=black,symbol=solidcircle,symbolsize=24):
display(pl1,pl2);

> tt:=-12..4: # tt:=-4..12: tt:=-4..4:
> pl1a:=phaseportrait([dgl3],[vars],tt,[[inits3]],
  scene=[t,x(t)], # scene=[t,y(t)], scene=[t,z(t)],
  linecolor=black,linestyle=solid,
  method=rkf45,stepsize=0.05):
display(pl1a,...);

```

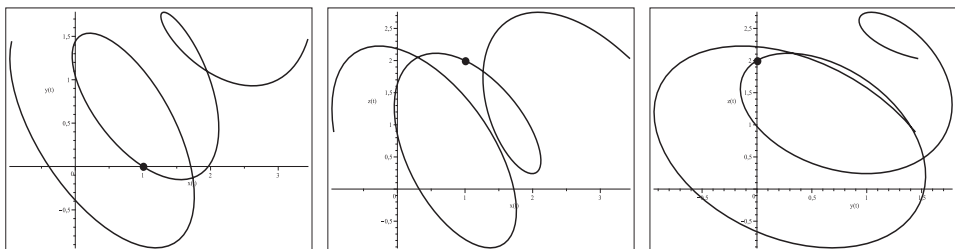


Abb. 2.290 Dateien *sys_Loes_08b1, 2, 3.ps*, SysgDGI (8), AB $x(0)=1$, $y(0)=0$, $z(0)=2$,
 Näherungen der Phasenkurven $(x(t), y(t))$, $(x(t), z(t))$, $(y(t), z(t))$
 für $t \in [-4, 4]$, Kommandos `phaseportrait`, `pointplot`

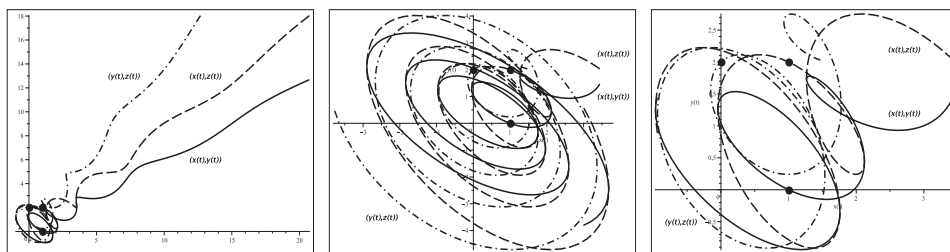


Abb. 2.291 Dateien *sys_Loes_08b4, 5, 6.ps*, SysgDGI (8), AB $x(0)=1, y(0)=0, z(0)=2$, Näherungen der Phasenkurven $(x(t), y(t)), (x(t), z(t)), (y(t), z(t))$, $t \in [-12, 4], [-4, 12]$ bzw. $[-4, 4]$,
Kommandos `phaseportrait`, `pointplot`

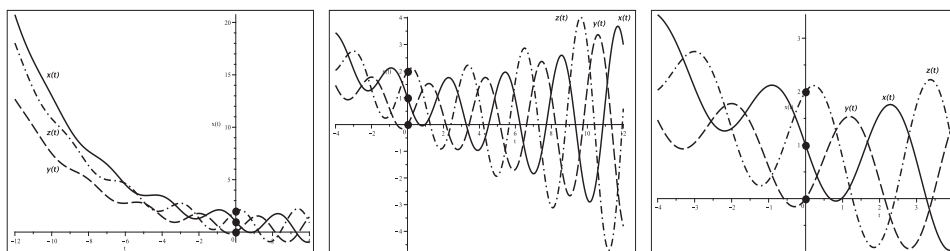


Abb. 2.292 Dateien *sys_Loes_08b7, 8, 9.ps*, SysgDGI (8), AB $x(0)=1, y(0)=0, z(0)=2$, Näherungen der Koordinatenfunktionen $x(t), y(t), z(t)$, $t \in [-12, 4], [-4, 12]$ bzw. $[-4, 4]$,
Kommandos `phaseportrait`, `pointplot`

(9) Van der Polsche DGI

Zunächst wird diesbezüglich auf den Abschnitt 1.5 verwiesen.

Die Van der Polsche DGI 2. Ordnung $x'' = \varepsilon(1 - x^2)x' - x$, $x(t_0) = x_0$, $x'(t_0) = x'_0$, transformiert man mittels $x' = y$ auf das SysgDGI und AWP

$$\begin{aligned} x' &= y, & x(t_0) &= x_0, \\ y' &= -x + \varepsilon(1 - x^2)y, & y(t_0) &= y_0. \end{aligned}$$

Wegen der Nichtlinearität kann man das SysgDGI bei $\varepsilon \neq 0$ nicht analytisch lösen (Ausnahme Nulllösung für Null-AB).

Der Parameter $\varepsilon \geq 0$ charakterisiert das Schwingungsverhalten der Lösung. Er bestimmt entscheidend die Zeitspanne des Anfangsverlaufs bis zur Phase der Oszillation der Trajektorie, Abschnitte langsamer oder schneller Bewegung, flache und steile Kurvenstücke, die Einschwingdauer in der oszillierenden Phase sowie die Länge der Periode. Je größer ε ist, desto ausgeprägter sind der Anfangsverlauf und steile Kurvenstücke, aber umso kürzer die Einschwingdauer. Bei $\varepsilon \gg 1$ (z.B. $\varepsilon = 100, 1000$) beschreibt die gDGI Relaxationsschwingungen. Dabei wechseln sich Zeiten sehr langsamer Bewegung mit solchen sehr schneller ab.

Gewissen Einfluss auf die Anfangsphase haben auch die AB, was für die dritte Variante der Tabelle in den Abb. 2.295, 2.298 und 2.299 deutlich wird.

Var.	ε	AB	Periode $T \approx T_k$	A-Phase, E-Dauer
1	0.5	$x(0) = 0, y(0) = 0.3$ $x(0) = 0, y(0) = 0, 0.3, 1, 2.5$	$T_1 = 6.38$	$\approx 2T_1, 2T_1$
2	1	$x(0) = 0, y(0) = -0.05$ $x(0) = 0, y(0) = 0, -0.05, 3, -1$	$T_2 = 6.67$	$\approx \frac{3}{2}T_2, T_2$
3	5	$x(0) = 2, y(0) = 0$ $x(0) = 0, 2, 1, 3, y(0) = 0$ $x(0) = 0, y(0) = 0, -2, 2, 4, 6$	$T_3 = 11.61$	$\approx \frac{1}{2}T_3, \frac{1}{2}T_3$

Tab. 2.3 Drei Testvarianten und ihre Charakteristika,
A-Phase = Anfangsphase, E-Dauer = Einschwingdauer

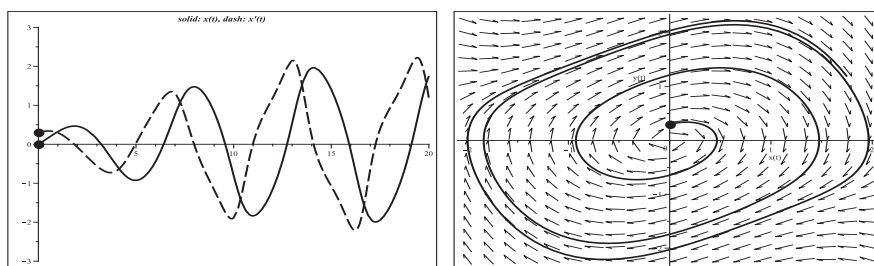


Abb. 2.293 Dateien *sys_Loes_09b11, 2.ps*, SysgDGl (9), $x(0)=0, y(0)=0.3, \varepsilon = 0.5$,
Näherung der Koordinatenfunktionen $x(t), y(t)$ in $[0, 20] \times [-3, 3]$,
Kommandos `dsolve/numeric, plot, pointplot`,
Näherung der Phasenkurve $(x(t), y(t))$ in $[-2, 2] \times [-2.2, 2.2]$,
Kommandos `phaseportrait\method=rkf45, stepsize=0.05, pointplot`

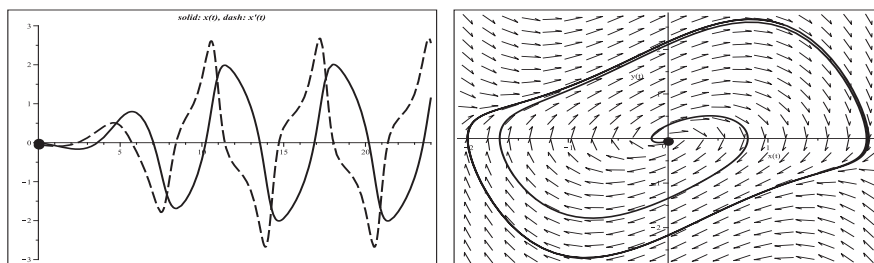


Abb. 2.294 Dateien *sys_Loes_09b21, 2.ps*, SysgDGl (9), $x(0)=0, y(0)=-0.05, \varepsilon = 1$,
Näherung der Koordinatenfunktionen $x(t), y(t)$ in $[0, 24] \times [-3, 3]$,
Kommandos `dsolve/numeric, plot, pointplot`,
Näherung der Phasenkurve $(x(t), y(t))$ in $[-2, 2] \times [-3, 3]$,
Kommandos `phaseportrait, pointplot`

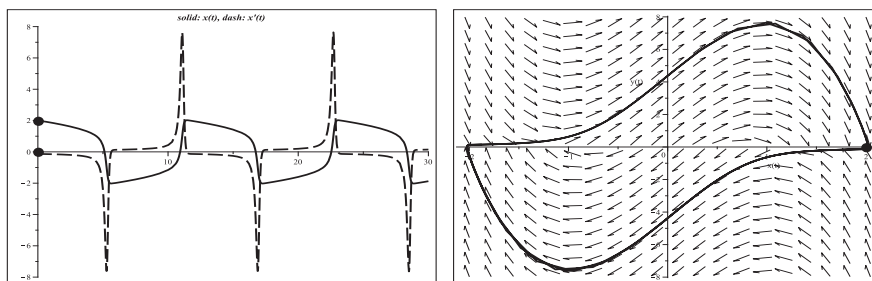


Abb. 2.295 Dateien *sys_Loes_09b31, 2.ps*, SysgDGI (9), $x(0)=2, y(0)=0, \varepsilon=5$, Näherung der Koordinatenfunktionen $x(t), y(t)$ in $[0, 30] \times [-8, 8]$, Kommandos `dsolve/numeric, plot, pointplot`, Näherung der Phasenkurve $(x(t), y(t))$ in $[-2, 2] \times [-8, 8]$, Kommandos `phaseportrait, pointplot`

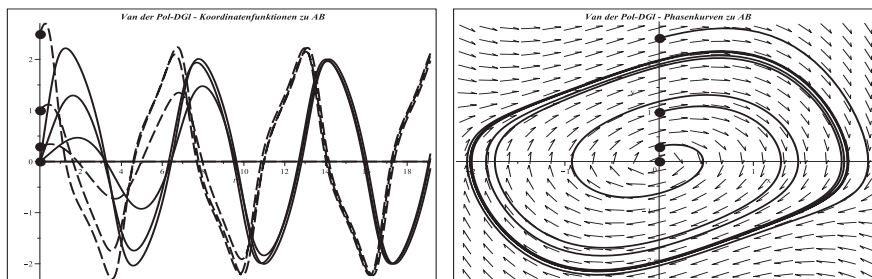


Abb. 2.296 Dateien *sys_Loes_09c11, 2.ps*, SysgDGI (9), $x(0)=0, y(0)=0, 0.3, 1, 2.5, \varepsilon=0.5$, Näherung der Koordinatenfunktionen $x(t), y(t), t \in [0, 3T_1]$, Kommandos `dsolve/numeric, plot, pointplot`, Näherung der Phasenkurven $(x(t), y(t))$ in $[-2, 2.2] \times [-2.5, 3]$, Kommandos `phaseportrait, pointplot`

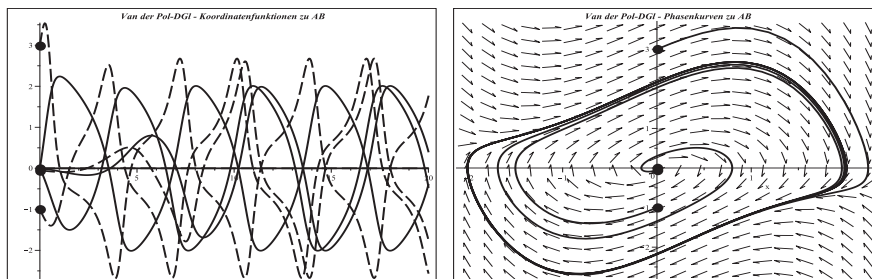


Abb. 2.297 Dateien *sys_Loes_09c21, 2.ps*, SysgDGI (9), $x(0)=0, y(0)=0, -0.05, 3, -1, \varepsilon=1$, Näherung der Koordinatenfunktionen $x(t), y(t), t \in [0, 3T_2]$, Kommandos `dsolve/numeric, plot, pointplot`, Näherung der Phasenkurven $(x(t), y(t))$ in $[-2, 2.2] \times [-3, 4]$, Kommandos `phaseportrait, pointplot`

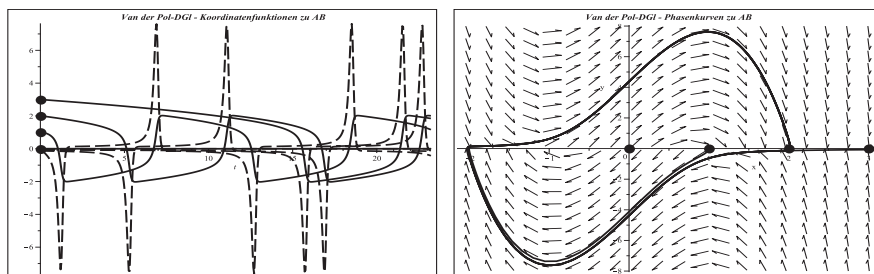


Abb. 2.298 Dateien *sys_Loes_09c31, 2.ps*, SysgDGI (9), $x(0)=0, 2, 1, 3$, $y(0)=0$, $\varepsilon = 5$, Näherung der Koordinatenfunktionen $x(t), y(t)$, $t \in [0, 2T_3]$, Kommandos `dsolve/numeric`, `plot`, `pointplot`, Näherung der Phasenkurven $(x(t), y(t))$ in $[-2, 3] \times [-8, 8]$, Kommandos `phaseportrait`, `pointplot`

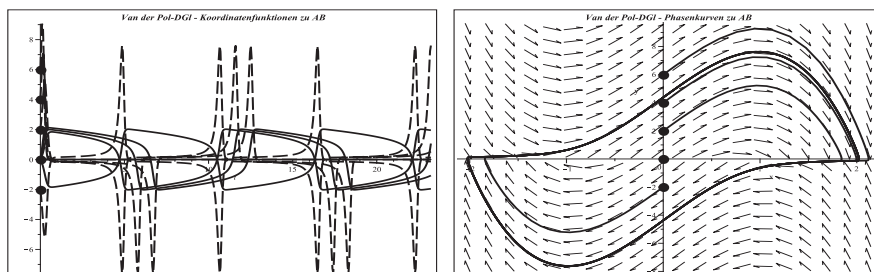


Abb. 2.299 Dateien *sys_Loes_09c41, 2.ps*, SysgDGI (9), $x(0)=0, y(0)=0, -2, 2, 4, 6$, $\varepsilon = 5$, Näherung der Koordinatenfunktionen $x(t), y(t)$, $t \in [0, 2T_3]$, Kommandos `dsolve/numeric`, `plot`, `pointplot`, Näherung der Phasenkurven $(x(t), y(t))$ in $[-2, 2.2] \times [-8, 10]$, Kommandos `phaseportrait`, `pointplot`

(10) Populationsmodell Gras – Hase – Fuchs – Jäger

Im Abschnitt 1.6 ist die Modellierung dieses Ökosystems bzw. als Räuber-Beute-Modell mittels gDGI vorgestellt und charakterisiert worden.

Das nichtlineare autonome SysgDGI 1. Ordnung für die Spezies Gras $g(t)$, Hase $h(t)$ und Fuchs $f(t)$ in Abhängigkeit von der Zeit t hat als AWP die Form

$$\begin{aligned} g'(t) &= 1 - h(t)g(t), & g(t_0) &= g_0, \\ h'(t) &= h(t)(g(t) - f(t)) - c_3 h(t), & h(t_0) &= h_0, \\ f'(t) &= f(t)h(t) - c_1 f(t) - c_2 \sqrt{f(t)}, & f(t_0) &= f_0, \\ & c_1 + c_2 = c = \text{const} > 0, & c_i &\geq 0. \end{aligned}$$

Dazu findet man dort auch die ersten Simulationsverläufe mit ausgewählten Parametern und AB sowie Betrachtungen zum Lösungsverhalten, Langzeitverhalten, stationären Zuständen und Gleichgewichtslagen.

Einfache Situationen und Lösungen sind wie folgt:

- (a) $h = 0$, $f = 0$, $g = t$ Gras wächst,
- (b) $h = 0$, f nimmt ab, g nimmt zu,
- (c) $g = 0$ (System ohne 1. gDGl), h nimmt ab, f nimmt zu und dann ab,
- (d) $f = 0$ (System ohne 3. gDGl), g , h nehmen zu bzw. ab.

Nun werden einige Modelle genauer untersucht sowie dazu die Maple-Anweisungen notiert und erläutert. Die Auswahl hier kann natürlich nicht alle möglichen Situationen berücksichtigen.

Modell 1

Parameter $c = c_1 + c_2 = \frac{4}{5}$, $c_1 = \frac{4}{5}$, $c_2 = 0$, $c_3 = 0$,

AB $g(0) = 1$, $h(0) = 1$, $f(0) = \frac{1}{10}$.

Das SysgDGl tendiert zu Schwingungen. Diese werden jedoch mit der Zeit herausgedämpft und es stellt sich ein Gleichgewicht ein.

Die Grenzwerte sind $g(\infty) = \frac{5}{4}$, $f(\infty) = \frac{5}{4}$, $h(\infty) = \frac{4}{5}$. Dies ist eine stabile Gleichgewichtslage (GGL) und zugleich eine stationäre Lösung (Fixpunkt).

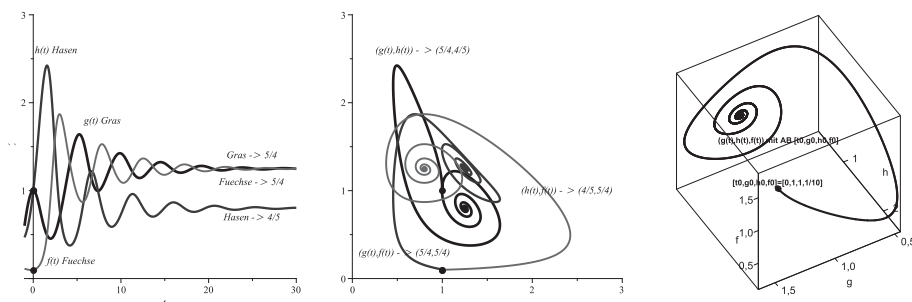


Abb. 2.300 Dateien *oeko15, 16, 17sw.eps*, SysgDGl (10),
Näherungen zu Lösungs- und Phasenkurven im 2D, 3D,
 $c_1 = \frac{4}{5}$, $c_2 = 0$, $c_3 = 0$, AB $g(0) = 1$, $h(0) = 1$, $f(0) = \frac{1}{10}$

Rechnungen in Maple (ausführliche Darstellungen)

```
> # Allgemein
# AB, Parameter, rechte Seiten, SysgDGl, NLGS, Jacobi-Matrix
t0:=0;
g0:=1; h0:=1; f0:=1/10;
c1:=2*c1': c2:=2*c2': c3:=2*c3':
AB:=[g0,h0,f0]; AB1:=[t0,g0,h0,f0];
> # rechte Seiten des SysgDGl bzw. NLGS
f1:=unapply(1-h*g,g,h,f,c1,c2,c3);
f2:=unapply(h*(g-f-c3),g,h,f,c1,c2,c3);
f3:=unapply(f*h-c1*f-c2*sqrt(f),g,h,f,c1,c2,c3);
```

```

> # SysgDGL
sys3:=diff(g(t),t)=f1(g(t),h(t),f(t),c1,c2,c3),
          diff(h(t),t)=f2(g(t),h(t),f(t),c1,c2,c3),
          diff(f(t),t)=f3(g(t),h(t),f(t),c1,c2,c3);
fcns:=g(t),h(t),f(t):
init:=g(t0)=g0,h(t0)=h0,f(t0)=f0;
> # NLGS
fp:=solve([f1(g,h,f,c1,c2,c3),f2(g,h,f,c1,c2,c3),f3(g,h,f,c1,c2,c3)],
          [g,h,f]);
> # Jacobi-Matrix der rechten Seite
f123:=[f1(g,h,f,c1,c2,c3), f2(g,h,f,c1,c2,c3), f3(g,h,f,c1,c2,c3)]:
Jf123:=matrix(3,3,[[diff(f123[1],g), diff(f123[1],h), diff(f123[1],f)],
                  [diff(f123[2],g), diff(f123[2],h), diff(f123[2],f)],
                  [diff(f123[3],g), diff(f123[3],h), diff(f123[3],f)]]);

```

Ergebnisse sind u.a.

$$\begin{aligned}
 fp &:= \left[\left[g = c_3, h = \frac{1}{c_3}, f = 0 \right], \right. \\
 &\quad \left[g = \text{RootOf}(_Z^3 c_1 + _Z^2 c_2 + (c_1 c_3 - 1) _Z + c_3 c_2)^2 + c_3, \right. \\
 &\quad \left. h = \frac{c_1 \text{RootOf}(_Z^3 c_1 + _Z^2 c_2 + (c_1 c_3 - 1) _Z + c_3 c_2) + c_2}{\text{RootOf}(_Z^3 c_1 + _Z^2 c_2 + (c_1 c_3 - 1) _Z + c_3 c_2)}, \right. \\
 &\quad \left. f = \text{RootOf}(_Z^3 c_1 + _Z^2 c_2 + (c_1 c_3 - 1) _Z + c_3 c_2)^2 \right] \\
 Jf123 &:= \begin{bmatrix} -h & -g & 0 \\ h & g - f - c_3 & -h \\ 0 & f & h - c_1 - \frac{c_2}{2\sqrt{f}} \end{bmatrix}
 \end{aligned}$$

Man bemerkt, dass im Fall $c_3 = 0$ der erste Lösungsanteil in fp i. Allg. gar nicht erst auftaucht und $f = 0$ besonderer Betrachtung bedarf.

Die analytische Lösung allgemein bzw. mit AB ist zu kompliziert.

```

> loes:=dsolve([sys3],[fcns]);
loe3:=dsolve([sys3,init],[fcns]);

```

Gemäß der Theorie der gDGL wird die Untersuchung des linearisierten SysgDGL mittels der Jacobi-Matrizen und ihrer reellen oder komplexen Eigenwerte (EW) λ an den Fixpunkten des nichtlinearen Gleichungssystems durchgeführt.

Somit haben wir folgende Fälle für die Bewertung der GGL im Fixpunkt (x_0, y_0, z_0) :

$\lambda_1 < \lambda_2 < \lambda_3 < 0$,	asymptotisch stabil, anziehender Punkt,
$0 < \lambda_1 < \lambda_2 < \lambda_3$,	instabil, abstoßender Punkt,
$\lambda_1 < 0 < \lambda_2, \lambda_3$,	stabil und instabil, Sattelpunkt,
$\Re(\lambda_1), \Re(\lambda_2), \Re(\lambda_3) < 0$,	asymptotisch stabil, Strudel, anziehender Punkt,
$\Re(\lambda_1), \Re(\lambda_2), \Re(\lambda_3) > 0$,	instabil, Strudel, abstoßender Punkt,
$\Re(\lambda_1) > 0, \Re(\lambda_2) < 0$,	stabil, instabil, Sattelpunkt,
$\Re(\lambda_1) = \Re(\lambda_2) = \Re(\lambda_3) = 0$,	stabiler Zyklus, Ellipse.

Kehren wir zum Modell 1 mit den Parametern $c_1 = \frac{4}{5}$, $c_2 = 0$, $c_3 = 0$ und seiner GGL $(g_\infty, h_\infty, f_\infty) = (\frac{5}{4}, \frac{4}{5}, \frac{5}{4})$ zurück.

```

> c1:=4/5: c2:=0: c3:=0:
  f1(g,h,f,c1,c2,c3): f2(g,h,f,c1,c2,c3): f3(g,h,f,c1,c2,c3):
  fp:=solve([f1(g,h,f,c1,c2,c3),f2(g,h,f,c1,c2,c3),f3(g,h,f,c1,c2,c3)],
            [g,h,f]);
> fpu1:=vector(3, []):
  hh:=op(fp):
  fpu1:=[rhs(hh[1]),rhs(hh[2]),rhs(hh[3])];
> Jf123:=matrix(3,3,[[diff(f123[1],g), diff(f123[1],h), diff(f123[1],f)],
                    [diff(f123[2],g), diff(f123[2],h), diff(f123[2],f)],
                    [diff(f123[3],g), diff(f123[3],h), diff(f123[3],f)]]);
> A:=matrix(3,3, []): A1:=A:
  A1:=eval(Jf123,{g=fpu1[1],h=fpu1[2],f=fpu1[3]});

> # Loesung mittels EWP zur Koeffizientenmatrix A1
  lambda:='lambda':
  charpoly(A1,lambda);
  lambda:=eigenvalues(A1); lambda:=evalf(lambda);
  v:=evalf([eigenvectors(A1)]);
  nv:=nops(v);
> # EWP
  l1a:=v[1][1]; l2a:=v[2][1]; l3a:=v[3][1]; # 1.-3. EW
  print('\n');
  print('Loesung des EWP');
  for i from 1 by 1 to nv do
    printf("          %2d. EW mit AVF, EV \n",i):
    v[i][1];          # i-ter EW
    v[i][2];          # seine algebraische Vielfachheit (AVF)
    op(v[i][3]);      # zugehoeriger EV
  end do;
> # Matrix der EV, spaltenweise, Spalten als Vektoren linear unabhaengig
  V:=matrix(nv,0, []);
  for i from 1 by 1 to nv do
    V:=concat(V,op(v[i][3])); # EV als Spalten einer Matrix
  end do:
  V:=evalm(V);

```

Ausgewählte Ergebnisse

$$\begin{aligned}
 fp &:= \left[\left[g = \frac{5}{4}, h = \frac{4}{5}, f = \frac{5}{4} \right] \right] \\
 fpu1 &:= \left[\left[\frac{5}{4}, \frac{4}{5}, \frac{5}{4} \right] \right] \\
 Jf123 &:= \begin{bmatrix} -h & -g & 0 \\ h & g-f & -h \\ 0 & f & h-\frac{4}{5} \end{bmatrix}
 \end{aligned}$$

$$A1 := \begin{bmatrix} -\frac{4}{5} & -\frac{5}{4} & 0 \\ \frac{4}{5} & 0 & -\frac{4}{5} \\ 0 & \frac{5}{4} & 0 \end{bmatrix}$$

$$\lambda := -0.4345015158, -0.1827492420 - 1.344541955 I, -0.1827492420 + 1.344541955 I$$

$$V := \begin{bmatrix} 1.188791568 & -0.774395786 + 0.4914280466 I & -0.774395786 - 0.4914280466 I \\ -0.3476012129 & -0.1461993934 - 1.075633565 I & -0.1461993934 + 1.075633565 I \\ 1. & 1. & 1. \end{bmatrix}$$

Die drei EW (reell und komplex) haben negativen Realteil, sodass der Fixpunkt (stationäre Lösung) $(\frac{5}{4}, \frac{4}{5}, \frac{5}{4})$ asymptotisch stabil und ein anziehender Strudel ist.

Man kontrolliert, dass die Lösung des AWP mit dem Fixpunkt als AB im Fixpunkt bleibt.

```
> g01:=fpu1[1]; h01:=fpu1[2]; f01:=fpu1[3];
  init1:=g(t0)=g01,h(t0)=h01,f(t0)=f01;
  loe31:=dsolve([sys3,init1],[fcns]);

> p11:=phaseportrait([sys3],[fcns],t=-3..3,[[init1]],
  linecolor=green,scene=[t,g(t)]):
  p12:=textplot([[1,1.35,'g(t) Gras']]):
  p13:=pointplot([t0,g01],symbol=solidcircle,symbolsize=20,color=green):
  p21:=phaseportrait([sys3],[fcns],t=-3..3,[[init1]],
  linecolor=yellow,scene=[t,h(t)]):
  p22:=textplot([[1,0.7,'h(t) Hasen']]):
  p23:=pointplot([t0,h01],symbol=solidcircle,symbolsize=20,color=yellow):
  p31:=phaseportrait([sys3],[fcns],t=-3..3,[[init1]],
  linecolor=red,scene=[t,f(t)]):
  p32:=textplot([[1,1.15,'f(t) Fuechse']]):
  p33:=pointplot([t0,f01],symbol=solidcircle,symbolsize=20,color=red):
  display(p11,p12,p13,p21,p22,p23,p31,p32,p33,
  view=[-3..3,0..2],labels=['t',' ']);
```

Es folgt die näherungsweise Lösung des AWP mittels `phaseportrait`, `method=rkf45` mit Szenemenü, wo sowohl die Koordinatenfunktionen als auch Phasenkurven gewählt werden können.

Mit `DEplot3d` und Szenemenü erfolgt die 3D-Darstellung. Bei der Dimension des Systems ≥ 3 werden keine Richtungsfelder erzeugt. Die Grafiken werden aus dem Maple-Arbeitsblatt als eps-Dateien exportiert und die Farben dann in Grautöne umgewandelt.

```
> # Loesungskurven
  p11:=phaseportrait([sys3],[fcns],t=-1..30,[[init]],stepsize=0.01,
  linecolor=green,scene=[t,g(t)]):
  p12:=textplot([[8,1.8,'g(t) Gras'],[20,1.4,'Gras -> 5/4']]):
```

```

p13:=pointplot([t0,g0],symbol=solidcircle,symbolsize=25,color=green):
p21:=phaseportrait([sys3],[fcns],t=-1..30,[[init]],stepsize=0.01,
    linecolor=yellow,scene=[t,h(t)]):
p22:=textplot([[2.8,2.6,'h(t) Hasen'],[25,0.7,'Hasen -> 4/5']]):
p23:=pointplot([t0,h0],symbol=solidcircle,symbolsize=20,color=yellow):
p31:=phaseportrait([sys3],[fcns],t=-1..30,[[init]],stepsize=0.01,
    linecolor=red,scene=[t,f(t)]):
p32:=textplot([[4.5,0.2,'f(t) Fuechse'],[25,1.1,'Fuechse -> 5/4']]):
p33:=pointplot([t0,f0],symbol=solidcircle,symbolsize=20,color=red):
display(p11,p12,p13,p21,p22,p23,p31,p32,p33,
    view=[-1..30,0..3],labels=['t',' ']);

> # Phasenkurven
p11:=phaseportrait([sys3],[fcns],t=0..30,[[init]],stepsize=0.01,
    linecolor=green,scene=[g(t),h(t)]):
p12:=textplot([[0.8,2.6,'(g(t),h(t)) -> (5/4,4/5)']]):
p13:=pointplot([g0,h0],symbol=solidcircle,symbolsize=20,color=green):
p21:=phaseportrait([sys3],[fcns],t=0..30,[[init]],stepsize=0.01,
    linecolor=yellow,scene=[g(t),f(t)]):
p22:=textplot([[0.6,0.3,'(g(t),f(t)) -> (5/4,5/4)']]):
p23:=pointplot([g0,f0],symbol=solidcircle,symbolsize=20,color=yellow):
p31:=phaseportrait([sys3],[fcns],t=0..30,[[init]],stepsize=0.01,
    linecolor=red,scene=[h(t),f(t)]):
p32:=textplot([[2.4,1,'(h(t),f(t)) -> (4/5,5/4)']]):
p33:=pointplot([h0,f0],symbol=solidcircle,symbolsize=20,color=red):
display(p11,p12,p13,p21,p22,p23,p31,p32,p33,
    view=[0..3,0..3],labels=['',' ']);

> # 3D-Phasenkurve, scene=[g(t),h(t),f(t)]
# Ausdehnung des Gebiets richtet sich nach "Dimension" der Phasenkurve
p3:=DEplot3d([sys3],[fcns],t=0..30,[[init]],linecolor=blue,thickness=2,
    stepsize=0.01,labels=['g','h','f'],orientation=[60,45]):
p4:=pointplot3d([g0,h0,f0],symbol=solidcircle,symbolsize=20,color=blue):
p5:=textplot3d([[1,1,0.2,' [t0,g0,h0,f0]=[0,1,1,1/10]'],
    [1,1,1,' (g(t),h(t),f(t)) mit AB [t0,g0,h0,f0]'],
    font=[HELVETICA,BOLD,8],color=blue]):
p6:=pointplot3d(fp1,symbol=solidcircle,symbolsize=20,color=black):
display(p3,p4,p5,p6);

```

Modell 2

Parameter $c = c_1 + c_2 = \frac{4}{5}$, $c_1 = \frac{3}{10}$, $c_2 = \frac{5}{10}$, $c_3 = 0$,

AB $g(0) = 1$, $h(0) = 1$, $f(0) = \frac{1}{10}$.

Das SysgDGL tendiert zu Schwingungen. Diese werden jedoch mit der Zeit herausgedämpft und es stellt sich ein Gleichgewicht ein. Die Grenzwerte (Fixpunkte) sind $g(\infty) = \frac{1}{18}(85 - 5\sqrt{145}) = 1.377334838$, $h(\infty) = \frac{1}{40}(17 + 5\sqrt{145}) = 0.7260398645$, $f(\infty) = g(\infty)$. Dies ist eine stabile GGL und zugleich eine stationäre Lösung.

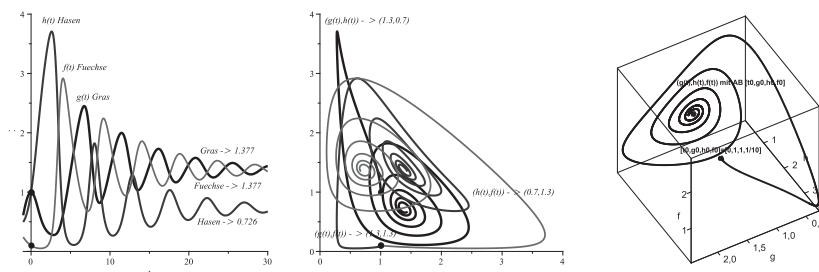


Abb. 2.301 Dateien *oeko25, 26, 27sw.eps*, SysgDGI (10),

Näherungen zu Lösungs- und Phasenkurven im 2D, 3D,

$$c_1 = \frac{3}{10}, c_2 = \frac{5}{10}, c_3 = 0, \text{ AB } g(0) = 1, h(0) = 1, f(0) = \frac{1}{10}$$

Wegen der Ungleichung $1 > \sqrt{f(t)} > f(t)$ haben wir am Anfang eine höhere Abschussquote der Füchse als im ersten Modell. Die Zahl der Füchse wird so stark dezimiert, sodass es zu einer richtigen Hasenplage kommt. Die Folge ist wieder ein starker Rückgang der Grasmenge. Die Vorgehensweise ist analog zum Modell 1. Es werden ausgewählte wichtige Ergebnisse von Rechnungen in Maple gezeigt.

$$fp := \left[\left[g = \frac{1}{18}(85 - 5\sqrt{145}), h = \frac{1}{40}(17 + 5\sqrt{145}), f = \frac{1}{18}(85 - 5\sqrt{145}) \right] \right]$$

$$Jf123 := \begin{bmatrix} -h & -g & 0 \\ h & g - f & -h \\ 0 & f & h - \frac{3}{10} - \frac{1}{4\sqrt{f}} \end{bmatrix}$$

$$A2 := \begin{bmatrix} -\frac{1}{40}(17 + \sqrt{145}) & -\frac{5}{18}(17 - \sqrt{145}) & 0 \\ \frac{1}{40}(17 + \sqrt{145}) & 0 & -\frac{1}{40}(17 + \sqrt{145}) \\ 0 & \frac{5}{18}(17 - \sqrt{145}) & \frac{1}{40}(5 + \sqrt{145}) - \frac{1}{4\sqrt{(85 - 5\sqrt{145})/18}} \end{bmatrix}$$

$$\begin{bmatrix} -0.7260398644 & -1.377334839 & 0. \\ 0.7260398644 & 0. & -0.7260398644 \\ 0. & 1.377334839 & 0.2130199322 \end{bmatrix}$$

$$\lambda := -0.2881258357, -0.1124470482 - 1.329622840 I, -0.1124470482 + 1.329622840 I$$

$$V := \begin{bmatrix} 1.144393043 & -0.7312990978 + 0.5822604949 I & -0.7312990978 - 0.5822604949 I \\ -0.3638518054 & -0.2363020023 - 0.9653591872 I & -0.2363020023 - 0.9653591872 I \\ 1. & 1. & 1. \end{bmatrix}$$

Die drei EW (reell und komplex) haben negativen Realteil, sodass die stationäre Lösung (Fixpunkt) $(\frac{1}{18}(85 - 5\sqrt{145}), \frac{1}{40}(17 + 5\sqrt{145}), \frac{1}{18}(85 - 5\sqrt{145}))$ asymptotisch stabil und ein anziehender Strudel ist.

Einige weitere Modelle beinhalten u.a. spezielle Parametersituationen und AB, Besonderheiten bei den Lösungen, Hinweise zum Anfangsverhalten der Lösungskurven und den Aspekt der Auswahl des numerischen Verfahrens im Maple-Kommando.

Modell 3

Parameter $c = c_1 + c_2 = \frac{3}{10}$, $c_1 = \frac{1}{10}$, $c_2 = \frac{1}{5}$, $c_3 = 0$,

AB $g(0) = 1$, $h(0) = 1$, $f(0) = \frac{1}{10}$.

Das SysgDGL liefert gedämpfte Schwingungen.

Die EW der Jacobi-Matrix sind -0.07205752160 , $-0.03555436314 \pm 1.409102087i$.

Die GGL ist $g_\infty = 12 - 2\sqrt{11} = 5.366750420$, $h_\infty = \frac{1}{50}(6 + \sqrt{11}) = 0.1863324958$, $f_\infty = g_\infty$.

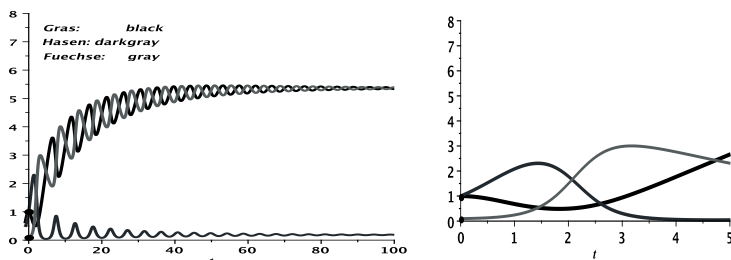


Abb. 2.302 Dateien *oeko31sw*, *ksw.eps*, SysgDGL (10),

Näherungen zu Lösungskurven in $[-1, 100] \times [0, 8]$ bzw. $[0, 5] \times [0, 8]$,

$c_1 = \frac{1}{10}$, $c_2 = \frac{1}{5}$, $c_3 = 0$, AB $g(0) = 1$, $h(0) = 1$, $f(0) = \frac{1}{10}$

Modell 4

Parameter $c = c_1 + c_2 = \frac{1}{4}$, $c_1 = \frac{1}{20}$, $c_2 = \frac{1}{5}$, $c_3 = \frac{1}{100}$,

AB $g(0) = g_0$, $h(0) = h_0$, $f(0) = f_0$.

Das SysgDGL führt zunächst auf die Lösung des nichtlinearen Gleichungssystems

$f_k(g, h, f, c_1, c_2, c_3) = 0$, $k = 1, 2, 3$, und die drei GGL

(1) $g_\infty = \frac{1}{100}$, $h_\infty = 100$, $f_\infty = 0$,

(2) $g_\infty = 8.3999951351197697$, $h_\infty = 0.1190476879943743$, $f_\infty = 8.3899951351197697$,

(3) $g_\infty = 0.0100040072142311$, $h_\infty = 99.9599439090228147$, $f_\infty = 0.0000040072142311$.

Zur Deutung der Stabilität berechnet man die EW der entsprechenden Jacobi-Matrix

$$Jf123 = \begin{bmatrix} -h & -g & 0 \\ h & g - f - \frac{1}{100} & -h \\ 0 & f & h - \frac{1}{20} - \frac{1}{10\sqrt{f}} \end{bmatrix}.$$

Im ersten Fall taucht in $Jf123$ wegen $f = 0$ die Division durch Null auf.

$f = 0$ bedeutet aber auch das Verschwinden der 3. Gleichung im SysgDGL. Damit nimmt man aus der Matrix den linken oberen (2×2) -Block und berechnet dazu die EW.

Somit erhält man für die drei Fälle von GGL folgende EW:

- (1) $\lambda_1 = -99.989998999 < 0$, $\lambda_2 = -0.0100010002 < 0$,
- (2) $\lambda_1 = -0.0423410858 < 0$, $\lambda_{2,3} = -0.0210913790 \pm 1.411548309I$, $\Re(\lambda_{2,3}) < 0$,
- (3) $\lambda_1 = 49.954937564 > 0$, $\lambda_2 = -99.949938900 < 0$, $\lambda_3 = -0.0099969909 < 0$.

Die zweite GGL ist stabil, die dritte instabil. Bei der ersten Situation muss man genauer hinsehen. Wenn die AB $f_0 = 0$ ist, wird die Teillösung $f = 0$ werden. Die numerische Lösung des AWP mit den AB $g(0) > 0$, $h(0) > 0$, $f(0) = 0$ wird deshalb immer gegen die stationäre Lösung $g_\infty = \frac{1}{100}$, $h_\infty = 100$, $f_\infty = 0$ tendieren. Man hat eine "teilweise" Stabilität für das auf zwei Gleichungen reduzierte SysgDGL.

Sobald in der AB jedoch $g(0) > 0$, $h(0) > 0$, $f(0) > 0$ ist, wird im weiteren Verlauf i. Allg. die GGL (2) angestrebt, manchmal brechen jedoch die Verfahren ab.

Wir betrachten einige einfache Situationen von AB, wobei $f(0) > 0$ ist.

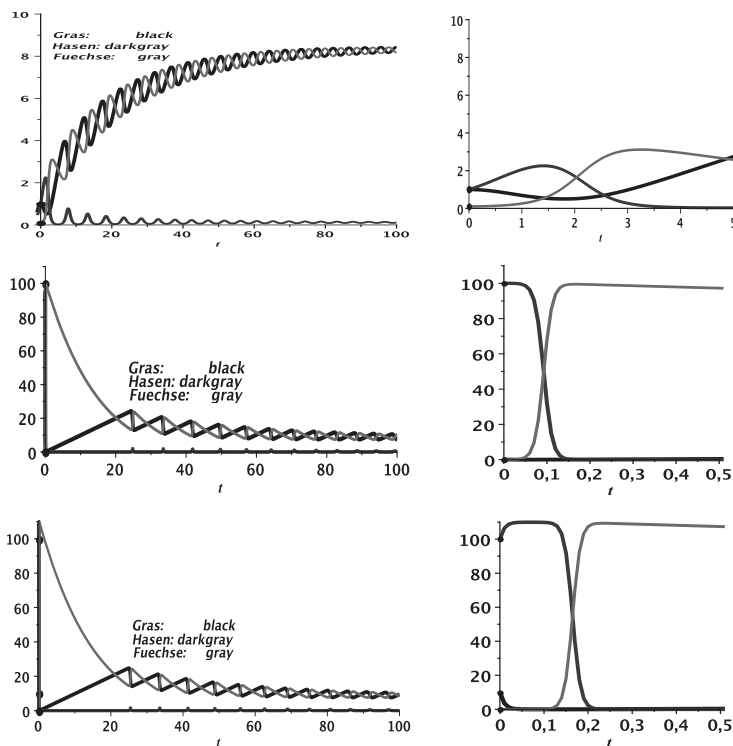


Abb. 2.303 Dateien *oeko41, 2, 3sw, ksw.eps*, SysgDGL (10), $c_1 = \frac{1}{20}$, $c_2 = \frac{1}{5}$, $c_3 = \frac{1}{100}$,

Näherungen zu Lösungskurven mit der GGL (2)

in $[-1, 100] \times [0, 10 \setminus 100 \setminus 110]$ bzw. $[0, 5 \setminus 0.5 \setminus 0.5] \times [0, 10 \setminus 100 \setminus 110]$,

AB v.o.n.u. $g(0) = 1$, $h(0) = 1$, $f(0) = \frac{1}{10}$,

$g(0) = \frac{1}{10}$, $h(0) = 100$, $f(0) = \frac{1}{100}$,

$g(0) = 10$, $h(0) = 100$, $f(0) = \frac{1}{100000}$

Zu den drei GGL werden noch einige Tests durchgeführt.

(a) Die numerische Lösung von AWP mit der AB = GGL (1) bzw. GGL (2) liefert konstante Lösungen, die jeweils mit diesen GGL übereinstimmen.

(b) Die numerische Lösung von AWP mit der AB gleich der instabilen GGL (3) liefert in Abhängigkeit vom Verfahren Näherungslösungen mit ganz unterschiedlichen Ergebnissen. Es wurden sieben in Maple implementierte Verfahren getestet.

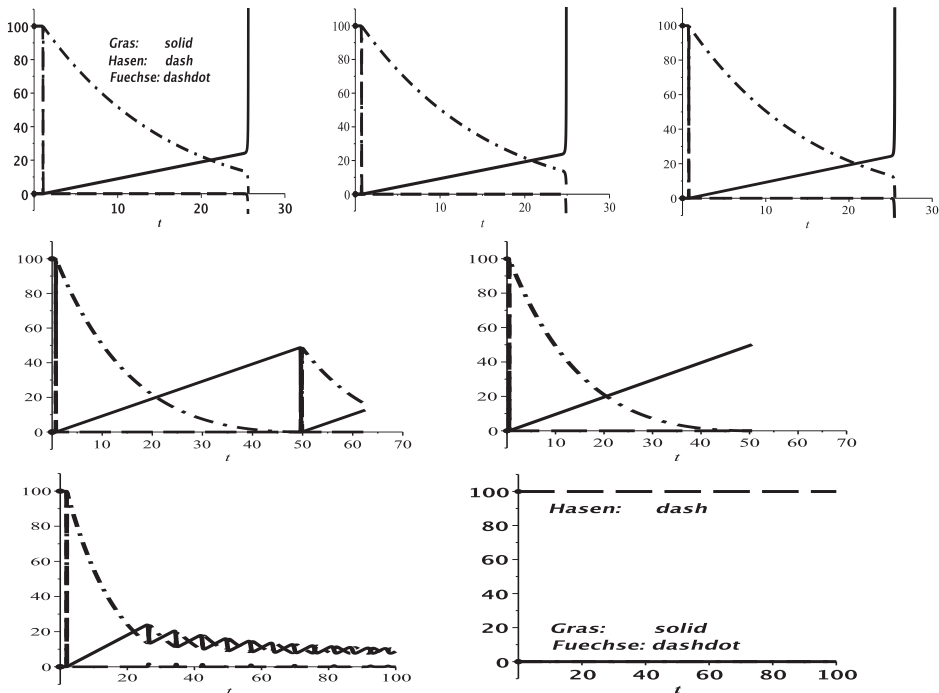


Abb. 2.304 Dateien *oeko_rkf45.eps*, *gear*, *dverk78*, *rk4*, *lsode*, *ck45*, *rosenbrock*, SysgDGl (10), $c_1 = \frac{1}{20}$, $c_2 = \frac{1}{5}$, $c_3 = \frac{1}{100}$, Näherungen mittels Verfahren v.l.o.n.r.u. *rkf45*, *gear*, *dverk78*, *rk4*, *lsode*, *ck45*, *rosenbrock* zur stationären Lösung mit der AB = instabile GGL (3)
 $g_0 = g_\infty = 0.0100040072142311$, $h_0 = h_\infty = 99.9599439090228147$,
 $f_0 = f_\infty = 0.0000040072142311$, in $[0, 30 \setminus 70 \setminus 100] \times [-10, 110]$

Das Verhalten der Verfahren ist aus den Warnungen erkennbar. Alle Methoden bis auf *ck45*, das zur stabilen GGL (2) tendiert, und *rosenbrock*, das an der AB "stehen bleibt", brechen die Rechnung ab.

```
method=rkf45,stepsize=0.01, 0.001, ...
```

```
Warning, plot may be incomplete, the following error(s) were issued:
cannot evaluate the solution further right of 25.631396,
probably a singularity
```

```

methode=gear
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution past 24.915964,
step taken with h = hmin but the requested error not achieved
method=dverk78
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution past 25.510077, step size < hmin,
problem may be singular or error tolerance may be too small
method=classical[rk4]
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution past 62.51,
maxfun limit exceeded (see ?dsolve,maxfun for details)
method=lsode
Warning, plot may be incomplete, the following errors(s) were issued:
complex argument to max/min
method=ck45
-
method=rosenbrock
-

```

(c) Zahlreiche AB erweisen sich als ungeeignet, wie in der letzten Grafik der nächsten Abbildung zu sehen ist. Es kann dann passieren, dass die Rechnung vorzeitig abbricht. Man erhält die Fehlermeldung

```

Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution further right of ..., probably a singularity

```

Näherungskurven zu AB nahe der instabilen GGL (3), erzeugt mittels `phaseportrait`, `method=rkf45`, `stepsize=0.01`, führen entweder zum Abbruch oder tendieren gegen GGL (2). Man teste hier auch andere Methoden.

(g0,	h0,	f0)	Rechnung bis bzw. Trend zu
(0.0100040072142311,	99.9599439090228147,	0.0000040072142311)	25.6
(0.010004007214231,	99.959943909022814,	0.000004007214231)	0.67
(0.01000400721423,	99.959943909022,	0.00000400721423)	0.56
(0.010004007214,	99.9599439090,	0.000004007214)	0.36
(0.01000400721,	99.959943909,	0.00000400721)	0.29
(0.010004,	99.9599,	0.000004)	0.14
(0.010004,	99.9599,	0.000001)	0.014
(0.010004,	99.9599,	0.00001 ... 0.001, 0.1)	GGL (2)
(0.010004,	99.9599,	0.005 ... 0.008)	GGL (2)
(0.010004,	99.9599,	0.009 ... 0.01)	24.4
(0.010004,	99.9599,	0.011 ... 0.015)	GGL (2)
(0.01,	99.9,	0.0001, 0.001, 0.01, 0.014)	24.5
(0.01,	99.9,	0.015, 0.016, 0.1, 0.2)	GGL (2)
(0.01,	99.99,	0.01)	GGL (2)

(d) Die Besonderheit bei der GGL (1) $(0.01, 1000, 0)$, wenn in der AB mit $f_0 = 0$ bzw. $f_0 > 0$ gerechnet wird, haben wir schon erwähnt. Mit dem Kommando `dsolve(sys3,initw,fcns,numeric,output=listprocedure,maxfun=200000)` erhält man beispielsweise

```

Annaeherung an die GGL (1)
g0:=0.01: h0:=90.0: f0:=0:
g(700)=0.01000090217399760, h(700)=99.99088675625142, f(700)=0

```

```

keine Annaeherung an die GGL (1), weil daneben die instabile GGL (3) liegt
g0:=0.01: h0:=100: f0:=0.00001:
g(700)=8.399998926181106, h(700)=0.1190492073230743, f(700)=8.389990355303944

```

Sobald Füchse da sind, vermehren diese sich und die Hasenpopulation geht zurück. Meistens ist über einen sehr langen Zeitraum zu rechnen, um die Asymptotik mit $h(t) \approx 0.1$ und $g(t) \approx f(t) \gg h(t)$ zur stabilen GGL (2) zu erkennen.

(e) Es werden noch Rechnungen mittels `phaseportrait,method=rkf45,stepsize=0.01` zu weiteren AB gemacht. Dabei werden die Anfangsverläufe "gezoomt". Man bemerkt, dass die numerischen Verfahren durchaus empfindlich auf kleine Veränderungen in den AB reagieren.

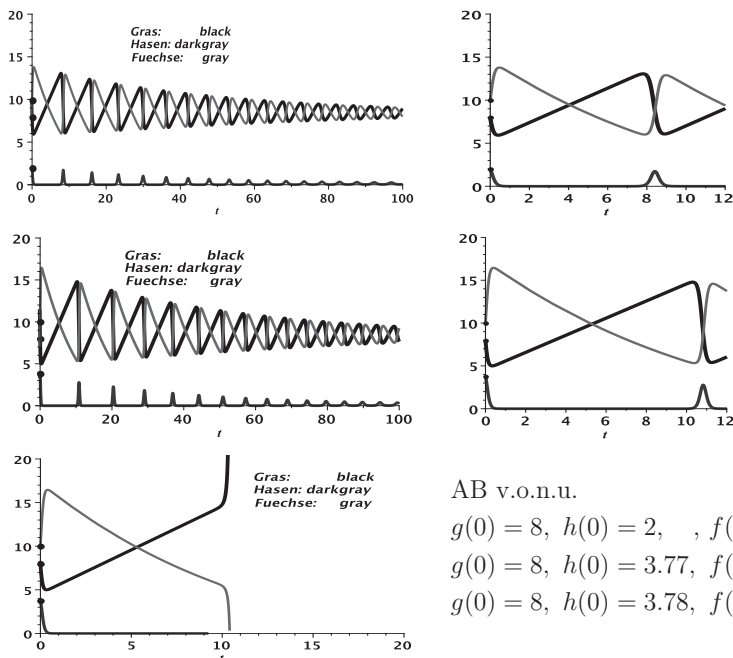


Abb. 2.305 Dateien *oeko4, 5, 6sw, ksw.eps*, SysgDGl (10), $c_1 = \frac{1}{20}$, $c_2 = \frac{1}{5}$, $c_3 = \frac{1}{100}$,
 Näherungen zu Lösungskurven
 in $[-1, 100 \setminus 20] \times [0, 20]$ bzw. $[0, 12] \times [0, 20]$,
 Kommando `phaseportrait,method=rkf45,stepsize=0.01`

Berechnet man nun die letzte Näherung zur AB $g(0) = 8, h(0) = 3.78, f(0) = 10$, die bei $t \approx 10.414364$ wegen wahrscheinlicher Singularität endete, mit der Methode `phaseportrait,method=gear`, so erhält man den gewünschten Trend zur GGL (2).

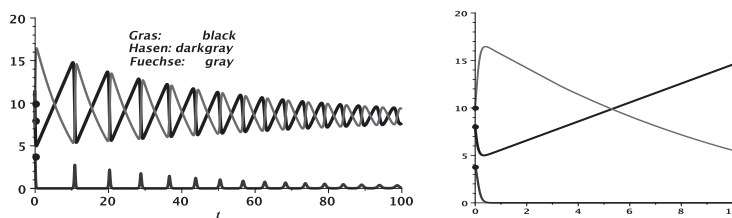


Abb. 2.306 Dateien `oeko46sw_gear, ksw_gear.eps`,
 SysgDGL (10), $c_1 = \frac{1}{20}, c_2 = \frac{1}{5}, c_3 = \frac{1}{100}$,
 Näherungen zu Lösungskurven in $[-1, 100] \times [0, 20]$ bzw. $[0, 10] \times [0, 20]$,
 AB $g(0) = 8, h(0) = 3.78, f(0) = 10$,
 Kommando `phaseportrait,method=gear,stepsize=0.01`

2.7 Spezielle gewöhnliche Differentialgleichungen

Spezielle gDGL bzw. Systeme treten u.a. im Zusammenhang mit Transformationen sowie speziellen Funktionen wie orthogonalen Polynomen, Schwingungs-, Zylinder- oder Kugelfunktionen auf. Letztere bzw. ihre Linearkombinationen sind Lösungen dieser gDGL. Oft genügen diese Funktionen auch noch rekursiven Beziehungen, z.B. einer 3-Term-Rekursion, haben eine Reihenentwicklung, die sich mittels Potenzreihenansatz finden lässt, und können durch hypergeometrische Funktionen ausgedrückt werden. Aus der Potenzreihe erkennt man Definitionsbereich, Konvergenz, asymptotisches Verhalten sowie Grenzwerte der Funktion und ihrer Ableitungen. Außerdem wird zur Unterscheidung und "Katalogisierung" der Funktionen oft die Art/Gattung und Ordnung (nichtnegative ganzzahlige, ganzzahlige, reelle) verwendet.

Bei der Betrachtung spezieller gDGL werden wir uns auf wenige ausgewählte Aspekte beschränken.

Beispiel 2.10

(1) Besselsche DGL

$$x^2 y'' + x y' + (x^2 - n^2) y = 0, \quad n \geq 0, \text{ ganzzahlig,}$$

$$y(x) = c_1 J_n(x) + c_2 Y_n(x)$$

mit den Bessel-Funktionen 1. bzw. 2. Art

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k (x/2)^{2k+n}}{\Gamma(n+k+1) k!} \quad \text{mit der Gammafunktion } \Gamma(x),$$

$$Y_n(x) = \frac{2}{\pi} \left(\gamma + \ln \left(\frac{x}{2} \right) \right) J_n(x) - \frac{1}{\pi} \sum_{k=0}^{n-1} \frac{(n-k-1)!}{k!} \left(\frac{x}{2} \right)^{2k-n} - \frac{1}{\pi} \sum_{k=0}^{\infty} (-1)^k \frac{H_k + H_{k+n}}{k! (n+k)!} \left(\frac{x}{2} \right)^{2k+n},$$

wobei γ die Eulersche Konstante und H_n die harmonische Reihe (n -te Partialsumme) sind. Im Ursprung $x = 0$ sind die Funktionen $J_n(x)$ für ganzzahlige n endlich, während die Funktionen $Y_n(x)$ eine logarithmische Singularität besitzen.

Als SysgDGI notiert man

$$\begin{aligned} y' - z &= 0, \\ x^2 z' + xz + (x^2 - n^2)y &= 0, \end{aligned}$$

$$\begin{aligned} y(x) &= \frac{1}{(n+x)(n-x)n^2} \cdot \\ & \left[c_1 x^n n^3 H\left(0, n, -2, \frac{n^2}{4}, 1 - \frac{n^2}{4}, \frac{x^2}{n^2}\right) + c_1 2x^{2+n} H'\left(0, n, -2, \frac{n^2}{4}, 1 - \frac{n^2}{4}, \frac{x^2}{n^2}\right) \right. \\ & \left. - c_2 x^{-n} n^3 H\left(0, -n, -2, \frac{n^2}{4}, 1 - \frac{n^2}{4}, \frac{x^2}{n^2}\right) + c_2 2x^{2-n} H'\left(0, -n, -2, \frac{n^2}{4}, 1 - \frac{n^2}{4}, \frac{x^2}{n^2}\right) \right], \\ z(x) &= c_1 x^{n-1} H\left(0, n, -2, \frac{n^2}{4}, 1 - \frac{n^2}{4}, \frac{x^2}{n^2}\right) + c_2 x^{-n-1} H\left(0, -n, -2, \frac{n^2}{4}, 1 - \frac{n^2}{4}, \frac{x^2}{n^2}\right) \end{aligned}$$

mit der konfluenten Heunschen Funktion $H(\alpha, \beta, \gamma, \delta, \eta, z)$ und ihrer Ableitung (Schreibweise aus Maple).

Für die exakte und numerische Lösung von AWP zur gDGI (1) sollen zunächst die Bessel-Funktionen dargestellt werden.

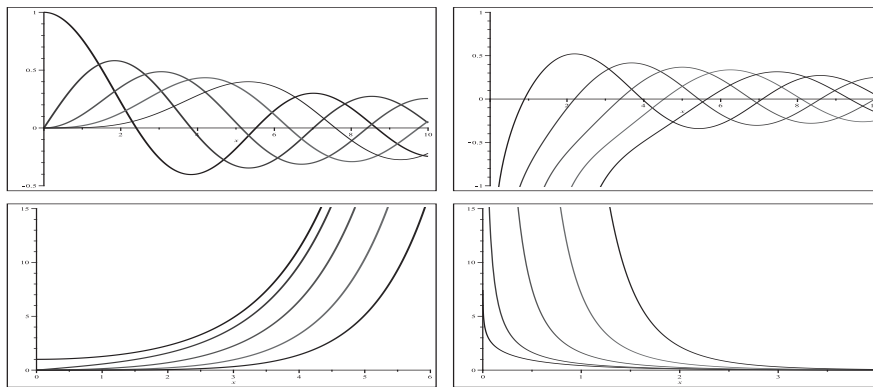


Abb. 2.307 Dateien *spez_dgl_011..4.ps*,
Bessel-Funktionen $J_k(x)$, $Y_k(x)$, $I_k(x)$, $K_k(x)$, $k = 0(1)4$ (v.l.o.n.r.u)

Weil die gDGI von 2. Ordnung und nicht autonom ist, kann kein Richtungsfeld erzeugt werden.

In der allgemeinen parameterabhängigen Lösung $y(x) = c_1 J_n(x) + c_2 Y_n(x)$ zeigt sich, dass für AWP mit AB $y(x_0) = y_0$, $x_0 = 0$, die Singularität von $Y_n(x)$ stört.

Nur AB $y(0) = y_0$, $y'(0) = y'_0$, die sich mit $J_n(x)$ "vertragen" und gleichzeitig $c_2 = 0$ bewirken, können verwendet werden. Außerdem passen z.B. AB $y(0) = 0$, $y'(0) = 0$ sowohl zur Nulllösung als auch zu den Funktionen $J_n(x)$, $n = 2, 3, \dots$. Also stellt sich die Frage, ob dann mit einer weiteren AB die Eindeutigkeit der Lösung erreichbar ist. Welche Ergebnisse werden nun mit Maple berechnet? Einige Fälle sollen betrachtet und unterschieden werden.

(a) AWP mit AB bei $x_0 \neq 0$ liefern eine eindeutige Lösung.

Mit $n = 0$ und den AB $y(1) = J_0(1) + Y_0(1) = 0.853454650$, $y'(1) = J'_0(1) + Y'_0(1) = -J_1(1) - Y_1(1) = 0.341162235$ erhält man die Lösung

$$y(x) = J_0(x) + Y_0(x).$$

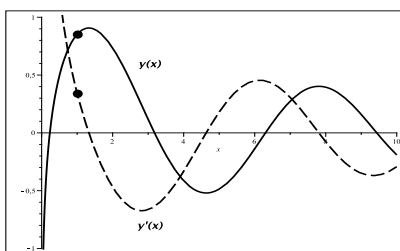


Abb. 2.308

Datei *spez_dgl_01a1.ps*, gDGl (1),
 AB $y(1) = J_0(1) + Y_0(1)$, $y'(1) = J'_0(1) + Y'_0(1)$,
 Lösung $y(x) = J_n(x) + Y_n(x)$
 mit Ableitung $y'(x)$, $x \in (0, 10]$,
 Kommandos `dsolve`, `plot`, `pointplot`

Mit $n = 1$ und den AB $y(1) = 1$, $y'(1) = 0$ erhält man die Lösung

$$\begin{aligned} y(x) &= \frac{-(Y_0(1) - Y_1(1))J_1(x) + (J_0(1) - J_1(1))Y_1(x)}{Y_1(1)J_0(1) - Y_0(1)J_1(1)} \\ &= 1.365759945 J_1(x) - 0.510739871 Y_1(x). \end{aligned}$$

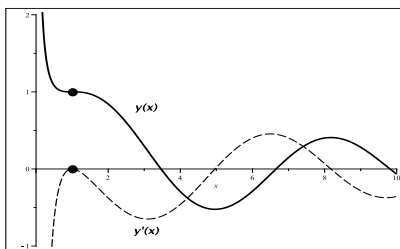


Abb. 2.309

Datei *spez_dgl_01a2.ps*, gDGl (1),
 AB $y(1) = 1$, $y'(1) = 0$,
 Lösung $y(x) = 1.365 J_1(x) - 0.510 Y_1(x)$
 mit Ableitung $y'(x)$, $x \in (0, 10]$,
 Kommandos `dsolve`, `plot`, `pointplot`

(b) AWP mit AB bei $x_0 = 0$ liefern eine eindeutige Lösung, wenn die AB zum "Ausblenden" des Lösungsanteils $Y_n(x)$ führt. Dabei kann die AB durchaus mehr Anteile haben.

Das sind z.B.

$$n = 0, \quad y(0) = 1, \quad y'(0) = 0 \quad \text{mit der Lösung} \quad y(x) = J_0(x) = 1 - \frac{x^2}{4} + \frac{x^4}{64} - \frac{x^6}{2304}x^6 + \mathcal{O}(x^8),$$

$$n = 1, \quad y(0) = 0, \quad y'(0) = \frac{1}{2} \quad \text{mit der Lösung} \quad y(x) = J_1(x) = \frac{x}{2} - \frac{x^3}{16} + \frac{x^5}{384} + \mathcal{O}(x^7),$$

$$n = 0, 1, \quad y(0) = 0, \quad y'(0) = 0 \quad \text{mit der Lösung} \quad y(x) = 0,$$

$$n = 0, 1, 2, \quad y(0) = 0, \quad y'(0) = 0, \quad y''(0) = 0 \quad \text{mit der Lösung} \quad y(x) = 0,$$

$$n = 0, 1, 2, 3, \quad y(0) = 0, \quad y'(0) = 0, \quad y''(0) = 0, \quad y'''(0) = 0 \quad \text{mit der Lösung} \quad y(x) = 0.$$

Die AB $y(0) = 0, y'(0) = 0$ würde aber auch zu den Funktionen $J_n(x), n = 2, 3, \dots$, passen, denn es gilt $J_n^{(m)}(0) = 0, m = 0(1)n-1, n \geq 2$. Damit kommt man zur nächsten Situation.

(c) Da die Funktion $J_n(x), n \geq 2$, und ihre anfänglichen Ableitungen bei $x = 0$ verschwinden, kann zur gDGL (1) mit der AB $y(0) = y'(0) = y^{(m)}(0) = 0, m \geq 1$, nur die parameterabhängige Lösung

$$y(x) = c J_n(x), n = m + 1, m + 2, \dots$$

bestimmt werden. Für $m = 1$ liefert Maple die Formel

$$y(x) = c \frac{2J_1(x) - xJ_0(x)}{x},$$

die wegen der Rekursionsbeziehungen einfach $cJ_2(x)$ ist. Natürlich kann manchmal die Darstellung mit der Division durch x problematisch sein.

Eindeutig lösbare AWP sind aber z.B.

$$n = 2, y(0) = y'(0) = 0, y''(0) = \frac{1}{4} \text{ mit der Lösung } y(x) = J_2(x) = \frac{x^2}{8} - \frac{x^4}{96} + \mathcal{O}(x^6),$$

$$n = 3, y(0) = y'(0) = y''(0) = 0, y'''(0) = \frac{1}{8} \text{ mit der Lösung } y(x) = J_3(x),$$

$$n = 4, y(0) = y'(0) = y''(0) = y'''(0) = 0, y^{(4)}(0) = \frac{1}{16} \text{ mit der Lösung } y(x) = J_4(x).$$

Die Besonderheit an der Stelle $x = 0$ ist die Ursache dafür, dass eine gDGL 2. Ordnung mit einer größeren Anzahl von AB gelöst werden kann.

(d) AWP, bei denen keine Lösung erzeugt wird, sind solche, wo in der gDGL (1) der Parameter n nicht festgelegt ist, bzw. x_0 und die AB nicht zur allgemeinen Lösung passen wie z.B. für

$$n = \frac{1}{2}, 1, 2, \quad y(0) = 1, y'(0) = 0,$$

$$n = \frac{1}{2}, 2, \quad y(0) = 0, y'(0) = \frac{1}{2},$$

$$n = 0, \frac{1}{2}, 1, 2, \quad y(0) = 1, y'(0) = \frac{1}{2},$$

$$n = 0, 1, \quad y(0) = y'(0) = 0, y''(0) \neq 0,$$

$$n = 0, 1, 2, \quad y(0) = y'(0) = y''(0) = 0, y'''(0) \neq 0.$$

Bezüglich der Numerik verwenden wir `dsolve/numeric` mit der hinreichend genauen Methode `rkf45`. Betrachtet werden verschiedene Möglichkeiten der Ergebniserzeugung und -verarbeitung mit den Ausgabeoptionen `procedurelist` (Standardeinstellung) und `listprocedure`. Die Verfahren verlangen für eine eindeutige Lösung der gDGL (1) zweiter Ordnung zunächst genau die zwei AB $y(x_0) = y_0, y'(x_0) = y'_0$. Mehr AB wie bei Spezialfällen der exakten Lösung gemäß Teil (b) und (c) sind nicht zugelassen.

Unpassende AB, wie schon bei `dsolve` gesehen, ergeben auch keine Lösung. Dazu gehören die AB $y(0) = y_0, y'(0) = y'_0$, die sich nicht mit dem Parameter n vertragen. Diese führen bei der numerischen Lösung zur Fehlermeldung

```
Error, (in dsolve/numeric/make_proc)
ode system is singular at the initial point
```


Die Ursache liegt hauptsächlich in der Singularität im Punkt $x = 0$ des Anteils $Y_n(x)$ der exakten Lösung.

Sehr genaue numerische Lösungen erhält man z.B. zu AWP mit den AB

$$n = 0, \quad y(0) = 1, \quad y'(0) = 0,$$

$$n = 1, \quad y(0) = 0, \quad y'(0) = \frac{1}{2},$$

$$n = 1, \quad y(1) = 1, \quad y'(1) = 1,$$

keine numerische Lösungen zu AWP mit den AB

$$n = 0, \quad y(0) = 1, \quad y'(0) = 0, \quad y''(0) = -\frac{1}{2} \quad (\text{exakte Lösung ist } J_0(x)),$$

$$n = 1, \quad y(0) = 0, \quad y'(0) = \frac{1}{2}, \quad y''(0) = 0 \quad (\text{exakte Lösung ist } J_1(x)),$$

$$n = 3, \quad y(0) = y'(0) = y''(0) = 0, \quad y'''(0) = \frac{1}{8} \quad (\text{exakte Lösung ist } J_3(x)).$$

Betrachten wir ausgewählte Maple-Anweisungen und grafischen Ergebnisse zum AWP mit $n = 0$, $y(0) = 1$, $y'(0) = 0$. Verwendet werden u.a. die Kommandos `dsolve`, `dsolve/numeric`, `rkf45`, `output=procedurelist`, `odeplot`.

```
> # gdG1 und allgemeine Loesung
  dg11:=x^2*diff(y(x),x$2)+x*diff(y(x),x)+(x^2-n^2)*y(x)=0: # Digits=16
  odeadvisor(dg11):                # bestimmt den Typ der gdG1
  dsolve(dg11,y(x)):
> # AWP und Loesung
  n:=0; init1:=y(0)=1,D(y)(0)=0;
  dsol1:=dsolve([dg11,init1],y(x));
  fy:=unapply(rhs(dsol1),x);      # Loesung als Funktion
> # AWP und numerische Loesung
  lsg1a:=dsolve([dg11,init1],y(x),numeric,method=rkf45,range=0..10,
               output=procedurelist); # default
  lsg1a(0);                        # Kontrolle der AB
> # Grafik mit odeplot
  tf3:=font=[HELVETICA,BOLD,20]:
  # pl1:=odeplot(lsg1a,[[x,y(x)],[x,diff(y(x),x)]],x=0..10,color=black,
  pl1:=odeplot(lsg1a,[[x,y(x)],[x,D(y)(x)]],x=0..10,color=black,
               linestyle=[solid,dash],thickness=3,view=[0..10,-1..1],
               labels=[' x', ' ']):
  pl2:=pointplot([[0,1],[0,0]],color=black,symbol=circle,symbolsize=24):
  pl3:=textplot([[2.3,0.6,' y(x)'],[2,-0.7,' y'(x)']],tf3):
  display(pl1,pl2,pl3);
> pl1:=odeplot(lsg1a,[y(x),diff(y(x),x)],x=0..10,color=black,
  # pl1:=odeplot(lsg1a,[y(x),D(y)(x)],x=0..10,color=black,
               numpoints=500,thickness=3,view=[-0.5..1,-1..0.5]):
  pl2:=pointplot([[1,0]],color=black,symbol=solidcircle,symbolsize=24):
  display(pl1,pl2);
> pl1:=odeplot(lsg1a,[x,y(x),D(y)(x)],x=0..10,color=black,
               thickness=2,view=[0..10,-0.5..1,-1..0.5],axes=boxed,
               orientation=[30,70],tickmarks=[3,10,3]):
  pl2:=pointplot3d([[0,1,0]],color=black,symbol=solidcircle,symbolsize=24):
  display(pl1,pl2);
```

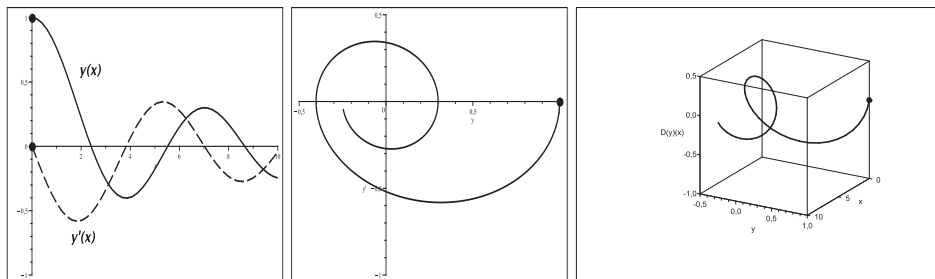


Abb. 2.310 Datei *spez_dgl_01b1..3.ps*, gDGl (1), AB $y(0) = 1$, $y'(0) = 0$,
 exakte Lösung $y(x) = J_0(x)$,
 v.l.n.r. Näherungslösungen zu $y(x)$, $y'(x)$, $x \in [0, 10]$,
 und zu den Phasenkurven $(y(x), y'(x))$ und $(x, y(x), y'(x))$, $x \in [0, 10]$,
 Kommandos `dsolve/numeric/procedurelist`, `odeplot`

Es folgen noch Bemerkungen zu den Ausgabeoptionen im Kommando `dsolve/numeric`. Das Ergebnis kann in jedem Fall für die grafische Ausgabe mit `odeplot` verwendet werden. Man vergleiche die Vorgehensweise mit dem `SysgDGl` (3.1) im Abschnitt 2.6.

(a) `output=procedurelist`

```
> lsg1a:=dsolve([dgl1,init1],y(x),numeric,method=rkf45,range=0..10,
               output=procedurelist); # default
> # Funktionsdefinition und deren Verwendung
yy:= x->rhs(op(2,lsg1a(x)));
yys:=x->rhs(op(3,lsg1a(x)));
# Plot mit Operatorform
p11:=plot([yy,yys],0..10,color=black,linestyle=[solid,dash],
          numpoints=1000,thickness=3,view=[0..10,-1..1]):
p12:=pointplot([[0,1],[0,0]],color=black,symbol=solidcircle,symbolsize=24):
display(p11,p12);
```

(b) `output=listprocedure`

Zur Weiterverarbeitung haben wir die Varianten `subs`, `eval`, `assign`.

```
> lsg2a:=dsolve([dgl1,init1],y(x),type=numeric,output=listprocedure);
> # Grafik mit odeplot
p11:=odeplot(lsg2a,[[x,y(x)],[x,diff(y(x),x)]],x=0..10,color=black,
            linestyle=[solid,dash],numpoints=500,thickness=3,view=[0..10,-1..1]):
p12:=pointplot([[0,1],[0,0]],color=black,symbol=solidcircle,symbolsize=24):
display(p11,p12);
> # subs
x1:=subs(lsg2a,x);
yy1:=subs(lsg2a,y(x));
yy1s:=subs(lsg2a,diff(y(x),x));
yy1(0),yy1s(0): # Kontrolle der AB
> p11:=plot([yy1(x),yy1s(x)],x=0..10,color=black,linestyle=[solid,dash],
           numpoints=500,thickness=3,view=[0..10,-1..1]):
```

```

display(p11,p12);
> # eval
yy2:=eval(y(x),lsg2a);
yy2s:=eval(diff(y(x),x),lsg2a);
yy2(0),yy2s(0):
p11:=plot([yy2(x),yy2s(x)],x=0..10,color=black,linestyle=[solid,dash],
          numpoints=500,thickness=3,view=[0..10,-1..1]):
display(p11,p12);
> # assign(subs(y(x)=yy3,diff(y(x),x)=yy3s,lsg2a)); # Fehler
assign(subs(y(x)=yy3,lsg2a[2]));
assign(subs(diff(y(x),x)=yy3s,lsg2a[3]));
# oder
assign(subs(y(x)=yy3,lsg2a[2]),subs(diff(y(x),x)=yy3s,lsg2a[3]));
yy3(0),yy3s(0):
# p11:=plot([yy3(x),yy3s(x)],x=0..10,color=black,linestyle=[solid,dash],
p11:=plot([yy3,yy3s],0..10,color=black,linestyle=[solid,dash],
          thickness=3,view=[0..10,-1..1]):
display(p11,p12);

```

(2) Riccati-DGl

$$y' + y^2 + \varepsilon y - 1 = 0, \quad \varepsilon \text{ reell,}$$

$$y(x) = -\frac{\varepsilon}{2} + \frac{\sqrt{4 + \varepsilon^2}}{2} \tanh\left(\frac{\sqrt{4 + \varepsilon^2}}{2}(x + c)\right)$$

mit dem Sonderfall $\varepsilon = 0$ und $y(x) = \tanh(x + c)$.

Zunächst wird auf den Abschnitt 2.2, Bsp. 2.4 Teil (20), verwiesen.

Bei der Funktion $\tanh(x)$ und ihrer inversen Funktion $\operatorname{artanh}(x)$ ist auf Definitionsbereich, Wertevorrat, Grenzfälle und Asymptoten zu achten. Es sind Rechnungen im Komplexen, aber mit reellen Ergebnissen, zu erwarten.

Wir bleiben beim Fall $\varepsilon = 0$.

Die gDGl und ihr überall definiertes Richtungsfeld lassen auf moderate Lösungsverläufe für $|y| < 1$ sowie auf die beiden stationären Lösungen $y(x) = \pm 1$ schließen. Für $|y| \gg 1$ gibt es sehr steile Elemente, damit Lösungswege mit vertikalen Asymptoten und dazu Funktionszweige auf der jeweils anderen Seite der Asymptote.

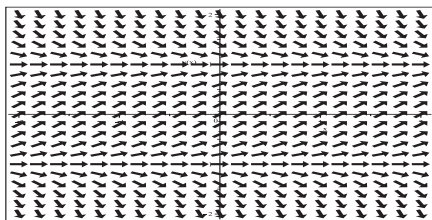


Abb. 2.311

Datei *spez_dgl_02a1.ps*,
 Richtungsfeld zur gDGl (2),
 Kommando `dfieldplot`

Näherungslösungen von AWP im Richtungsfeld werden mittels `phaseportrait` erzeugt. AB $y(0) = y_0$ mit $|y_0| > 1$ zeigen die Asymptotik, sodass die numerischen Verfahren nahe der Stellen der Asymptoten abbrechen.

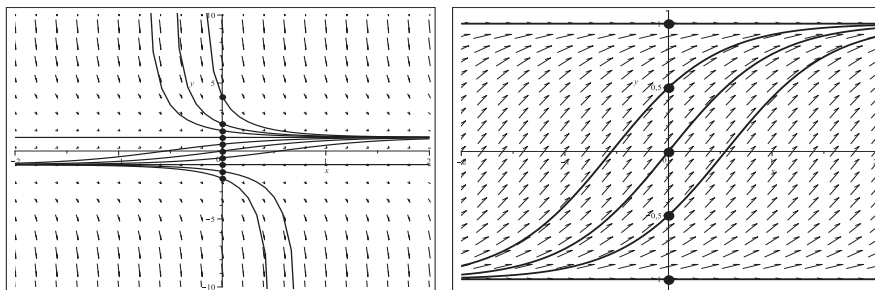


Abb. 2.312 Dateien *spez_dgl_02b1..2.ps*, gDGI (2),
 AB $y(0) = -2, -\frac{3}{2}, -1, -\frac{1}{2}, 0, \frac{1}{2}, 1, \frac{3}{2}, 2, 4$,
 Näherungslösungen in $[-2, 2] \times [-10, 10]$ bzw. $[-2, 2] \times [-1, 1]$,
 Kommandos `phaseportrait`, `pointplot`, Numerik mit `rkf45`

Die Lösung des AWP zur AB $y(0) = y_0$ gemäß `dsol:=dsolve([dgl12,y(0)=y0],y(x));`
`f:=unapply(rhs(dsol),x);` ist

$$y(x) = \tanh(x + \operatorname{artanh}(y_0)).$$

Für $|y_0| > 1$ werden die Rechnungen zwar im Komplexen durchgeführt, aber die Ergebnisse sind reell. Im Detail heißt dies Folgendes:

$y_0 > 1$: positive Lösung in $(-\Re(\operatorname{artanh}(y_0)), \infty)$, negativer Zweig in $(-\infty, -\Re(\operatorname{artanh}(y_0)))$,

$y_0 < -1$: negative Lösung in $(-\infty, -\Re(\operatorname{artanh}(y_0)))$, positiver Zweig in $(-\Re(\operatorname{artanh}(y_0)), \infty)$,

$|y_0| = 1$: reelle Grenzfälle $y(x) = \pm 1$,

$|y_0| < 1$: Rechnung im Reellen.

Zu den AB in Abb. 2.312 zeigen wir die exakten Lösungskurven, Asymptoten und Zusatzzweige.

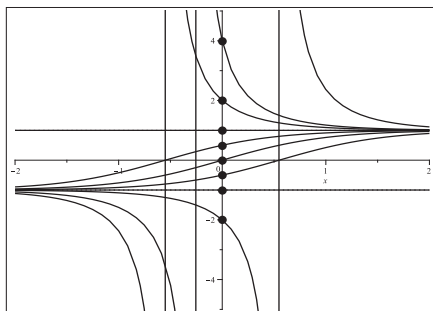


Abb. 2.313

Datei *spez_dgl_02c1.ps*,

Lösungen von AWP zur gDGI (2),

AB $y(0) = y_0 = -2, -\frac{3}{2}, -1, -\frac{1}{2}, 0, \frac{1}{2}, 1, \frac{3}{2}, 2, 4$,

und Funktionszweige bei $|y_0| > 1$,

Kommandos `dsolve`, `plot`, `pointplot`

Die Stelle der Asymptote ist $z = -\Re(\operatorname{artanh}(y_0)) = \Re(\operatorname{artanh}(-y_0))$, $|y_0| > 1$, speziell

$$y_0 = 4: \quad z = -\frac{1}{2} \ln\left(\frac{5}{3}\right) = -0.255412811,$$

$$y_0 = \pm 2: \quad z = \mp \frac{1}{2} \ln(3) = \mp 0.549306144.$$

(3) Legendre-DGI

$$(1-x^2)y'' - 2xy' + n(n+1)y = ((1-x^2)y')' + n(n+1)y = 0, \quad n \geq 0, \text{ ganzzahlig,}$$

$$y(x) = c_1 P_n(x) + c_2 Q_n(x)$$

mit den Legendre-Funktionen 1. Art (Legendre-Polynome) $P_n(x)$ und 2. Art $Q_n(x)$.

Als SysgDGI notiert man

$$y' - z = 0,$$

$$(1-x^2)z' - 2xz + n(n+1)y = 0.$$

Die Lösungen $y(x)$, $z(x)$ beinhalten die hypergeometrischen Funktionen, deren spezielle Varianten eben die Legendre-Funktionen sind.

Aus der Gestalt der gDGI ergeben sich einige Aspekte. Zunächst degeneriert die gDGI bei $|x| = 1$ und man muss mit Singularitäten und Beschränkungen des Definitionsbereichs der Lösung rechnen. Dann wird eine Fallunterscheidung zwischen $n = 0$ und $n > 0$ sinnvoll sein. Für $n = 0$ erkennt man die konstante Lösung $y(x) = c = cP_0(x)$ zur AB $y(x_0) = c$, $y'(x_0) = 0$, für $n > 0$ die Nulllösung zur AB $y(x_0) = y'(x_0) = 0$ sowie für $n = 1$ die Lösung $y(x) = cx = cP_1(x)$ zur AB $y(0) = 0$, $y'(0) = c$. Dabei könnte die Stelle x_0 sogar beliebig sein, sodass über die Singularität hinweg integriert wird. Dann stellt sich die Frage, wie sich Näherungsverfahren in solchen Situationen verhalten.

In der obigen Lösung sind die beiden Legendre-Funktionen enthalten, die orthogonalen Legendre-Polynome

$$P_n(x) = \frac{1}{2^n} \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{(-1)^k (2n-2k)!}{k! (n-k)! (n-2k)!} x^{n-2k} = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2-1)^n, \quad n = 0, 1, 2, \dots,$$

mit ihren vielen Eigenschaften, u.a. erfüllen sie die Rekursionsbeziehungen (3-Term-Rekursion)

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \quad P_0(x) = 1, \quad P_1(x) = x, \quad n > 1,$$

sowie die weniger bekannten Legendre-Funktionen 2. Art für $|x| < 1$

$$Q_0(x) = \frac{1}{2} \ln \left(\frac{1+x}{1-x} \right) = \ln \left(\sqrt{\frac{1+x}{1-x}} \right) = -\ln \left(\sqrt{\frac{1-x}{1+x}} \right) = \operatorname{artanh}(x),$$

$$Q_1(x) = \frac{x}{2} \ln \left(\frac{1+x}{1-x} \right) - 1 = x \operatorname{artanh}(x) - 1,$$

$$Q_2(x) = \frac{3x^2-1}{4} \ln \left(\frac{1+x}{1-x} \right) - \frac{3x}{2} = \frac{3}{2} \left(\left(x^2 - \frac{1}{3} \right) \operatorname{artanh}(x) - x \right),$$

$$Q_3(x) = \frac{5x^3-3x}{4} \ln \left(\frac{1+x}{1-x} \right) - \frac{5x^2}{2} + \frac{2}{3} = \frac{5}{2} \left(\left(x^2 - \frac{3}{5} \right) x \operatorname{artanh}(x) - x^2 + \frac{4}{15} \right),$$

$$Q_4(x) = \frac{35}{8} \left(\left(x^4 - \frac{6}{7}x^2 + \frac{3}{35} \right) \operatorname{artanh}(x) - x \left(x^2 - \frac{11}{21} \right) \right),$$

$$Q_5(x) = \frac{63}{8} \left(\left(x^4 - \frac{10}{9}x^2 + \frac{5}{21} \right) x \operatorname{artanh}(x) - x^4 + \frac{7}{9}x^2 - \frac{64}{945} \right).$$

Für die Legendre-Funktionen 2. Art gilt dieselbe Rekursionsformel

$$(n+1)Q_{n+1}(x) = (2n+1)xQ_n(x) - nQ_{n-1}(x), \quad n > 1,$$

$$Q_0(x) = \operatorname{artanh}(x), \quad Q_1(x) = x \operatorname{artanh}(x) - 1.$$

Maple enthält in der Bibliothek `orthopoly` u.a. die Legendre-Polynome $P_n(x)$.

Es gibt aber auch die beiden Legendre-Funktionen `LegendreP(n,x)` und `LegendreQ(n,x)`.

Erstere ist identisch mit $P_n(x)$. Aber die zweite liefert

$$\operatorname{LegendreQ}(0,x) = \frac{1}{2} \ln(x+1) - \frac{1}{2} \ln(x-1) = \ln\left(\sqrt{\frac{x+1}{x-1}}\right)$$

$$= \ln\left(\sqrt{\frac{1+1/x}{1-1/x}}\right) = \operatorname{artanh}\left(\frac{1}{x}\right) = Q_0\left(\frac{1}{x}\right),$$

$$\operatorname{LegendreQ}(1,x) = x \operatorname{artanh}\left(\frac{1}{x}\right) - 1 = x^2 \left(Q_1\left(\frac{1}{x}\right) + 1\right) - 1,$$

$$\operatorname{LegendreQ}(2,x) = \frac{3}{2} \left(\left(x^2 - \frac{1}{3}\right) \operatorname{artanh}\left(\frac{1}{x}\right) - x \right),$$

$$\operatorname{LegendreQ}(3,x) = \frac{5}{2} \left(\left(x^2 - \frac{3}{5}\right) x \operatorname{artanh}\left(\frac{1}{x}\right) - x^2 + \frac{4}{15} \right),$$

$$\operatorname{LegendreQ}(4,x) = \frac{35}{8} \left(\left(x^4 - \frac{6}{7}x^2 + \frac{3}{35}\right) \operatorname{artanh}\left(\frac{1}{x}\right) - x \left(x^2 - \frac{11}{21}\right) \right),$$

$$\operatorname{LegendreQ}(5,x) = \frac{63}{8} \left(\left(x^4 - \frac{10}{9}x^2 + \frac{5}{21}\right) x \operatorname{artanh}\left(\frac{1}{x}\right) - x^4 + \frac{7}{9}x^2 - \frac{64}{945} \right).$$

Abweichend von $Q_n(x)$ ist der Definitionsbereich von $\operatorname{LegendreQ}(n,x)$ eigentlich $|x| > 1$.

In diesem Bereich sind die Legendre-Funktionen 2. Art reellwertig, bei $|x| = 1$ nicht definiert (Fehlermeldung), sonst komplexwertig, z.B. gilt $\operatorname{LegendreQ}(0, \frac{1}{2}) = \frac{\ln(3)}{2} - \frac{\pi}{2}i$.

Ausnahme bilden wegen $\operatorname{artanh}(\pm\infty) = \mp \frac{1}{2}i\pi$ die reellen Werte $\operatorname{LegendreQ}(2k+1, 0)$.

Für die Funktion $Q_n(x)$ wird eine Maple-Prozedur erstellt.

```
> Q:=proc(n,x) option remember;
  if n=0 then arctanh(x)
  elif n=1 then x*arctanh(x)-1
  elif not type(n,integer) then Q(n,x)
  else expand((2*n-1)/n*x*Q(n-1,x)-(n-1)/n*Q(n-2,x))
  end if
end proc;
```

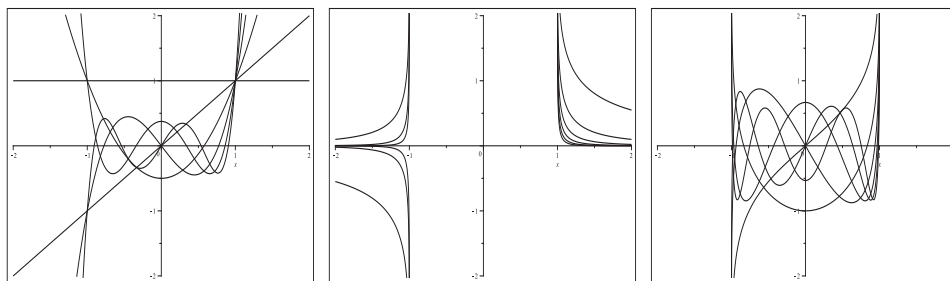


Abb. 2.314 Dateien `spez_dgl_03a1..3.ps`, Legendre-Funktionen in $[-2, 2] \times [-2, 2]$, v.l.n.r. $P_n(x) \equiv \operatorname{LegendreP}(n,x)$, $\operatorname{LegendreQ}(n,x)$, $Q_n(x)$, $n = 0, 1, \dots, 5$, Kommando `plot`

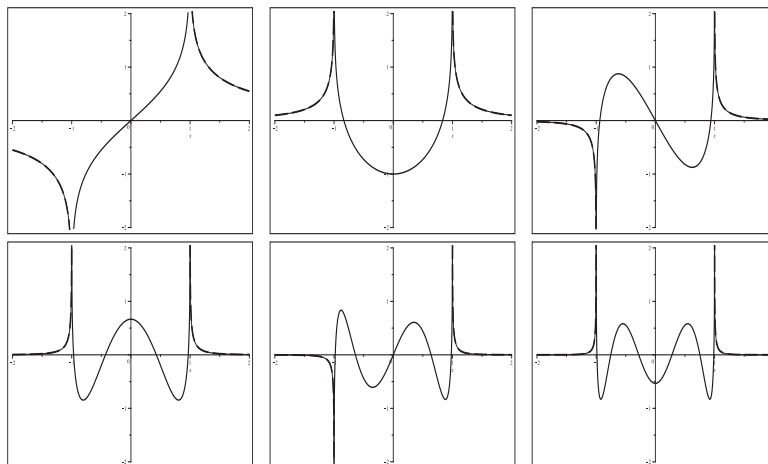


Abb. 2.315 Dateien *spez_dgl_03a4..9.ps*, Legendre-Funktionen 2. Art, $x \in [-2, 2]$, v.l.o.n.r.u. $\text{LegendreQ}(n, x)$ (dash) und $Q_n(x)$ (solid), $n = 0, 1, \dots, 5$, Asymptotik der Kurven bei $x = \pm 1$, Kommando `plot`

Weil die gDGL von 2. Ordnung und nicht autonom ist, wird kein Richtungsfeld erzeugt.

In Maple ist die allgemeine parameterabhängige Lösung

$$y(x) = c_1 \text{LegendreP}(n, x) + c_2 \text{LegendreQ}(n, x).$$

Für das AWP mit den AB $y(x_0) = y_0$, $y'(x_0) = y'_0$ erhält man Folgendes.

```
> Digits:=16:
dgl3:=(1-x^2)*diff(y(x),x$2)-2*x*diff(y(x),x)+n*(n+1)*y(x)=0;
dsolve(dgl3,y(x));
> dsol3:=dsolve([dgl3,y(x0)=y0,D(y)(x0)=y0s],y(x)):
fa:=unapply(rhs(dsol3),x,n,x0,y0,y0s);

fa := (x, n, x0, y0, y0s) ->
((- y0 n LegendreQ(n + 1, x0) - y0 LegendreQ(n + 1, x0) - LegendreQ(n, x0) y0s + LegendreQ(n, x0) y0s x0^2
+ LegendreQ(n, x0) n x0 y0 + LegendreQ(n, x0) x0 y0) LegendreP(n, x)
/ (- n LegendreP(n, x0) LegendreQ(n + 1, x0) - LegendreP(n, x0) LegendreQ(n + 1, x0)
+ n LegendreP(n + 1, x0) LegendreQ(n, x0) + LegendreP(n + 1, x0) LegendreQ(n, x0))
- ((- LegendreP(n, x0) y0s + LegendreP(n, x0) y0s x0^2 + n x0 LegendreP(n, x0) y0
- LegendreP(n + 1, x0) y0 - n LegendreP(n + 1, x0) y0 + x0 LegendreP(n, x0) y0) LegendreQ(n, x)
/ (- LegendreP(n, x0) n LegendreQ(n + 1, x0) - LegendreP(n, x0) LegendreQ(n + 1, x0)
+ n LegendreP(n + 1, x0) LegendreQ(n, x0) + LegendreP(n + 1, x0) LegendreQ(n, x0))
```

Für die Auswertung der Funktion $\text{fa}(x, n, x_0, y_0, y_0s)$ werden wegen $\text{LegendreQ}(n, x)$ die Rechnungen im Komplexen gemacht. Situationen mit Imaginärteil gleich Null bei exakter Bestimmung führen jedoch entsprechend der Gleitkommaarithmetik auf Werte nahe Null.

Bei diesem Beispiel ist interessant, wie Maple mit komplexen Formeln umgeht, deren Imaginärteil betragsmäßig "klein" ist.

Je nach Größe dieser muss man bei der grafischen Darstellung des Ergebnisses auf die “Ersatzfunktionen“ $\text{simplify}(\text{fa}(x, n, x_0, y_0, y_0s))$ bzw. wenn der Imaginärteil etwas größer ist, $\text{Re}(\text{fa}(x, n, x_0, y_0, y_0s))$ zurückgreifen, um diesen auszublenden.

Es empfehlen sich weitere Vorbetrachtungen und dann eine Fallunterscheidung bzgl. der Bereiche.

(a) Die Berechnung der Lösungsfunktion $fa(x, n, x_0, y_0, y_0')$ mit $|x_0| = 1$ sowie ihre Auswertung an den Stellen $x = \pm 1$ macht wegen der Singularität von $\text{LegendreQ}(n, x)$ wenig Sinn. Man erhält Fehlermeldungen wie **Error, (in LegendreQ) numeric exception: division by zero, Float(undefined)** oder **Float(∞)**.

(b) Die exakte Lösung des AWP mit den AB $y(x_0) = P_n(x_0)$, $y'(x_0) = P'_n(x_0)$ ist $y(x) = P_n(x)$. Die Ergebnisse mit Maple haben nicht diese einfache Formel, sondern eine komplexe Darstellung, sind aber in der Auswertung reell bzw. “fast reell“.

```
> # dsol3:=dsolve([dgl3,y(x0)=LegendreP(n,x0),
#                D(y)(x0)=D[2](LegendreP)(n,x0)],y(x)):
dsol3:=dsolve([dgl3,y(x0)=P(n,x0),D(y)(x0)=D[2](P)(n,x0)],y(x)):
fh:=unapply(rhs(dsol3),x,n,x0):
simplify(fh(x,n,x0));      # keine Vereinfachung, mit LegendreQ
> k:=0;                    # konkreter Wert, k=0,1,2,3,4,5
expand(LegendreP(k,x), LegendreP(k,0), D[2](LegendreP)(k,0));
fh(x,k,x0);
simplify(fh(x,k,x0));     # etwas vereinfacht und LegendreQ -> ln(x+-1)
simplify(fh(x,k,0));     # Vereinfachung wegen x0=0
convert(evalf(%),rational);
evalf(fh(x,k,0));
simplify(evalf(fh(x,k,0))); # kuerzeste Darstellung
```

$fh := (x, n, x_0) \rightarrow$

$$\begin{aligned} & \left((-P(n, x_0) n \text{LegendreQ}(n+1, x_0) - P(n, x_0) \text{LegendreQ}(n+1, x_0) - \text{LegendreQ}(n, x_0) D_2(P)(n, x_0) \right. \\ & + \text{LegendreQ}(n, x_0) D_2(P)(n, x_0) x_0^2 + \text{LegendreQ}(n, x_0) n x_0 P(n, x_0) + \text{LegendreQ}(n, x_0) x_0 P(n, x_0)) \text{LegendreP}(n, x) \\ & \left. / (-n \text{LegendreP}(n, x_0) \text{LegendreQ}(n+1, x_0) - \text{LegendreP}(n, x_0) \text{LegendreQ}(n+1, x_0) \right. \\ & \left. + n \text{LegendreP}(n+1, x_0) \text{LegendreQ}(n, x_0) + \text{LegendreP}(n+1, x_0) \text{LegendreQ}(n, x_0)) \right) \\ & - \left((-\text{LegendreP}(n, x_0) D_2(P)(n, x_0) + \text{LegendreP}(n, x_0) D_2(P)(n, x_0) x_0^2 + n x_0 \text{LegendreP}(n, x_0) P(n, x_0) \right. \\ & - \text{LegendreP}(n+1, x_0) P(n, x_0) - n \text{LegendreP}(n+1, x_0) P(n, x_0) + x_0 \text{LegendreP}(n, x_0) P(n, x_0)) \text{LegendreQ}(n, x) \\ & \left. / (-\text{LegendreP}(n, x_0) n \text{LegendreQ}(n+1, x_0) - \text{LegendreP}(n, x_0) \text{LegendreQ}(n+1, x_0) \right. \\ & \left. + n \text{LegendreP}(n+1, x_0) \text{LegendreQ}(n, x_0) + \text{LegendreP}(n+1, x_0) \text{LegendreQ}(n, x_0)) \right) \end{aligned}$$

$k := 0$

1, 1, 0

$$\frac{-\text{LegendreQ}(1, x_0) - \text{LegendreQ}(0, x_0) D_2(P)(0, x_0) + \text{LegendreQ}(n, x_0) D_2(P)(0, x_0) x_0^2 + \text{LegendreQ}(0, x_0) x_0}{-\text{LegendreQ}(1, x_0) + \text{LegendreQ}(0, x_0) x_0}$$

$$-\frac{(-D_2(P)(0, x_0) + D_2(P)(0, x_0) x_0) \text{LegendreQ}(0, x)}{-\text{LegendreQ}(1, x_0) + \text{LegendreQ}(0, x_0) x_0}$$

$$1 - \frac{1}{2} D_2(P)(0, x_0 \ln(x_0 + 1)) + \frac{1}{2} D_2(P)(0, x_0 \ln(x_0 - 1)) + \frac{1}{2} D_2(P)(0, x_0^2 \ln(x_0 + 1)) - \frac{1}{2} D_2(P)(0, x_0^2 \ln(x_0 - 1)) \\ + \frac{1}{2} D_2(P)(0, x_0 \ln(x + 1)) - \frac{1}{2} D_2(P)(0, x_0 \ln(x - 1)) + \frac{1}{2} D_2(P)(0, x_0^2 \ln(x + 1)) - \frac{1}{2} D_2(P)(0, x_0^2 \ln(x - 1))$$

$$1 + \frac{1}{2} i \pi D_2(P)(0, 0) + \frac{1}{2} D_2(P)(0, 0) \ln(x + 1) - \frac{1}{2} D_2(P)(0, 0) \ln(x - 1)$$

1

$$1.0000000000000000 - 0.1$$

1.

Für weitere Parameter k bei $x_0 = 0$ zeigen wir zum Vergleich die letzten vier Resultate.
 $k = 1$

$$D_2(P)(1, 0) x$$

$$x$$

$$1.000000000000000 x$$

$$x$$

$k = 2$

$$-\frac{1}{2} + \frac{3}{4} D_2(P)(2, 0) x + \frac{3}{2} x^2 + \frac{1}{8} I \pi D_2(P)(2, 0) - \frac{3}{8} I \pi D_2(P)(0, 0) x^2$$

$$-\frac{1}{8} D_2(P)(2, 0) \ln(x+1) - \frac{1}{8} D_2(P)(2, 0) \ln(x-1) - \frac{3}{8} D_2(P)(2, 0) x^2 \ln(x+1) + \frac{3}{8} D_2(P)(2, 0) x^2 \ln(x-1)$$

$$-\frac{1}{2} + \frac{3}{2} x^2$$

$$(1.000000000000000 - 3.429203673205104 \cdot 10^{-37} I) \text{LegendreP}(2., x)$$

$$(1.000000000000000 - 3.429203673205104 \cdot 10^{-37} I) \text{LegendreP}(2., x)$$

$k = 3$

$$-\frac{1}{3} D_2(P)(3, 0) x (5x^2 - 3)$$

$$\frac{1}{2} x (5x^2 - 3)$$

$$(1.000000000000000 \text{LegendreP}(3., x))$$

$$(1.000000000000000 \text{LegendreP}(3., x))$$

$k = 4$

$$\frac{3}{8} - \frac{15}{4} x^2 + \frac{35}{4} x^4 + \frac{55}{64} D_2(P)(4, 0) x - \frac{105}{64} D_2(P)(4, 0) x^3 + \frac{9}{128} I \pi D_2(P)(4, 0) - \frac{45}{64} I \pi D_2(P)(4, 0) x^2 + \frac{105}{128} I \pi D_2(P)(4, 0) x^4$$

$$+ \frac{9}{128} D_2(P)(4, 0) \ln(x+1) - \frac{9}{128} D_2(P)(4, 0) \ln(x-1) - \frac{45}{64} D_2(P)(4, 0) x^4 \ln(x+1) + \frac{45}{64} D_2(P)(4, 0) x^4 \ln(x-1)$$

$$+ \frac{105}{128} D_2(P)(4, 0) x^4 \ln(x+1) - \frac{105}{128} D_2(P)(4, 0) x^4 \ln(x-1)$$

$$\frac{3}{8} - \frac{15}{4} x^2 + \frac{35}{4} x^4$$

$$(0.999999999999999 - 3.821902754903827 \cdot 10^{-37} I) \text{LegendreP}(4., x)$$

$$(0.999999999999999 - 3.821902754903827 \cdot 10^{-37} I) \text{LegendreP}(4., x)$$

$k = 5$

$$\frac{1}{15} D_2(P)(5, 0) x (63x^4 - 70x^2 + 15)$$

$$\frac{1}{8} x (63x^4 - 70x^2 + 15)$$

$$0.999999999999999 \text{LegendreP}(5., x)$$

$$0.999999999999999 \text{LegendreP}(5., x)$$

(c) Nun noch zu einem speziellen Fall für $k=5$ bei $x_0 = 1.5$, der AB $y(x_0) = P_5(x_0)$, $y'(x_0) = P_5'(x_0)$ und der Lösung $P_5(x) = \frac{63}{8}x^5 - \frac{35}{4}x^3 + \frac{15}{8}x$.

Mit `evalf(fh(x,k,x0))` erhält man

$$0.999999999999999 \text{LegendreP}(5., x) - 2.999999999999999 \cdot 10^{-12} \text{LegendreQ}(5., x).$$

Die Berechnung mittels `evalf([fh(0,k,x0), fh(0.5,k,x0)])` liefert

$$[1.599999999999999 \cdot 10^{-12} + 5.02654824574 \cdot 10^{-33} I, 0.0898437499983 + 4.23378697456 \cdot 10^{-13} I].$$

`simplify(fh(0,k,x0))` "verbannt" den Imaginärteil, falls dieser sehr nahe bei Null ist. Das ist hier mit der Größenordnung $\mathcal{O}(10^{-13})$ bei `Digits=16` nicht möglich.

Nur mit `Re(fh(0,k,x0))` bleibt der Realteil übrig, sodass zur Grafik das Kommando `plot(Re(fh(x,k,x0)), x=-2..2, y=-4..4, color=black, numpoints=1000)` verwendet werden muss.

```

> # Rechnung im Komplexen, Imaginarteil~0, Grafik
pp:=array(1..3, []):
opts:=thickness=5,color=black,numpoints=1000:
p11:=plot(simplify(fh(x,k,x0)),x=-2..2,y=-40..40,opts):
# hier wie plot(fh(x,k,x0),x=-2..2,y=-40..40,opts):
p12:=pointplot([x0,P(k,x0)],symbol=solidcircle,symbolsize=24,color=black):
pp[1]:=display(p11,p12):
p11:=plot(Re(fh(x,k,x0)),x=-2..2,y=-40..40,opts):
pp[2]:=display(p11,p12):
p11:=plot(Re(fh(x,k,x0)),x=-2..2,y=-4..4,opts):
pp[3]:=display(p11)

```

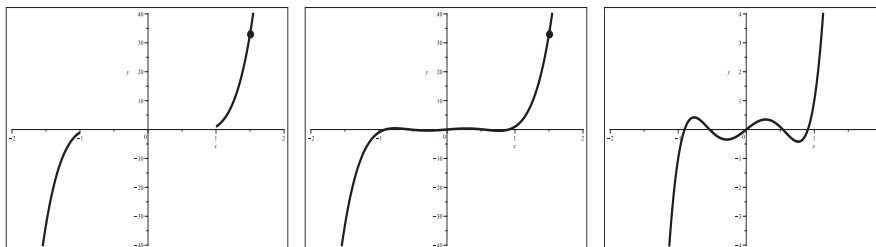


Abb. 2.316 Dateien *spez_dgl_03b1..3.ps*, gDGI (1), AWP mit $x_0 = 1.5$ und $AB\ y(x_0) = P_5(x_0)$, $y'(x_0) = P'_5(x_0)$, Lösung $P_5(x)$ gemäß $fh(x, k, x_0)$, Grafik mit $simplify(fh(0, k, x_0))$ bzw. $Re(fh(x, k, x_0))$ in $[-2, 2]$, Kommandos `dsolve`, `plot`, `pointplot`

Es ändert sich an dieser Situation nichts Wesentliches bei Nutzung von $fa(x, n, x_0, y_0, y'_0)$. Mit `fa(x, k, x0, P(k, x0), evalf(D[2](P)(k, x0)))` erhält man

$$1.00000000000 \text{LegendreP}(5, x) - 9.99999999999 \cdot 10^{-13} \text{LegendreQ}(5, x).$$

Die Berechnung mittels `evalf([fa(0, k, x0, P(k, x0), evalf(D[2](P)(k, x0)))]` und `fa(0.5, k, x0, P(k, x0), evalf(D[2](P)(k, x0)))` liefert

$$[5.33333333333 \cdot 10^{-13} + 1.67551608191 \cdot 10^{-33} I, 0.08984374999 + 1.41126232485 \cdot 10^{-13} I].$$

Die Grafik dazu entspricht der Abb. 2.316.

(d) In den Formeln der Lösungen von AWP tauchen Funktionen mit Singularitäten bzw. komplexen Werten auf. Die grafische Darstellung der Lösung wird von einer einzelnen Stelle jedoch nicht beeinträchtigt. Vergleichbar ist dies mit der Situation, wenn z.B. die Funktion $\frac{\sin(x)}{x}$ gezeichnet wird. Die Belegung dieser Funktion an der Stelle $x = 0$ mit Werten wie 1 , $infinity$, $1/x$, $\cot(x)$, $\ln(-1)$ ist ohne negative Auswirkung. Zu vermeiden sind jedoch Ausdrücke wie $1/0$ oder $\cot(0)$.

(e) Die Lösungsformeln enthalten den natürlichen Logarithmus sowie die artanh -Funktion, die im Reellen beschränkte Definitionsbereiche besitzen, singuläre Stellen bzw. Polstellen haben und auch im Komplexen berechnet werden können. Bei Umformungen ist wegen $\ln(zi) = \ln(z) + \frac{\pi}{2}i$, $z > 0$, auf all dies besonders zu achten.

Beim Vergleich der beiden Funktionen $\ln(x+1) - \ln(x-1)$ und $\ln\left(\frac{x+1}{x-1}\right)$ erhält man gleiche reelle Funktionswerte für $|x| > 1$, obwohl für $x < -1$ die erste im Komplexen auszuwerten ist. Für $|x| < 1$ folgen verschiedene komplexe Größen mit der Beziehung $\ln(x+1) - \ln(x-1) - \ln\left(\frac{x+1}{x-1}\right) + 2\pi i = 0$.

Die drei Funktionen $g_1(x) = \frac{\pi}{2}i - \frac{1}{2}(\ln(x-1) - \ln(x+1))$, $g_2(x) = \ln\left(\sqrt{\frac{1+x}{1-x}}\right)$ und $g_3(x) = -\ln\left(\sqrt{\frac{1-x}{1+x}}\right)$ erfüllen die Beziehungen

$$g_1(x) = g_2(x) = \begin{cases} \alpha(x) & \text{für } |x| < 1, \\ \beta(x) + \frac{\pi}{2}i & \text{für } |x| > 1, \end{cases} \quad g_3(x) = \overline{g_2(x)},$$

mit den reellen Funktionen $\alpha(x) = g_2(x)$, $\beta(x) = \ln\left(\sqrt{\frac{x+1}{x-1}}\right)$ mit $\beta(x) = -\beta(-x)$.
Damit ergeben sich

$$g_2(x) + g_3(x) = 2\Re(g_1(x)) = \begin{cases} 2\alpha(x) & \text{für } |x| < 1, \\ 2\beta(x) & \text{für } |x| > 1, \end{cases}$$

$$g_2(x) - g_3(x) = 2\Im(g_1(x)) = \begin{cases} 0 & \text{für } |x| < 1, \\ \pi i & \text{für } |x| > 1. \end{cases}$$

Es ergibt sich nun Folgendes bei der Gegenüberstellung $g_1(x) = g_2(x)$, $x \in \mathbb{R}$.

```
|x|<1 -> g1(x) reell
           g1(x): Rechnung im Komplexen
           g2(x): Rechnung im Reellen

|x|>1 -> g1(x) komplex
x>1      g1(x): erst Rechnung im Reellen, dann Addition von Pi/2*I
           g2(x): Rechnung im Komplexen
x<-1     g1(x),g2(x): Rechnung im Komplexen
```

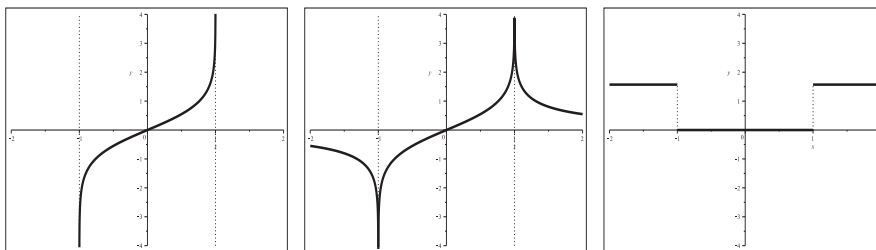


Abb. 2.317 Dateien *spez_dgl_03b4..6.ps*,
v.l.n.r. Grafiken mit $g_1(x)$, $\Re(g_1(x))$, $\Im(g_1(x))$, $x \in [-2, 2]$,
Kommando `plot`

Auf ähnliche Weise kann man die Funktionen $g_4(x) = \frac{1}{2}(\ln(x+1) - \ln(x-1))$ und $g_5(x) = \ln\left(\sqrt{\frac{x+1}{x-1}}\right)$ vergleichen.

(f) Es folgt die vergleichende Auswertung der Lösungsfunktion $fa(x, n, x_0, y_0, y'_0)$ mit verschiedenen AB sowie den Werten $n = 0, 1, 2$.

Für $|x_0| > 1$ werden neben der Lösungstrajektorie zur AB noch der andere Funktionszweig in $|x| > 1$ angezeigt. Für $n > 2$ "wiederholen" sich die Situationen, wobei gewisse Symmetrien/Asymmetrien sichtbar sind. Zusätzlich werden in den Abbildungen die Vertikalen $|x| = 1$ eingetragen sowie bei $n = 0, 1$ eventuell weitere Asymptoten.

Bzgl. der AB werden die 4 Fälle $(x_0, 1, 0)$, $(x_0, 1, 2)$, $(x_0, 1, -2)$, $(x_0, 0, 1)$ betrachtet.

1. Fall $x_0 > 1$, speziell $x_0 = 2$

$$\begin{aligned} n = 0, \quad f_{01}(x) &= 1, \quad x \in \mathbb{R}, \\ f_{02}(x) &= 1 + g_0(x), \quad |x| > 1, \\ f_{03}(x) &= 1 - g_0(x), \quad |x| > 1, \\ f_{04}(x) &= \frac{1}{2}g_0(x), \quad |x| > 1, \\ g_0(x) &= 3 \ln(3) - 3[\ln(x+1) - \ln(x-1)], \\ \lim_{x \rightarrow \pm\infty} (\ln(x+1) - \ln(x-1)) &= 0. \end{aligned}$$

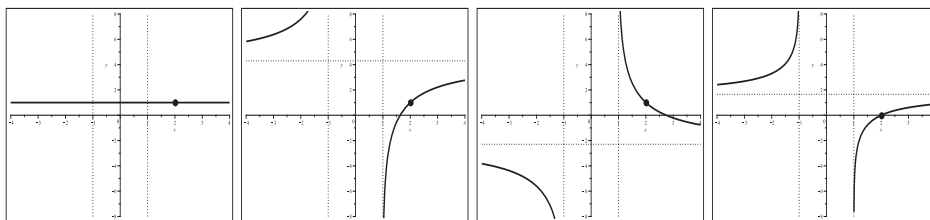


Abb. 2.318 Dateien *spez_dgl_03c101.4.ps*, AWP zur gDGI (3), $fa(x, 0, x_0, y_0, y'_0)$, v.l.n.r. Grafiken mit $f_{0,k}(x)$, $k = 1, 2, 3, 4$, $x \in [-4, 4]$, dazu Asymptoten, Kommandos `dsolve`, `plot`, `pointplot`

$$\begin{aligned} n = 1, \quad f_{11}(x) &= 2x - 3g_1(x), \quad |x| > 1, \\ f_{12}(x) &= -4x + 9g_1(x), \quad |x| > 1, \\ f_{13}(x) &= 8x - 15g_1(x), \quad |x| > 1, \\ f_{14}(x) &= -3x + 6g_1(x), \quad |x| > 1, \\ g_1(x) &= 1 + \frac{x \ln(3)}{2} - \frac{x}{2}[\ln(x+1) - \ln(x-1)], \\ \lim_{x \rightarrow \pm\infty} \frac{x}{2}(\ln(x+1) - \ln(x-1)) &= 1, \end{aligned}$$

$$\begin{aligned} n = 2, \quad f_{21}(x) &= -5 + 15x^2 + \frac{3}{2}g_2(x), \quad |x| > 1, \\ f_{22}(x) &= 4 - 12x^2 - \frac{5}{4}g_2(x), \quad |x| > 1, \\ f_{23}(x) &= -14 + 42x^2 + \frac{17}{4}g_2(x), \quad |x| > 1, \\ f_{24}(x) &= \frac{9}{2} - \frac{27}{2}x^2 - \frac{11}{8}g_2(x), \quad |x| > 1, \\ g_2(x) &= 3 \ln(3) - 18x - 9 \ln(3)x^2 - 3(1 - 3x^2)[\ln(x+1) - \ln(x-1)]. \end{aligned}$$

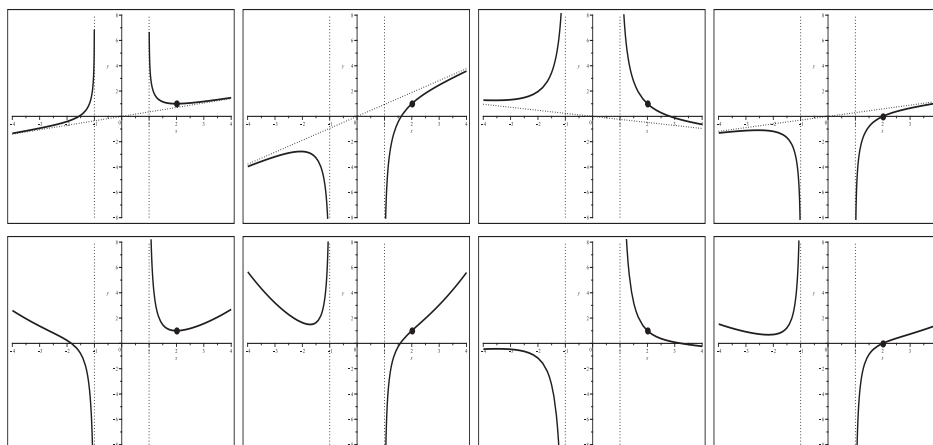


Abb. 2.319 Dateien *spez_dgl_03c111..4.ps*, *spez_dgl_c121..4.ps*, AWP zur gDGl (3), $fa(x, n, x_0, y_0, y'_0)$ für $n = 1, 2$, v.l.n.r. Grafiken mit $f_{1,k}(x)$, $k = 1, 2, 3, 4$, $x \in [-4, 4]$, dazu Asymptoten, bzw. mit $f_{2,k}(x)$, $k = 1, 2, 3, 4$, $x \in [-4, 4]$, Kommandos `dsolve`, `plot`, `pointplot`

2. Fall $x_0 < -1$, speziell $x_0 = -2$

Man bemerkt die Ähnlichkeit zur Situation im ersten Fall.

$$\begin{aligned} n = 0, \quad f_{01}(x) &= 1, \quad x \in \mathbb{R}, \\ f_{02}(x) &= 1 + \tilde{g}_0(x) = -2.295836866 - 3 \cdot \ln(x + 1.) + 3 \cdot \ln(x - 1.), \quad |x| > 1, \\ f_{03}(x) &= 1 - \tilde{g}_0(x) = 4.295836866 + 3 \cdot \ln(x + 1.) - 3 \cdot \ln(x - 1.), \quad |x| > 1, \\ f_{04}(x) &= \frac{1}{2} \tilde{g}_0(x) = -1.647918433 - 1.5 \ln(x + 1.) + 1.5 \ln(x - 1.), \quad |x| > 1, \\ \tilde{g}_0(x) &= -3 \ln(3) - 3[\ln(x + 1) - \ln(x - 1)] = g_0(x) - 6 \ln(3). \end{aligned}$$

Die Funktion $fa(x, 0, x_0, y_0, y'_0)$ ist nur mit $x_0 = -2.0$ auswert- und darstellbar.

Bei $fa(x, 0, -2, y_0, y'_0)$ erhält man die Fehlermeldung `Error, (in LegendreQ) numeric exception: division by zero`, bei $fa(x, 0, -2.0, y_0, y'_0)$ die Gleitkommadarstellung mit Dezimalzahlen, die oben mit angegeben wurde.

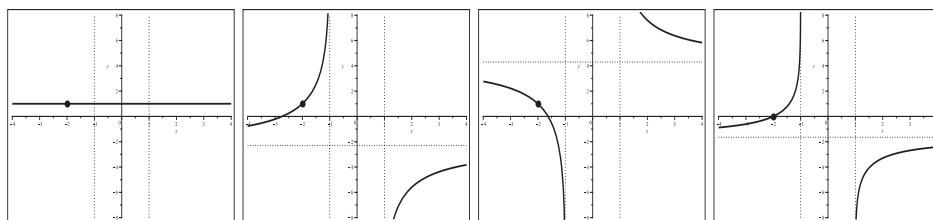


Abb. 2.320 Dateien *spez_dgl_03c201..4.ps*, AWP zur gDGl (3), $fa(x, 0, x_0, y_0, y'_0)$, v.l.n.r. Grafiken mit $f_{0,k}(x)$, $k = 1, 2, 3, 4$, $x \in [-4, 4]$, dazu Asymptoten, Kommandos `dsolve`, `plot`, `pointplot`

Lösungstrajektorien und Zusatzzweige mittels $fa(x, n, x_0, y_0, y'_0)$ für $n = 1, 2$.

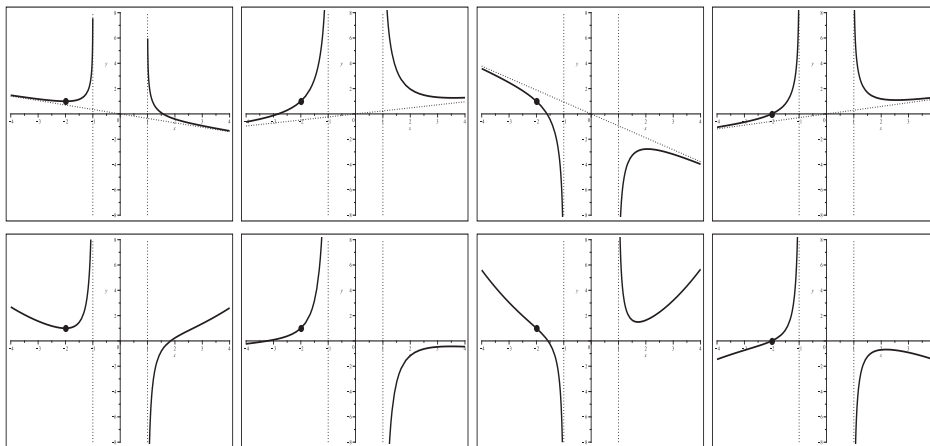


Abb. 2.321 Dateien *spez_dgl_03c211..4.ps*, *spez_dgl_c221..4.ps*, AWP zur gDGl (3), $fa(x, n, x_0, y_0, y'_0)$ für $n = 1, 2$, v.l.n.r. Grafiken mit $f_{1,k}(x)$, $k = 1, 2, 3, 4$, $x \in [-4, 4]$, dazu Asymptoten, bzw. mit $f_{2,k}(x)$, $k = 1, 2, 3, 4$, $x \in [-4, 4]$, Kommandos `dsolve`, `plot`, `pointplot`

3. Fall $|x_0| < 1$, speziell $x_0 = 0$

Die Funktion $fa(x, n, x_0, y_0, y'_0)$ ist für $|x| > 1$ bis auf spezielle n und AB komplexwertig mit nicht verschwindenden Imaginärteil. Aber auch für $|x| < 1$ erhält man eine komplexe Darstellung der Ergebnisse zu den Kommandos `fa(x,0,x0,y0,y0s)`; `simplify(fa(x,0,x0,y0,y0s))`. Erst weitere Umformungen zeigen die reelle Gestalt.

$$\begin{aligned}
 & -\frac{-y_0 \operatorname{LegendreQ}(1, 0) - \operatorname{LegendreQ}(0, 0) y'_0}{\operatorname{LegendreQ}(1, 0)} - \frac{y'_0 \operatorname{LegendreQ}(0, x)}{\operatorname{LegendreQ}(1, 0)} = \\
 & = y_0 + \frac{\operatorname{LegendreQ}(0, 0) y'_0}{\operatorname{LegendreQ}(1, 0)} - \frac{y'_0 \operatorname{LegendreQ}(0, x)}{\operatorname{LegendreQ}(1, 0)}, \quad \operatorname{LegendreQ}(1, 0) = -1 \\
 & = y_0 - \operatorname{LegendreQ}(0, 0) y'_0 + y'_0 \operatorname{LegendreQ}(0, x), \quad \operatorname{LegendreQ}(0, 0) = -\frac{1}{2}i\pi \\
 & = y_0 + \frac{1}{2}i\pi y'_0 + y'_0 \operatorname{LegendreQ}(0, x), \quad \operatorname{LegendreQ}(0, x) = \frac{1}{2} [\ln(x+1) - \ln(x-1)] \\
 & = y_0 + \frac{1}{2}i\pi y'_0 + \frac{1}{2}y'_0 \ln(x+1) - \frac{1}{2}y'_0 \ln(x-1), \\
 & \qquad \qquad \qquad \ln(x-1) = \ln((-1)(1-x)) = i\pi + \ln(1-x) \\
 & = y_0 + \frac{1}{2}y'_0 [\ln(1+x) - \ln(1-x)] \\
 & = y_0 + y'_0 \ln\left(\sqrt{\frac{1+x}{1-x}}\right) \\
 & = y_0 - y'_0 \ln\left(\sqrt{\frac{1-x}{1+x}}\right).
 \end{aligned}$$

$$\begin{aligned}
n = 0, \quad f_{01}(x) &= 1, \quad x \in \mathbb{R}, \\
f_{02}(x) &= 1 + 2g_3(x), \quad |x| < 1, \\
f_{03}(x) &= 1 - 2g_3(x), \quad |x| < 1, \\
f_{04}(x) &= g_3(x), \quad |x| < 1, \\
g_3(x) &= \frac{1}{2} [\ln(1+x) - \ln(1-x)] = \ln \left(\sqrt{\frac{1+x}{1-x}} \right).
\end{aligned}$$

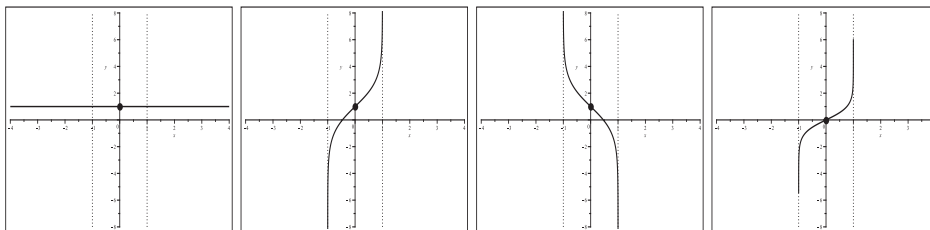


Abb. 2.322 Dateien *spez_dgl_03c301..4.ps*, $fa(x, n, x_0, y_0, y'_0)$ für $n = 0$, v.l.n.r. Grafiken mit $f_{0,k}(x)$, $k = 1, 2, 3, 4$, $x \in [-4, 4]$, Kommandos `dsolve`, `plot`, `pointplot`

Nun zu den Lösungstrajektorien mittels $fa(x, n, x_0, y_0, y'_0)$ für $n = 1, 2$.

Für $n = 1$ gibt es die besondere Situation mit dem Anfangswert $y_0 = 0$, denn die allgemeine Formel

$$\begin{aligned}
fa(x, n, x_0, y_0, y'_0) &= \\
&= - \frac{(-y'_0 \text{LegendreQ}(1, 0) - 2y_0 \text{LegendreQ}(2, 0))x}{\text{LegendreQ}(1, 0)} + \frac{y_0 \text{LegendreQ}(1, x)}{\text{LegendreQ}(1, 0)} \\
&= y'_0 x + y_0 \frac{2\text{LegendreQ}(2, 0)x + \text{LegendreQ}(1, x)}{\text{LegendreQ}(1, 0)}, \\
&\quad \text{LegendreQ}(1, 0) = -1, \quad \text{LegendreQ}(2, 0) = \frac{1}{4}\pi \\
&= y'_0 x - y_0 \left(\frac{1}{2}\pi x + \text{LegendreQ}(1, x) \right), \quad \text{LegendreQ}(1, x) = \frac{\pi}{2} [\ln(x+1) - \ln(x-1)] - 1 \\
&= y_0 + y'_0 x - \frac{1}{2}\pi y_0 x - \frac{1}{2}y_0 x [\ln(x+1) - \ln(x-1)]
\end{aligned}$$

führt dann auf die Geradengleichung $fa(x, n, 0, 0, y'_0) = y'_0 x$, die für alle reellen x definiert ist. Bei $y_0 \neq 0$ hat man jedoch den Definitionsbereich $|x| < 1$.

Für $n = 2$ gibt es eine spezielle Situation mit dem Anfangswert $y'_0 = 0$, denn die allgemeine Formel

$$\begin{aligned}
fa(x, n, x_0, y_0, y'_0) &= \\
&= \frac{2}{3} \frac{(-y'_0 \text{LegendreQ}(2, 0) - 3y_0 \text{LegendreQ}(3, 0)) \text{LegendreP}(2, x)}{\text{LegendreQ}(3, 0)} - \frac{1}{3} \frac{y'_0 \text{LegendreQ}(2, x)}{\text{LegendreQ}(3, 0)} \\
&= -\frac{2}{3} \left(3y_0 + \frac{y'_0 \text{LegendreQ}(2, 0)}{\text{LegendreQ}(3, 0)} \right) \text{LegendreP}(2, x) - \frac{1}{3} \frac{y'_0}{\text{LegendreQ}(3, 0)} \text{LegendreQ}(2, x) \\
&= -(2y_0 + \frac{1}{4}\pi y'_0) \text{LegendreP}(2, x) - \frac{1}{2} y'_0 \text{LegendreQ}(2, x)
\end{aligned}$$

führt dann auf die quadratische Parabel $fa(x, n, 0, y_0, 0) = -2y_0 P_2(x) = y_0(1 - 3x^2)$, die für alle reellen x definiert ist. Ist zusätzlich $y_0 = 0$, dann folgt die Nulllösung.

Bei $y'_0 \neq 0$ hat man ebenfalls den Definitionsbereich $|x| < 1$.

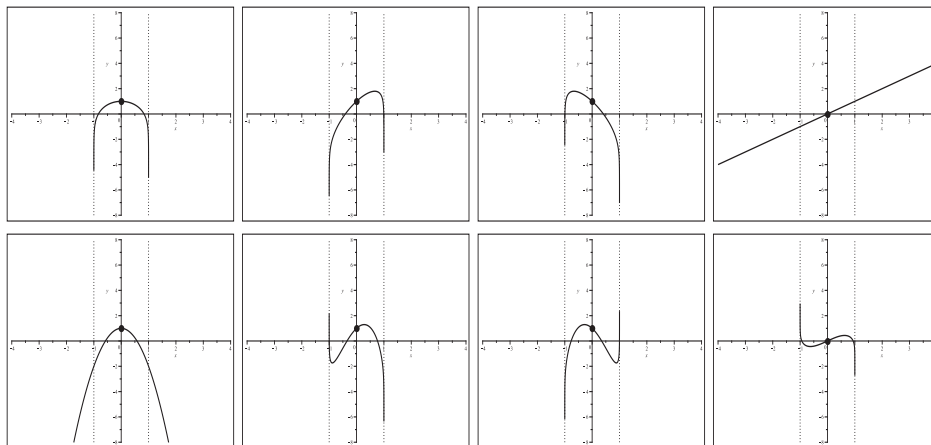


Abb. 2.323 Dateien *spez_dgl_03c311..4.ps*, *spez_dgl_c321..4.ps*,
AWP zur gDGl (3), $fa(x, n, x_0, y_0, y'_0)$ für $n = 1, 2$,
v.l.n.r. Grafiken mit $f_{1,k}(x)$ bzw. $f_{2,k}(x)$, $k = 1, 2, 3, 4$, $x \in [-4, 4]$,
Kommandos `dsolve`, `plot`, `pointplot`

Bei der Betrachtung weiterer Fälle für n “wiederholen“ sich jeweils die Situationen. Für ungerades n entsteht die besondere Situation mit dem Anfangswert $y_0 = 0$, für gerades n mit dem Anfangswert $y'_0 = 0$.

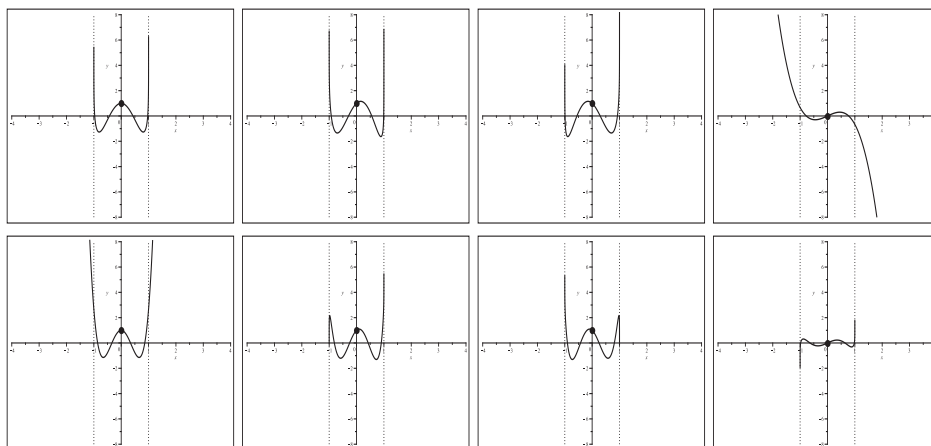


Abb. 2.324 Dateien *spez_dgl_03c331..4.ps*, *spez_dgl_c341..4.ps*,
AWP zur gDGl (3), $fa(x, n, x_0, y_0, y'_0)$ für $n = 3, 4$,
v.l.n.r. Grafiken mit $f_{3,k}(x)$ bzw. $f_{4,k}(x)$, $k = 1, 2, 3, 4$, $x \in [-4, 4]$,
Kommandos `dsolve`, `plot`, `pointplot`

(g) Nun wird das AWP mit verschiedenen AB gelöst.

Zunächst wird die AB $y(0) = y_0$, $y'(0) = y'_0$ verwendet.

```
> n:='n':
  dsol3:=dsolve([dgl3,y(0)=y0,D(y)(0)=y0s],y(x)): # x0=0 -> Problem
  fb:=unapply(rhs(dsol3),x,n,y0,y0s);
  fb(x,n,y0,y0s):
```

Hier tritt zusätzlich die Γ -Funktion in der umfangreichen Lösungsformel auf, die ebenfalls Singularitäten mit sich bringt. So wird bei den Aufrufen

```
fb(0,0,1,1); fb(0,0,0,1); fb(0,0,1,0); fb(0,0,0,0);
fb(0.5,0,1,1); fb(1,0,0,1); fb(1.5,0,1,0); fb(2,0,0,0);
fb(-0.5,0,1,1); fb(-1,0,0,1); fb(-1.5,0,1,0); fb(-2,0,0,0);
```

die Fehlermeldung `Error, (in GAMMA) numeric exception: division by zero` angezeigt, genauso beim Versuch der grafischen Ausgabe.

Die Lösung des AWP mit der AB $y(x_0) = 0$, $y'(x_0) = 0$ gemäß

```
> n:='n': x0:='x0':
  dsol3:=dsolve([dgl3,y(x0)=0,D(y)(x0)=0],y(x)): # x0 beliebig, auch +-1
  fc:=unapply(rhs(dsol3),x,n,x0);
  fc(x,n,x0), fc(1,n,x0), fc(-1,n,x0); # alle 0
```

ist immer die überall definierte Nulllösung.

Die Lösung des AWP mit der AB $y(x_0) = y_0$, $y'(x_0) = 0$ gemäß

```
> n:='n': x0:='x0': y0:='y0':
  dsol3:=dsolve([dgl3,y(x0)=y0,D(y)(x0)=0],y(x)):
  fd:=unapply(rhs(dsol3),x,n,x0,y0);
  fd := (x, n, x0, y0) ->
  
$$\frac{y_0 (\text{LegendreQ}(n+1, x_0) + x_0 \text{LegendreQ}(n, x_0)) \text{LegendreP}(n, x)}{\text{LegendreQ}(n, x_0) \text{LegendreP}(n+1, x_0) - \text{LegendreQ}(n+1, x_0) \text{LegendreP}(n, x_0)}$$

  
$$- \frac{y_0 (-\text{LegendreP}(n+1, x_0) + x_0 \text{LegendreP}(n, x_0)) \text{LegendreQ}(n, x)}{\text{LegendreQ}(n, x_0) \text{LegendreP}(n+1, x_0) - \text{LegendreQ}(n+1, x_0) \text{LegendreP}(n, x_0)}$$

```

ist entweder für $|x| < 1$ oder $|x| > 1$ definiert und an den Stellen ± 1 nicht auswertbar.

Keine Lösung hat das AWP mit der AB $y(1) = 0$, $y'(1) = 0$ gemäß

```
> n:='n':
  dsol3:=dsolve([dgl3,y(1)=0,D(y)(1)=0],y(x));
  fe:=unapply(rhs(dsol3),x,n);
```

Die konstante Lösung $y(x) = y_0$ hat das AWP mit $n=0$ und der AB $y(1) = y_0$, $y'(1) = 0$ – die Stelle kann auch x_0 sein – gemäß

```
> n:=0:
  dsol3:=dsolve([dgl3,y(1)=y0,D(y)(1)=0],y(x));
  ff:=unapply(rhs(dsol3),x);
```

Die Nulllösung $y(x) = 0$ hat das AWP mit $n \geq 1$ und der AB $y(1) = 0$, $y'(1) = 0$ gemäß

```
> n:=1: # 2,3,4,...
  dsol3:=dsolve([dgl3,y(1)=0,D(y)(1)=0],y(x));
  fg:=unapply(rhs(dsol3),x);
```

Bei der Lösung von AWP wird empfohlen, erst n auszuwählen, dann die gDGL zu lösen.

Falls $|x_0| < 1$ ist, hat man zwar komplexe Rechnungen, aber reelle Werte für $|x| < 1$.

Für $|x| > 1$ folgen aber komplexe Rechnungen und komplexe Werte ohne Grafik.

Falls $|x_0| > 1$ ist, erhält man komplexe Rechnungen und reelle Werte für $|x| > 1$.

(h) Numerik

Das numerische Verfahren berechnet zum AWP die entsprechende reelle Näherung der Lösungstrajektorie. Zusätzliche Zweige werden natürlich nicht verfolgt.

Bzgl. der zulässigen AB beachte man Folgendes:

```
|x0|<1  Naehering zu P(n,x) und Definition Q(n,x) mit 1/2*ln((1+x)/(1-x))
                                             bzw. artanh(x)

|x0|>1  Naehering zu P(n,x) und LegendreQ(n,x)
x0=+-1  verlangt y0=c, z0=0, speziell y0=z0=0
```

Die gDGL 2. Ordnung wird als SysgDGL gelöst. In der exakten Formel treten die hypergeometrischen Funktionen auf. Bei der numerischen Lösung werden AWP mit $n = 0$ und ausgewählten AB sowie das Verfahren rkf45 betrachtet.

```
> sys3:=diff(y(x),x)-z(x)=0,
      (1-x^2)*diff(z(x),x)-2*x*z(x)+n*(n+1)*y(x)=0:
      fcns:=y(x),z(x):
      dsys:=dsolve([sys3],[fcns]):
> n:=0;
      tf2:=font=[HELVETICA,BOLD,14]:
      inits31:=[y(2)=1,z(2)=1/2],[y(2)=0,z(2)=3/2];
      #inits31:=[y(2)=1,z(2)=0],[y(2)=2,z(2)=-1];
      #inits31:=[y(1)=1,z(1)=0],[y(1)=2,z(1)=0];
      #inits31:=[y(0)=2,z(0)=3],[y(0)=-1,z(0)=-1];
      #inits31:=[y(-1)=2,z(-1)=0],[y(-1)=-1,z(-1)=0];
      #inits31:=[y(-2)=2,z(-2)=1],[y(-2)=-1,z(-2)=-2];
> # Fehler, nur konstante Loesung y(x)=c machbar
      #inits31:=[y(1)=1,z(1)=1],[y(-1)=1,z(-1)=1];

> # ohne Richtungsfeld, Verfahren rkf45
      opts1:=symbol=solidcircle,symbolsize=20,color=black:
      p11:=phaseportrait([sys3],[fcns],x=-4..4,[inits31],numpoints=1000,
      linecolor=black,linestyle=solid,scene=[x,y(x)]):
      p12:=textplot([[2.5,1.5,' y(x)']]):
      p13:=pointplot([[2,1],[2,1/2]],opts1):
      p21:=phaseportrait([sys3],[fcns],x=-4..4,[inits31],
      linecolor=black,linestyle=dash,scene=[x,z(x)]):
      p22:=textplot([[2.5,0.6,' z(x)']]):
      p23:=pointplot([[2,0],[2,3/2]],opts1):
      p3:=textplot([[3,3.7,'solid: y(x)'],[3,3.2,'dash: y'(x)']],tf2):
      p4:=plot({[[-1,-4],[-1,4]],[[1,-4],[1,4]]},color=black,linestyle=dot):
> display(p11,p13,p21,p23,p3,p4,view=[-4..4,-4..4],labels=[' x', ' ']);
```

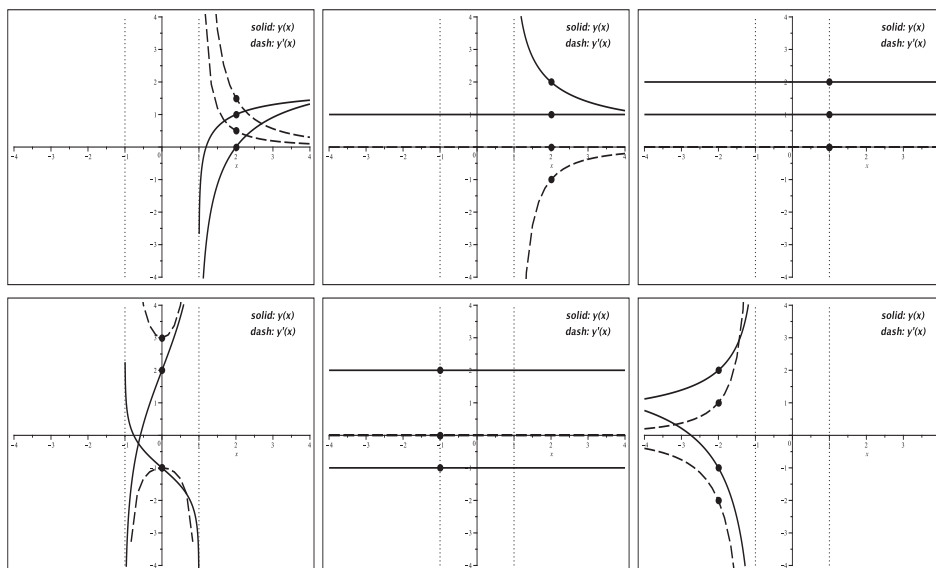


Abb. 2.325 Dateien *spez_dgl_03d01.6.ps*, AWP zur gDGL (3) als System,

Näherungslösungen und ihre Ableitungen für $n = 0$

in $[-4, 4] \times [-4, 4]$, v.l.o.n.r.u. paarweise zu den AB

$$y(2) = 1, y'(2) = \frac{1}{2}, \quad y(2) = 0, y'(2) = \frac{3}{2},$$

$$y(2) = 1, y'(2) = 0, \quad y(2) = 2, y'(2) = -1,$$

$$y(1) = 1, y'(1) = 0, \quad y(1) = 2, y'(1) = 0,$$

$$y(0) = 2, y'(0) = 3, \quad y(0) = -1, y'(0) = -1,$$

$$y(-1) = 2, y'(-1) = 0, \quad y(-1) = -1, y'(-1) = 0,$$

$$y(-2) = 2, y'(-2) = 1, \quad y(-2) = -1, y'(-2) = -2,$$

Kommandos `phaseportrait,method=rkf45, pointplot, textplot`

(4) Bernoulli-DGL

$$y' = p y + q y^\alpha$$

mit den stetigen Funktionen $p = p(x)$, $q = q(x)$ und dem reellen Parameter $\alpha \geq 1$, wobei $\alpha = 1$ die einfache gDGL $y' = (p + q) y$ ergibt. Zum AWP mit der AB $y(x_0) = 0$ entsteht die Nulllösung.

Die Lösung der gDGL bei $\alpha > 1$ erfolgt mittels der Hilfsfunktion $z = y^{1-\alpha}$.

Mit $z' = -(\alpha - 1) y^\alpha y'$ kann man folgende Umformungen und Berechnungen machen.

$$y' - p y - q y^\alpha = 0,$$

$$\frac{y'}{y^\alpha} - p \frac{y}{y^\alpha} - q = 0,$$

$$\frac{-(\alpha - 1) y'}{y^\alpha} + (\alpha - 1) p \frac{1}{y^{\alpha-1}} + (\alpha - 1) q = 0,$$

$$z' + (\alpha - 1) p z + (\alpha - 1) q = 0, \quad \tilde{p} = (\alpha - 1) p, \quad \tilde{q} = (\alpha - 1) q,$$

$$z' + \tilde{p} z + \tilde{q} = 0,$$

$$z(x) = c e^{-\int \tilde{p} dx} - \int \tilde{q} dx.$$

Mit der Lösung

$$y(x) = \frac{1}{\alpha^{-1} \sqrt[\alpha]{c e^{-\int \tilde{p} dx} - \int \tilde{q} dx}}$$

entscheidet der Nennerterm bzw. Radikand mit Positivität und Nullstellen über Definitionsbereich und Polstellen. Jedes AWP wird i. Allg. eine Lösungstrajektorie haben. Aber es können auch zusätzliche Funktionszweige entstehen.

Betrachten wir zwei konkrete lösbare Parameterfälle.

$$(4.1) \quad p(x) = -\frac{1}{1+x}, \quad q(x) = -(1+x), \quad \alpha = 4$$

Es gibt drei Lösungsformeln (reell, einfach bzw. kompliziert komplex)

$$y_1(x) = g(x), \quad y_{2,3}(x) = -\frac{1}{2}(1 \pm \sqrt{3}i) g(x), \quad \text{wobei} \quad g(x) = \frac{1}{\sqrt[3]{(x+1)^2 (c(x+1)-3)}}.$$

Die AB $y(x_0) = y_0$, $x_0 \neq -1$, führt in den beiden ersten Fällen auf den Parameterwert

$$c(x_0, y_0) = \frac{3}{x_0 + 1} + \frac{1}{(y_0(x_0 + 1))^3}.$$

Singuläre Stelle ist neben der doppelten Polstelle $x = -1$ der Wert $3/c(x_0, y_0) - 1$.

Die Lösungen von AWP werden nahe der -1 und evtl. einer weiteren Stelle extrem steile Verläufe haben. Für einige AWP werden der Definitionsbereich der Lösungstrajektorie sowie die Anzeige von Zusatzzweigen notiert.

AB	3. NS	Lösungstrajektorie im Intervall	Anzeige Zusatzweig(e)
$y(0) = 1$	$-\frac{1}{4}$	$(-\frac{1}{4}, \infty)$	kein
$y(-\frac{1}{2}) = \frac{1}{2}$	$-\frac{67}{70}$	$(-\frac{67}{70}, \infty)$	kein
$y(-\frac{3}{2}) = -\frac{1}{2}$	$-\frac{55}{58}$	$(-\infty, -1)$	1 in $(-1, -\frac{55}{58})$
$y(-\frac{3}{2}) = 1$	$-\frac{17}{14}$	$(-\infty, -\frac{17}{14})$	2 in $(-\frac{17}{14}, -1)$ und $(-1, \infty)$
$y(-2) = -1$	$-\frac{5}{2}$	$(-\frac{5}{2}, -1)$	2 in $(-\infty, -\frac{5}{2})$ und $(-1, \infty)$
$y(-\frac{1}{2}) = -\frac{1}{2}$	$-\frac{61}{58}$	$(-1, \infty)$	2 in $(-\infty, -\frac{61}{58})$ und $(-\frac{61}{58}, -1)$
$y(0) = -1$	$\frac{1}{2}$	$(-1, \frac{1}{2})$	1 in $(-\infty, -1)$
$y(0) = 0$		Lösung $y(x) = 0$	
$y(-1) = 0$		Lösung $y(x) = 0$	

Wie in Maple das AWP gelöst und die Lösungsformel darstellt wird sowie diese zusätzlichen Anzeigen von Funktionszweigen zustande kommen, soll bei drei ausgewählten AB gezeigt werden.

(a) Reeller Fall

$$y(0) = 1, \quad c = 4, \quad y(x) = \frac{1}{\sqrt[3]{(x+1)^2(4x+1)}}.$$

Im Intervall $(-\frac{1}{4}, \infty)$ ist die Funktion $y(x)$ reellwertig und darstellbar.

Für $x < -\frac{1}{4}$ entstehen komplexe Funktionswerte und die Kurve kann nicht gezeichnet werden.

(b) Einfacher komplexer Fall

$$y(-\frac{3}{2}) = -\frac{1}{2}, \quad c = 58, \quad y(x) = -\frac{1}{2}(1 + \sqrt{3}i)g(x), \quad \text{wobei } g(x) = \frac{1}{\sqrt[3]{(x+1)^2(58x+55)}}.$$

In den Intervallen $(-\infty, -1)$ und $(-1, -\frac{55}{58})$ hat $g(x)$ einen negativen Radikanden und die Funktion $y(x)$ zwar eine komplexe Darstellung, ist aber reellwertig. Damit kann sowohl die Lösungstrajektorie in $(-\infty, -1)$ als auch der Funktionszweig in $(-1, -\frac{55}{58})$ geplottet werden. Im Intervall $(-\frac{55}{58}, \infty)$ ist der Radikand positiv und es gibt keinen Plot.

(c) Komplizierter komplexer Fall

$$y(-\frac{3}{2}) = 1, \quad c = 14, \quad y(x) = -\frac{1}{2}(1 - \sqrt{3}i)\frac{\sqrt[3]{-r(x)^2}}{r(x)}, \quad \text{wobei } r(x) = (x+1)^2(14x+17).$$

Unabhängig vom Vorzeichen von $r(x)$ ist der Radikand unter der dritten Wurzel ≤ 0 .

Für die dritte Wurzel einer negativen Zahl $-z$ nimmt man normalerweise $-\sqrt[3]{z}$.

Aber in Maple erhält man für $(-1)^{1/3}$ nicht den Wert -1 , sondern die komplexe Größe $\frac{1}{2}(1 + \sqrt{3}i)$, analog für $(-z)^{1/3}$ den Wert $\frac{1}{2}(1 + \sqrt{3}i)\sqrt[3]{z}$.

In allen drei Intervallen $(-\infty, -\frac{17}{14})$, $(-\frac{17}{14}, -1)$ und $(-1, \infty)$ hat die Funktion $y(x)$ zwar eine komplexe Darstellung, ist aber reellwertig. Damit können sowohl die Lösungstrajektorie im Intervall $(-\infty, -\frac{17}{14})$ als auch die Funktionszweige in den anderen Intervallen geplottet werden.

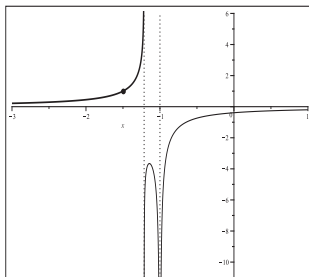


Abb. 2.326

Datei *spez_dgl_04b0.ps*, gDGl (4.1),

AWP für $p(x) = -\frac{1}{1+x}$, $q(x) = -(1+x)$, $\alpha = 4$,

Lösung zur AB $y(-\frac{3}{2}) = 1$ in $[-3, -\frac{17}{14}]$,

zusätzliche Funktionszweige (dünn gezeichnet)

in $(-\frac{17}{14}, -1)$ und $(-1, \infty)$,

Kommandos `dsolve`, `plot`, `pointplot`

Würde man $\sqrt[3]{-r(x)^2}$ ersetzen durch $-\sqrt[3]{r(x)^2}$, so folgt die Funktion

$$\tilde{y}(x) = \frac{1}{2}(1 - \sqrt{3}i) \frac{\sqrt[3]{r(x)^2}}{r(x)} = \frac{1}{2}(1 - \sqrt{3}i) \frac{1}{\sqrt[3]{r(x)}}.$$

Diese Funktion bleibt überall komplexwertig und kann nirgends gezeichnet werden.

Würde man $\sqrt[3]{-r(x)^2}$ ersetzen durch $\frac{1}{2}(1 + \sqrt{3}i) \sqrt[3]{r(x)^2}$, so ergibt sich die ursprüngliche Lösung in der Gestalt

$$y(x) = -\frac{\sqrt[3]{r(x)^2}}{r(x)}.$$

Ihre weitere Vereinfachung auf

$$\hat{y}(x) = -\frac{1}{\sqrt[3]{r(x)}}$$

ist "teilweise" fehlerhaft. Für $r(x) < 0$ ist $y(x)$ reell und positiv, während $\hat{y}(x)$ komplex wird. Nur bei $r(x) > 0$ stimmen $y(x)$ und $\hat{y}(x)$ überein und sind reellwertig.

So ist an der Anfangsstelle $y(-\frac{3}{2}) = 1$, $\hat{y}(-\frac{3}{2}) = -\frac{1}{2}(1 - \sqrt{3}i)$ und $\tilde{y}(-\frac{3}{2}) = \frac{1}{2}(1 + \sqrt{3}i)$.

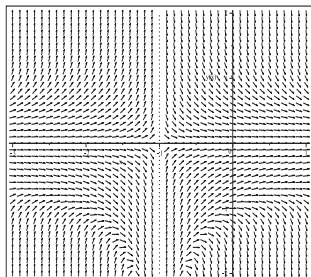


Abb. 2.327

Datei *spez_dgl_04a1.ps*, gDGl (4.1),

$p(x) = -\frac{1}{1+x}$, $q(x) = -(1+x)$, $\alpha = 4$,

Richtungsfeld in $[-3, 1] \times [-2, 2]$,

Asymptote bei $x = -1$,

Kommandos `dfieldplot`, `plot`

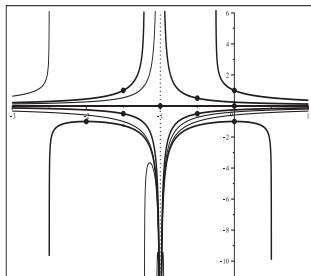


Abb. 2.328

Datei *spez_dgl_04b1.ps*, gDGl (4.1),

AWP für $p(x) = -\frac{1}{1+x}$, $q(x) = -(1+x)$, $\alpha = 4$,

Lösungen für $x \in [-3, 1]$ zu den AB

$y(-1) = 0$, $y(0) = 0$, $y(0) = 1$, $y(0) = -1$, $y(-\frac{1}{2}) = \frac{1}{2}$,

$y(-\frac{1}{2}) = -\frac{1}{2}$, $y(-\frac{3}{2}) = 1$, $y(-\frac{3}{2}) = -\frac{1}{2}$, $y(-2) = -1$,

zusätzliche Funktionszweige (dünn gezeichnet),

Kommandos `dsolve`, `plot`, `pointplot`

Die numerische Lösung der AWP wird nur die Lösungszweige in den Intervallen liefern, wo die AB liegen. Die Verwendung des Kommandos `phaseportrait` zeigt das Richtungsfeld sowie die Näherungslösungen. Mit den Verfahren `rkf45` bzw. `rosenbrock` lassen sie sich hinreichend genau bestimmen, wobei diese mit ihren steilen Verläufen mehr oder weniger nahe an die singulären Stellen kommen. In den beiden nachfolgenden Abbildungen ist nur die Asymptote bei $x = -1$ eingetragen.

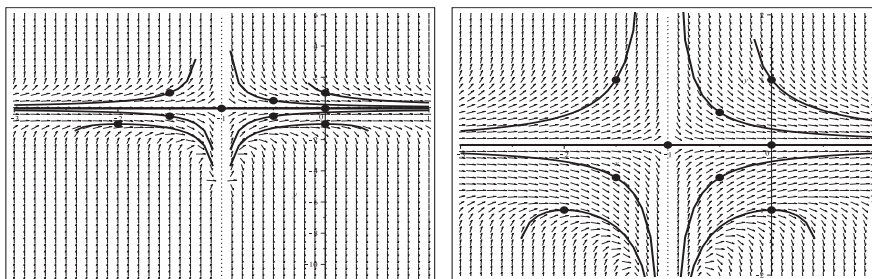


Abb. 2.329 Dateien *spez_dgl_04b2..3.ps*, gDGl (4.1),
 AWP für $p(x) = -\frac{1}{1+x}$, $q(x) = -(1+x)$, $\alpha = 4$, $x \in [-3, 1]$,
 AB $y(-1) = 0$, $y(0) = 0$, $y(0) = 1$, $y(0) = -1$, $y(-\frac{1}{2}) = \frac{1}{2}$,
 $y(-\frac{1}{2}) = -\frac{1}{2}$, $y(-\frac{3}{2}) = 1$, $y(-\frac{3}{2}) = -\frac{1}{2}$, $y(-2) = -1$,
 Näherungslösungen in $[-3, 1] \times [-11, 6]$, bzw. $[-3, 1] \times [-2, 2]$,
 dazu Richtungsfeld und Asymptote bei $x = -1$,
 Kommandos `phaseportrait,method=rkf45, plot,pointplot`

$$(4.2) \quad p(x) = x, \quad q(x) = -e^{-x^2/2}, \quad \alpha = 3$$

$$y(x) = \frac{e^{x^2/2}}{\sqrt{2x+c}}$$

Die Lösung ist einfacher und überschaubarer als im Fall (4.1), die Polstellen gut zu erkennen. Die AB $y(x_0) = y_0$ führt auf den Parameterwert $c(x_0, y_0) = e^{x_0^2} y_0^{-2} - 2x_0$. Das Richtungsfeld zur gDGl ist überall definiert und in der Abbildung mit den Näherungslösungen dargestellt.

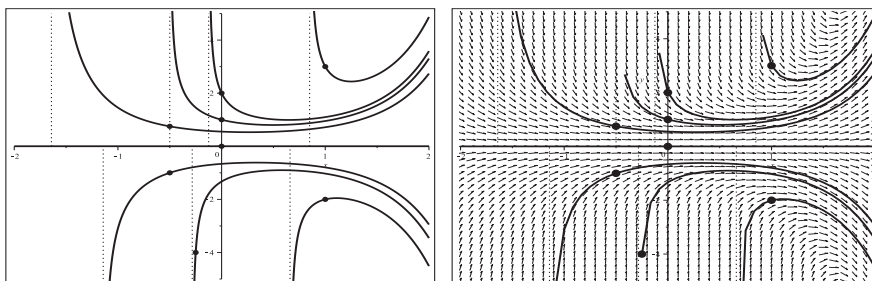


Abb. 2.330 Dateien *spez_dgl_04c1..2.ps*, gDGl (4.2),
 AWP für $p(x) = x$, $q(x) = -e^{-x^2/2}$, $\alpha = 3$, $x \in [-2, 2]$,
 links: Lösungen zu den AB $y(0) = 0$, $y(0) = 1$, $y(0) = 2$,
 $y(-\frac{1}{2}) = \frac{3}{4}$, $y(-\frac{1}{4}) = -4$, $y(-\frac{1}{2}) = 1-$, $y(1) = 3$, $y(1) = -2$,
 Kommandos `dsolve, plot,pointplot`,
 rechts: Näherungslösungen zu diesen AB in $[-2, 2] \times [-5, 5]$,
 dazu Richtungsfeld und Asymptoten,
 Kommandos `phaseportrait,method=rosenbrock, plot,pointplot`

(5) Abkühlungs-DGI

$$T' = -k(T - T_u)$$

Das Newtonsche Abkühlungsgesetz sagt Folgendes aus.

Ein Körper mit der Temperatur T_0 befindet sich in einer Umgebung mit der Temperatur $T_u < T_0$. Die konstante Abkühlungsrate ist $k > 0$ (sie korrespondiert mit der Wärmekapazität bzw. Wärmeleitfähigkeit des Materials). Dann beschreibt das Gesetz die Abkühlung des Körpers als Funktion der Zeit $T = T(t)$.

Die allgemeine Lösung der gDGI (5) ist

$$T(t) = T_u + c e^{-kt},$$

die des AWP mit der AB $T(0) = T_0$

$$T(t) = T_u + (T_0 - T_u) e^{-kt}.$$

Moderate Abkühlungsraten k machen bzgl. der Anzeige des Richtungsfeldes sowie bei der numerischen Lösung von AWP keine Probleme.

Von Interesse ist die Situation, wie groß die Abkühlungsrate sein muss, damit ausgehend von einer Anfangstemperatur die Materialtemperatur nach einer bestimmte Zeit t_e einen vorgegebenen Wert erreicht. Aus der Gleichung $T(t_e) = T_e$ folgt unmittelbar

$$k = \frac{1}{t_e} \ln \left(\frac{T_0 - T_u}{T_e - T_u} \right).$$

Unter den Bedingungen $T_u = 20$, $T_0 = 80$, $t_e = 20$, $T(t_e) = 60$ erhält man die Abkühlungsrate $k = \ln(3/2)/20 = 0.020273$, die auch bei anderen Anfangstemperaturen verwendet werden soll. Das Richtungsfeld zur gDGI ist überall definiert und in der Abbildung mit den Näherungslösungen dargestellt.

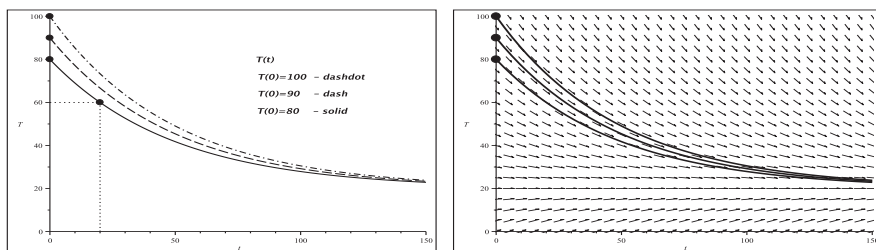


Abb. 2.331 Dateien *spez_dgl_05b..c.ps*, gDGI (5),

AWP mit $T_u = 20$, $k = \ln(3/2)/20$ in $[0, 150] \times [0, 100]$,

links: Lösungen zu den AB $T(0) = 100, 90, 80$,

Kommandos `dsolve`, `plot`, `pointplot`,

rechts: Näherungslösungen zu diesen AB, dazu Richtungsfeld,

Kommandos `phaseportrait`, `plot`, `pointplot`

(6.1) Lineare gDGL 2. Ordnung mit variablen Koeffizienten mit Ähnlichkeit zur Hermiteischen DGL

$$y'' + x y' + 6y = 0$$

Die allgemeine Lösung ist

$$y(x) = c_1 x e^{-x^2/2} \text{hypergeom}([-2], [3/2], x^2/2) + c_2 e^{-x^2/2} \text{hypergeom}([-5/2], [1/2], x^2/2).$$

Zur AB $y(0) = y_0$, $y'(0) = y'_0$ erhält man

$$y(x) = y'_0 x e^{-x^2/2} \text{hypergeom}([-2], [3/2], x^2/2) + y_0 e^{-x^2/2} \text{hypergeom}([-5/2], [1/2], x^2/2),$$

zur AB $y(x_0) = y_0$, $y'(x_0) = y'_0$ ist die Lösungsformel recht lang.

Mit der AB $y(x_0) = 0$, $y'(x_0) = 0$ entsteht die Nulllösung.

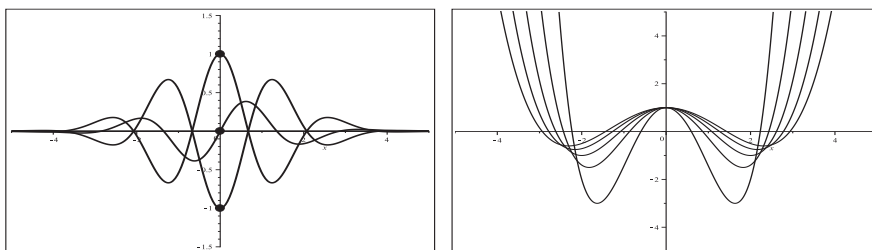


Abb. 2.332 Dateien *spez_dgl_061c1..2.ps*, gDGL (6.1),
links: Lösungen von AWP in $[-5, 5] \times [1.5, 1.5]$ mit den AB,
 $(y(0), y'(0)) = (0, 0), (1, 0), (-1, 0), (0, 1)$,
Kommandos `dsolve`, `plot`, `pointplot`,
rechts: hypergeometrische Funktionen
 $\text{hypergeom}([-2], [k/3], x^2/2)$, $k = 1(1)5$, in $[-5, 5] \times [-5, -5]$,
Kommando `plot`

Sehr gute Näherungslösungen können mittels `dsolve/numeric, phaseportrait, DEplot` erzeugt werden.

Notiert und löst man die gDGL (6.1) als System 1. Ordnung, beinhalten die Formeln die reelle und imaginäre Fehlerfunktionen

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad \operatorname{erfi}(x) = -i \operatorname{erf}(ix) = \frac{2}{\sqrt{\pi}} \int_0^x e^{t^2} dt.$$

Mit der AB $(y(0), y'(0)) = (1, 0)$ folgt

$$\begin{aligned} y(x) &= \frac{1}{8} (8 - 9x^2 + x^4) + \frac{\sqrt{2\pi}}{16} (-15x + 10x^3 - x^5) e^{-x^2/2} \operatorname{erfi}\left(\frac{\sqrt{2}}{2}x\right), \\ y'(x) &= \frac{1}{8} (-33x + 14x^3 - x^5) + \frac{\sqrt{2\pi}}{16} (-15 + 45x^2 - 15x^4 + x^6) e^{-x^2/2} \operatorname{erfi}\left(\frac{\sqrt{2}}{2}x\right). \end{aligned}$$

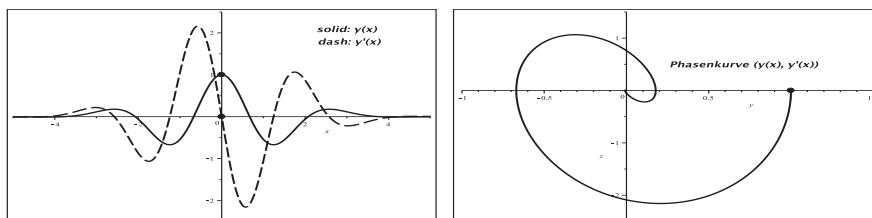


Abb. 2.333 Dateien *spez_dgl_061b1..2.ps*, gDGL (6.1),
links: Näherungslösung für AWP zur AB $(y(0), y'(0)) = (1, 0)$
in $[-5, 5] \times [2.5, 2.5]$, dazu Näherung der Ableitung,
rechts: Näherung der Phasenkurve $(y(x), y'(x))$ zu dieser AB,
Kommandos `phaseportrait,method=rkf45, pointplot, textplot`

(6.2) Hermitesche DGL

$$y'' - 2xy' + 2ny = 0, \quad n \text{ natürlicher Parameter}$$

In der allgemeinen Lösungsformel

$$y(x) = c_1 x \text{KummerM}((1-n)/2, 3/2, x^2) + c_2 x \text{KummerU}((1-n)/2, 3/2, x^2)$$

treten die Kummer-Funktionen $\text{KummerM}(\mu, \nu, x)$ und $\text{KummerU}(\mu, \nu, x)$ auf, die selber Lösungen der gDGL $xy'' + (\nu - x)y' - \mu y = 0$ sind.

Aber für spezielle Parameterwerte n und entsprechende AB $y(0) = y_0, y'(0) = y'_0$ erhält man die orthogonale Hermite-Polynome $H_n(x) = \text{orthopoly}[H](n, x) = \text{HermiteH}(n, x)$.

n	AB	$H_n(x)$
0	$y_0 = 1, y'_0 = 0$	1
1	$y_0 = 0, y'_0 = 2$	$2x$
2	$y_0 = -2, y'_0 = 0$	$4x^2 - 2$
3	$y_0 = 0, y'_0 = -12$	$8x^3 - 12x$
4	$y_0 = 12, y'_0 = 0$	$16x^4 - 48x^2 + 12$

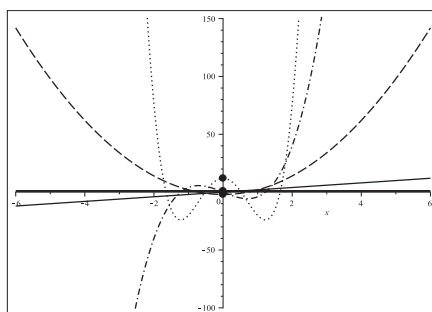


Abb. 2.334
Datei *spez_dgl_062c1.ps*,
Hermite-Polynome $H_n(x), n = 0(1)4$,
in $[-6, 6] \times [-100, 150]$,
Kommandos `plot, pointplot`

Zum AWP mit $n = 0$ und der AB $y(x_0) = y_0, y'(x_0) = 0$ entsteht die konstante Lösung $y(x) = y_0$.

Die numerische Lösung sollte für die überall definierten Hermite-Polynome $H_n(x)$ keine größeren Probleme darstellen.

Mit dem Kommando `dsolve(subs(n=k,dgl62),y(0)=y0,D(y)(0)=ys0,y(x),numeric)` finden wir Näherungen für $k = 0, 1$, für $k = 2, 3, 4$ nehmen wir die Option `stiff=true` hinzu, wo die Methode `rosenbrock` verwendet wird. Entgegen allen Erwartungen weicht das Verfahren bei $n = 2$ für betragsgrößere Argumente x von der Lösung $H_2(x)$ ab.

Zwei zusätzliche Rechnungen mit den Methoden `gear` und `rkf45` bei kleinerem Parameter `stepsize` liefern ebenfalls keine befriedigenden Ergebnisse. Für $n = 3, 4$ tritt dieses Phänomen nicht auf, obwohl die Anstiege von $H_{3,4}(x)$ doch viel größer sind.

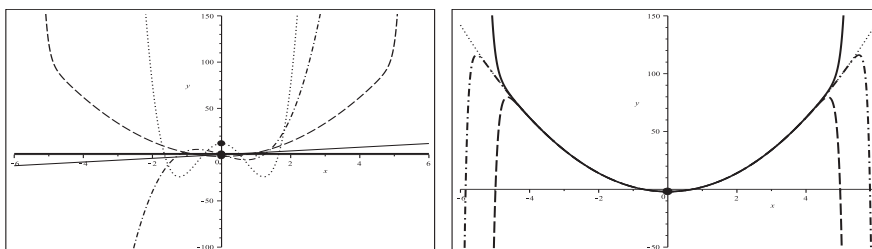


Abb. 2.335 Dateien `spez_dgl_062c2..3.ps`, gDGL (6.2),
AWP mit $n = 0, 1, 2, 3, 4$ in $[-6, 6] \times [-100(-50), 150]$,
links: Näherungslösungen zu den AB
 $(y(0), y'(0)) = (1, 0), (0, 2), (-2, 0), (0, -12), (12, 0)$,
Kommandos `dolve/numeric, odeplot, pointplot`,
rechts: $H_2(x)$ und Näherungslösungen zu $n = 2$, $(y(0), y'(0)) = (-2, 0)$,
Kommandos `phaseportrait, method=rosenbrock, gear, rkf45,`
`stepsize=0.001, pointplot`

Als SysgDGL ergänzen wir zum AWP mit $n = 2$ und $(y(0), y'(0)) = (-2, 0)$ noch die Darstellung von Näherungen zur Lösung, ihrer Ableitung und Phasenkurve.

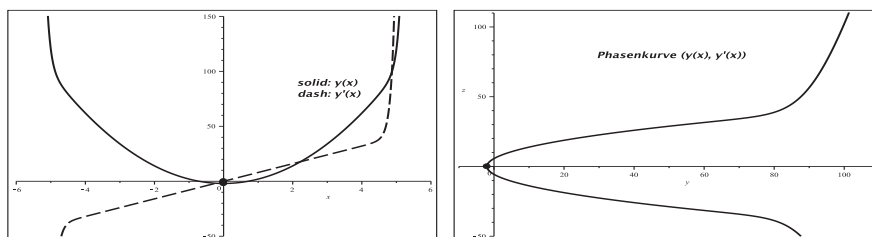


Abb. 2.336 Dateien `spez_dgl_062b1..2.ps`, gDGL (6.2),
AWP mit $n = 2$ in $[-6, 6] \times [-50, 150]$ bzw. $[-10, 110] \times [-50, 110]$,
links: Näherung von Lösung und Ableitung zu $(y(0), y'(0)) = (-2, 0)$,
rechts: Näherung der Phasenkurve zu dieser AB,
Kommandos `phaseportrait, pointplot, textplot`

(7) Clairautsche DGI

$$y = x y' + f(y')$$

Mit der Funktion $f(z) = z^2$ erhält man als allgemeine Lösung dieser gDGI

$$y(x) = -\frac{1}{4}x^2, \quad y(x) = c(x+c),$$

also u.a. einen Lösungsanteil, der keine AB braucht, sowie eine Geradenschar. Außerdem erkennt man schon die Mehrdeutigkeit bei $c = -\frac{1}{2}$ und $y(1) = -\frac{1}{4}$, $y'(1) = c$.

Mit der Berücksichtigung der AB $y(x_0) = y_0$ ergibt sich eine einzige Lösung

$$y(x) = y_0 - \frac{1}{2} \left(x_0 - \sqrt{x_0^2 + 4y_0} \right) (x - x_0).$$

Für bestimmte AB wird der Radikand negativ und damit die Lösung komplex.

Die konkrete AB $y(1) = 1$ liefert überraschenderweise fünf Lösungsdarstellungen, die sich jedoch auf die beiden Geradengleichungen

$$y_1(x) = \frac{3+\sqrt{5}}{2} - \frac{1+\sqrt{5}}{2}x, \quad y_2(x) = \frac{3-\sqrt{5}}{2} + \frac{-1+\sqrt{5}}{2}x$$

reduzieren.

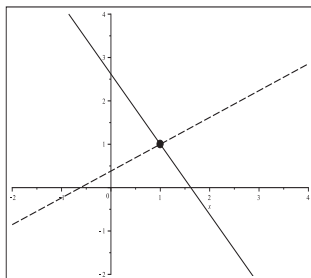


Abb. 2.337

Datei *spez_dgl_07c1.ps*,

gDGI (7) 1. Ordnung,

AWP mit der AB $y(1) = 1$,

Lösungen $y_1(x), y_2(x)$ in $[-2, 4] \times [-2, 4]$,

Kommandos `dsolve, plot, pointplot`

Für ein Richtungsfeld der gDGI braucht man die explizite Umstellung nach $y'(x)$ gemäß $y' = -\frac{1}{2}(x \pm \sqrt{x^2 + 4y})$. Seine Beschränkung wird durch die Bedingung $x^2 + 4y \geq 0$ bzw. $y \geq -\frac{1}{4}x^2$ erzeugt.

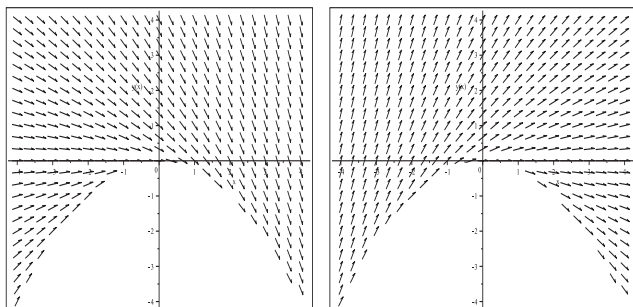


Abb. 2.338

Dateien *spez_dgl_07a1..2.ps*,

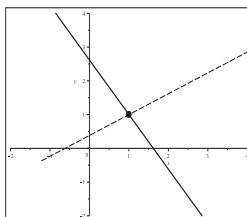
gDGI (7) mit $f(z) = z^2$,

Richtungsfelder

in $[-2, 4] \times [-2, 4]$,

Kommando `dfieldplot`

Die sehr genaue numerische Lösung zur AB $y(1) = 1$ mittels `dsolve/numeric,odeplot` kann für $y_1(x)$ nach rechts nur bis zur Stelle x gemäß $x^2 + 4y_1(x) = 0$ verfolgt werden, das ist der Wert $1 + \sqrt{5} = 3.236067$. Nach links verfolgt man $y_2(x)$ bis zur Nullstelle $1 - \sqrt{5} = -1.236067$ von $x^2 + 4y_2(x)$.

**Abb. 2.339**

Datei `spez_dgl_07b1.ps`,

gDGL (7) 1. Ordnung,

AWP mit der AB $y(1) = 1$,

Näherungslösungen zu $y_1(x)$, $y_2(x)$ in $[-2, 4] \times [-2, 4]$,

Kommandos `dsolve/numeric, odeplot, pointplot`

Die nichtlineare gDGL kann man durch Differenzieren auf eine zweite Gestalt transformieren. Aber solche Transformationen verursachen oft zusätzliche Effekte. Mit $f(z)$ erhält man die spezielle gDGL 2. Ordnung

$$y''(x + 2y') = 0,$$

sodass dann zur Lösungsmannigfaltigkeit sowohl eine Geradenschar ($y'' = 0$) als auch quadratische Parabeln ($x + 2y' = 0$) gehören. Die Besonderheit ist nun, dass die gDGL $y'' = 0$ zweiparametrische Lösungen liefert und somit das AWP AB der Form $y(x_0) = y_0$, $y'(x_0) = y'_0$ braucht, während die gDGL $x + 2y' = 0$ mit einem Anfangswert auskommt. Die allgemeine Lösung der gDGL 2. Ordnung enthält die Funktionen

$$y(x) = -\frac{1}{4}x^2 + c, \quad y(x) = c_1x + c_2,$$

also eine Parabel- und eine Geradenschar.

Betrachten wir nun verschiedene AB.

$$-y(x_0) = y_0$$

Lösungen sind die eindeutige Parabel $y_1(x) = y_0 - \frac{1}{4}(x^2 - x_0^2)$ sowie die Geradenschar $y_2(x) = y_0 + c(x - x_0)$.

Damit haben die spezielle Gerade $y_2(x) = y_0 - \frac{1}{2}(x - x_0)$ und die Parabel bei x_0 den gleichen Ableitungswert $-\frac{1}{2}$, falls $x_0 = 1$ ist. Also verläuft dann durch den Punkt $(1, y_0)$ nicht nur die unendliche Geradenschar, sondern dort auch eine Gerade und Parabel mit dem gleichen Anstieg.

$$-y(x_0) = y_0, \quad y'(x_0) = -\frac{1}{2}x_0$$

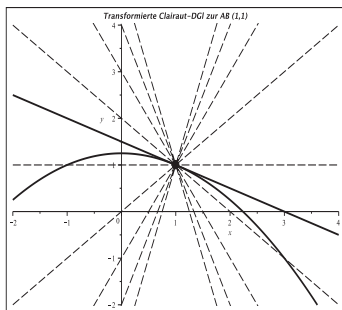
Die Lösungen sind die eindeutige Parabel $y_1(x) = y_0 - \frac{1}{4}(x^2 - x_0^2)$ sowie die Gerade $y_2(x) = y_0 - \frac{x_0}{2}(x - x_0)$.

$$-y(x_0) = y_0, \quad y'(x_0) \neq -\frac{1}{2}x_0, \quad \text{d.h. } y'(x_0) = y'_0$$

Die Lösung ist die eindeutige Gerade $y_2(x) = y_0 + y'_0(x - x_0)$, es gibt keine Parabel.

$$-y(1) = 1, \quad y'(1) = -\frac{1}{2}$$

Lösungen sind die eindeutige Parabel $y_1(x) = \frac{5-x^2}{4}$ sowie die Gerade $y_2(x) = \frac{3-x}{2}$.

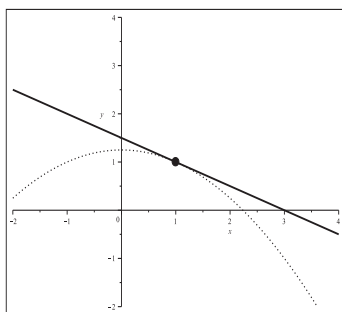
**Abb. 2.340**

Datei *spez_dgl_07c2.ps*, gDGI (7) 2. Ordnung,
AWP mit den AB $y(1) = 1$, $y'(1) = -\frac{1}{2}$ bzw. $y(1) = 1$,
Lösungen:

Parabel $y_1(x) = \frac{5-x^2}{4}$ und Gerade $y_2(x) = \frac{3-x}{2}$
bzw. Geradenschar $y_2(x) = 1 - k(x-1)$, $k = -4(1)4$,
in $[-2, 4] \times [-2, 4]$,

Kommandos `dsolve`, `plot`, `pointplot`

Trotz Mehrdeutigkeit der Lösung der gDGI (7) 2. Ordnung zur AB $y(1) = 1$, $y'(1) = -\frac{1}{2}$ erzeugen die numerisches Verfahren Näherungen, welche die Geraden approximieren.

**Abb. 2.341**

Datei *spez_dgl_07b2.ps*,

AWP mit der AB $y(1) = 1$, $y'(1) = -\frac{1}{2}$,

Näherungslösung in $[-2, 4] \times [-2, 4]$,

dazu die Parabelfunktion $y_1(x) = \frac{5-x^2}{4}$ (dot),

Kommandos `dsolve/numeric`, `odeplot`, `plot`

(8) Abroll-DGI

$$u' = v, \quad u(0) = 30,$$

$$u' = -\frac{u}{2025 + u^2} (31532 + v^2), \quad v(0) = 0$$

Das autonome nichtlineare SysgDGI beschreibt das Abrollen einer Kugel auf einer Falllinie eines parabolischen Zylinders.

Die analytische Lösung sowohl des SysgDGI als auch des AWP ist nicht möglich.

Somit bleibt nur die numerische Berechnung einer Näherungslösung. Da das SysgDGI autonom ist, kann im Phasenportrait auch das Richtungsfeld gezeigt werden.

```
> sys8:=diff(u(t),t)=v(t),
      diff(v(t),t)=-u(t)/(2025+u(t)^2)*(31532+v(t)^2):
fcons:=u(t),v(t):
inits8:=u(0)=30,v(0)=0:

> lsg8:=dsolve({sys8,inits8},{u(t),v(t)},numeric,output=listprocedure):
assign(subs(u(t)=fu,v(t)=fv,lsg8)):
> pl1:=plot([fu,fv],0..4,color=black,linestyle=[solid,dash],thickness=3):
pl2:=pointplot([[0,30],[0,0]],
               symbol=solidcircle,symbolsize=16,color=black):
```

```

p13:=textplot([[0.53,55,'solid: u(t)'],[0.5,70,'dash: v(t)']],tf1):
display(p11,p12,p13);

> p11:=phaseportrait([sys8],[fcs],t=0..2,[[inits8]],color=black,
    linecolor=black,scene=[u(t),v(t)],labels=['u','v']):
p12:=textplot([[0,60,'Phasenkurve (u(t), v(t))']],tf1):
p13:=pointplot([30,0],symbol=solidcircle,symbolsize=20,color=black):
display(p11,p12,p13,view=[-30..30,-120..120]);

```

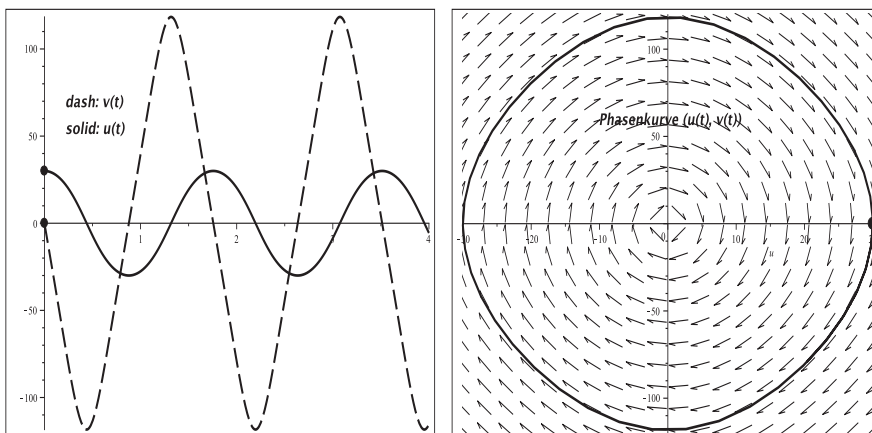


Abb. 2.342 Dateien *spez_dgl_08c1..2.ps*, SysgDGl (8),
links: Näherungslösung für AWP zur AB $(u(0), v(0)) = (30, 0)$
in $[0, 4] \times [-120, 120]$, zum Verlauf ca. 2 Perioden,
Kommandos `dsolve/numeric`, `plot`, `pointplot`, `textplot`,
rechts: Näherung der Phasenkurve $(u(t), v(t))$, $t \in [0, 2]$, zu dieser AB,
dazu Richtungsfeld,
Kommandos `phaseportrait`, `pointplot`, `textplot`

(9) Kettenlinie

Die Kettenlinie kann man als Lösung einer gDGl 2. Ordnung beschreiben.

Als AWP mit den Parametern $a > 0$ (Formkonstante) und $y_0 > 0$ notiert man z.B.

$$a^2 y'' = y - y_0, \quad y(x_0) = y_0 + a, \quad y'(x_0) = 0, \quad -\frac{w}{2} \leq x \leq \frac{w}{2}.$$

Das entsprechende Randwertproblem dazu ist

$$a^2 y'' = y - y_0, \quad y\left(-\frac{w}{2}\right) = y_0 + a \cosh\left(\frac{-w/2 - x_0}{a}\right) = y\left(\frac{w}{2}\right), \quad -\frac{w}{2} \leq x \leq \frac{w}{2}.$$

Die Lösung ist in beiden Fällen

$$y(x) = a \cosh\left(\frac{x - x_0}{a}\right) + y_0 \quad \text{mit} \quad y'(x) = \sinh\left(\frac{x - x_0}{a}\right).$$

Die Länge der Kettenlinie sowie das Längenstück zwischen ihrem Tiefpunkt (Scheitel) S und dem Kettenpunkt P bei $u \in (0, \frac{w}{2}]$, $w \leq 2$, betragen

$$l = \int_{w/2}^{w/2} \sqrt{1+y'^2} dx = 2a \sinh\left(\frac{w}{2a}\right) \quad \text{bzw.} \quad h(a, u) = \int_0^u \sqrt{1+y'^2} dx = a \sinh\left(\frac{u}{a}\right).$$

Es soll hier nicht um die numerische Lösung gehen, sondern um zwei spezielle Eigenschaften der Kettenlinie. Das wird in der Abbildung illustriert.

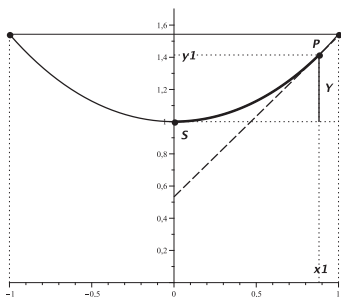


Abb. 2.343

Datei *spez_dgl_09.ps*,

Kettenlinie $y(x)$ mit $a = 1$ in $[-1, 1]$

mit Scheitelpunkt $S(0, 1)$ und

Kettenpunkt $P(x_1, y_1)$, wo $y'(x_1) = 1$

Man prüfe Folgendes nach.

1. Die Kettenlänge zwischen ihrem Tiefpunkt S und dem Kettenpunkt P , dessen Tangente 45° (zur Horizontalen oder auch Vertikalen) beträgt, entspricht als dimensionsloser Wert exakt ihrer Formkonstanten a .

Man rechnet bei $x_0 = 0$, $y_0 = 0$ mit den Kettenlinienfunktionen $y(x) = a \cosh(\frac{x}{a})$, $y'(x)$, der Länge $h(a, u)$ des Kurvenstücks SP sowie den Hyperbolicusfunktionen.

$$\begin{aligned} y'(x) &= \sinh\left(\frac{x}{a}\right) = 1, \\ \frac{x}{a} &= \operatorname{arsinh}(1) = \ln(1 + \sqrt{1^2 + 1}) = \ln(1 + \sqrt{2}), \\ x_1 &= a \operatorname{arsinh}(1) = a \ln(1 + \sqrt{2}), \\ h(a, x_1) &= a \sinh\left(\frac{x_1}{a}\right) = a \sinh\left(\frac{a \operatorname{arsinh}(1)}{a}\right) = a. \end{aligned}$$

Die Länge des Kurvenstücks SP mit dem Tangentenanstieg 1 im Punkt P korrespondiert mit der Formkonstanten a .

2. Liegt der Scheitel S bei 0, so gilt im 45° -Punkt P die Beziehung $a = Y(1 + \sqrt{2})$, wobei Y die Lothöhe bezeichnet. Der Scheitel S hat die Koordinaten $(0, a)$ (allgemein). Für die Lothöhe gilt $Y = y(x_1) - a$. Die Abszisse von P ist $x_1 = a \operatorname{arsinh}(1)$.

Weiter folgt

$$\begin{aligned} Y \cdot (1 + \sqrt{2}) &= (y(x_1) - a) (1 + \sqrt{2}) = (a \cosh(a \operatorname{arsinh}(1)/a) - a) (1 + \sqrt{2}) \\ &= (a \cosh(\operatorname{arsinh}(1)) - a) (1 + \sqrt{2}) = a (\cosh(\operatorname{arsinh}(1)) - 1) (1 + \sqrt{2}) \\ &= a (\sqrt{\sinh(\operatorname{arsinh}(1))^2 + 1} - 1) (1 + \sqrt{2}) \\ &= a (\sqrt{1^2 + 1} - 1) (1 + \sqrt{2}) = a (\sqrt{2} - 1) (1 + \sqrt{2}) \\ &= a. \end{aligned}$$

Kapitel 3

Differenzen, Differenzenquotienten, Taylor-Reihen

In vielen numerischen Untersuchungen, wie bei Folgen und Reihen, Ableitungen, Interpolation, Integration sowie den uns hier interessierenden Differentialgleichungen wird das Kalkül der Differenzen und Differenzenquotienten verwendet. Deshalb sollen einige grundlegende Informationen zu den Begriffen Differenz und dividierte Differenz (Differenzenquotient) gegeben werden.

Weiterhin betrachten wir kurz die Taylor-Reihenentwicklung, die bei der Numerik der gDGL für zahlreiche Herleitungen und Beweise gebraucht wird.

3.1 Differenzen

Für die Notation von Differenzenformeln benötigt man Stützstellen und Stützwerte, die man in ihrer Gesamtheit als Referenz bezeichnen kann.

Da sind zunächst die Argumente (Abszissenwerte), die entweder als äquidistante Stützstellen $\dots, t - 2h, t - h, t, t + h, t + 2h, \dots$ mit der konstanten Schrittweite $h > 0$ bzw. in der Form $t_i = t_{i-1} + h$ oder als nicht äquidistante Stützstellen $\dots < t_{i-1} < t_i < t_{i+1} < \dots$ mit den variablen Schrittweiten $h_i = t_{i+1} - t_i > 0$ vorliegen. Dazu nehmen wir die entsprechenden Funktionswerte $f(t + ih)$ bzw. $f(t_i)$.

Wir notieren die Differenzen mit den äquidistanten Stützstellen $t + kh$.

Definition 3.1 Vorwärtsdifferenzen Δ^k bzw. Rückwärtsdifferenzen ∇^k

Für die k -ten Differenzen ($k \geq 0$) definiert man die Beziehungen

$$\begin{aligned}\Delta^0 f(t) &= f(t), \\ \Delta^1 f(t) &= \Delta f(t) = f(t+h) - f(t), \quad h > 0, \\ \Delta^n f(t) &= \Delta(\Delta^{n-1} f(t)) = \Delta^{n-1}(\Delta f(t)), \\ \nabla^0 f(t) &= f(t), \\ \nabla^1 f(t) &= \nabla f(t) = f(t) - f(t-h) = \Delta f(t-h), \\ \nabla^n f(t) &= \nabla(\nabla^{n-1} f(t)) = \nabla^{n-1}(\nabla f(t)).\end{aligned}\tag{3.1}$$

Satz 3.1 *Mit der Definition 3.1 gelten die Beziehungen*

$$\begin{aligned}\Delta^k f_j &= \sum_{i=0}^k (-1)^i \binom{k}{i} f_{j+k-i}, \quad k \geq 0, \quad j \text{ ganzzahlig,} \\ \nabla^k f_j &= \sum_{i=0}^k (-1)^i \binom{k}{i} f_{j-i}, \\ \nabla^k f_i &= \Delta^k f_{i-k}, \quad k \geq 0, \quad i \text{ ganzzahlig.}\end{aligned}\tag{3.2}$$

Beweis.

(a) Den Nachweis machen wir mit vollständiger Induktion für $\Delta^k f_j$ bzgl. k , wobei auf den Induktionsanfang verzichtet und nur der allgemeine Schritt $k \rightarrow k+1$ ausgeführt wird.

$$\begin{aligned}\Delta^{k+1} f_j &= \Delta^k(\Delta f_j) = \Delta^k(f_{j+1} - f_j) \\ &= \Delta^k f_{j+1} - \Delta^k f_j \\ &= \sum_{i=0}^k (-1)^i \binom{k}{i} f_{j+1+k-i} - \sum_{i=0}^k (-1)^i \binom{k}{i} f_{j+k-i} \\ &= f_{j+1+k} + \sum_{i=1}^k (-1)^i \binom{k}{i} f_{j+1+k-i} - \sum_{i=0}^{k-1} (-1)^i \binom{k}{i} f_{j+k-i} - (-1)^k f_j \\ &= f_{j+k+1} + \sum_{i=1}^k (-1)^i \binom{k}{i} f_{j+k+1-i} - \sum_{l=1}^k (-1)^{l-1} \binom{k}{l-1} f_{j+k-(l-1)} + (-1)^{k+1} f_j \\ &= f_{j+k+1} + \sum_{i=1}^k (-1)^i \binom{k}{i} f_{j+k+1-i} + \sum_{i=1}^k (-1)^i \binom{k}{i-1} f_{j+k+1-i} + (-1)^{k+1} f_j \\ &= f_{j+k+1} + \sum_{i=1}^k (-1)^i \left[\binom{k}{i} + \binom{k}{i-1} \right] f_{j+k+1-i} + (-1)^{k+1} f_j \\ &= f_{j+k+1} + \sum_{i=1}^k (-1)^i \binom{k+1}{i} f_{j+k+1-i} + (-1)^{k+1} f_j \\ &= \sum_{i=0}^{k+1} (-1)^i \binom{k+1}{i} f_{j+k+1-i}.\end{aligned}$$

(b) Auch der Nachweis der dritten Beziehung erfolgt mit vollständiger Induktion bzgl. k für alle i .

$$\begin{aligned}k=0, \quad \nabla^0 f_i &= f_i = \Delta^0 f_i, \\ k=1, \quad \nabla^1 f_i &= f_i - f_{i-1} = \Delta^1 f_{i-1}, \\ \nabla^{k+1} f_i &= \nabla(\nabla^k f_i) = \nabla^k f_i - \nabla^k f_{i-1} \\ &= \Delta^k f_{i-k} - \Delta^k f_{i-1-k} = \Delta^k f_{i-k} - \Delta^k f_{i-k-1} \\ &= \Delta^k (f_{i-k} - f_{i-k-1}) = \Delta^k (\Delta f_{i-k-1}) = \Delta^{k+1} f_{i-k-1} \\ &= \Delta^{k+1} f_{i-(k+1)}.\end{aligned}\quad \square$$

Die Koeffizienten in den beiden ersten Summenformeln (3.2) entsprechen den Einträgen im Pascalschen Dreieck.

So wie die Differenzen Summenformeln von Funktionswerten sind, ist auch die umgekehrte Richtung möglich, aus der Linearkombination von Funktionswerten

$$\sum_{i=0}^k a_i f_i$$

die Darstellung als Vorwärtsdifferenz bzw. Rückwärtsdifferenz zu finden.

Die Koeffizienten der Vorwärtsdifferenz bestimmen wir sukzessiv mit dem Ansatz

$$\sum_{i=0}^k a_i f_i = \sum_{i=0}^k b_i \Delta^i f_0 \quad (3.3)$$

durch Einsetzen der ersten Formel (3.2) und durch einen Koeffizientenvergleich.

$$\begin{aligned} a_k f_k + a_{k-1} f_{k-1} + a_{k-2} f_{k-2} + \dots + a_1 f_1 + a_0 f_0 &= \\ &= \sum_{i=k(-1)}^0 b_i \Delta^i f_0 = \sum_{i=k(-1)}^0 b_i \sum_{j=i(-1)}^0 (-1)^{i-j} \binom{i}{i-j} f_j \\ &= b_k f_k \\ &\quad + [-b_k \binom{k}{1} + b_{k-1}] f_{k-1} \\ &\quad + [b_k \binom{k}{2} - b_{k-1} \binom{k-1}{1} + b_{k-2}] f_{k-2} \\ &\quad \dots \\ &\quad + [b_k (-1)^{k-1} \binom{k}{k-1} + b_{k-1} (-1)^{k-2} \binom{k-1}{k-2} + \dots + b_2 (-1)^1 \binom{2}{1} + b_1] f_1 \\ &\quad + [b_k (-1)^k + b_{k-1} (-1)^{k-1} + \dots + b_2 (-1)^2 + b_1 (-1)^1 + b_0] f_0. \end{aligned}$$

Daraus ergeben sich der Reihe nach die Größen

$$b_k = a_k, \quad b_j = a_j - \sum_{i=k(-1)}^{j+1} b_i (-1)^{i-j} \binom{i}{i-j}, \quad j = k-1, k-2, \dots, 0. \quad (3.4)$$

Analog kann man für die Rückwärtsdifferenz vorgehen, d.h.

$$\sum_{i=0}^k a_i f_i = \sum_{i=0}^k c_i \nabla^i f_i \quad (3.5)$$

und

$$c_k = a_k, \quad c_j = a_j - \sum_{i=k(-1)}^{j+1} c_i (-1)^{i-j} \binom{i}{i-j}, \quad j = k-1, k-2, \dots, 0. \quad (3.6)$$

Man könnte auch den Ansatz

$$\sum_{i=0}^k a_i f_i = \sum_{i=0}^k \tilde{c}_i \nabla^i f_k$$

verwenden.

3.2 Dividierte Differenzen

Differenzen und dividierte Differenzen treten u.a. im Zusammenhang mit der Polynominterpolation auf. Deshalb soll darauf kurz eingegangen werden

Definition 3.2 Dividierte Differenzen für paarweise verschiedene t_k

Der Quotient

$$f[t_1, t_0] = \frac{f(t_1) - f(t_0)}{t_1 - t_0}$$

heißt 1. dividierte Differenz von t_1 und t_0 bezüglich f .

$$f[t_2, t_1, t_0] = \frac{f[t_2, t_1] - f[t_1, t_0]}{t_2 - t_0}$$

heißt 2. dividierte Differenz von t_2, t_1, t_0 bezüglich f .

Sei die k -te dividierte Differenz bereits definiert. Dann ist

$$f[t_{k+1}, t_k, \dots, t_1, t_0] = \frac{f[t_{k+1}, \dots, t_2, t_1] - f[t_k, \dots, t_1, t_0]}{t_{k+1} - t_0} \quad (3.7)$$

die $(k+1)$ -te dividierte Differenz von t_{k+1}, \dots, t_1, t_0 bezüglich f .

Die 0. dividierte Differenz $f[t_k]$ ist einfach der Funktionswert $f(t_k)$.

Dividierte Differenzen sind symmetrisch in Bezug auf die Argumente. Sie werden auch Steigungen genannt. Neben (3.7) gilt ebenfalls

$$\begin{aligned} f[t_{k+1}, t_k, \dots, t_1, t_0] &= \frac{f[t_{k+1}, t_{k-1}, \dots, t_1, t_0] - f[t_k, \dots, t_1, t_0]}{t_{k+1} - t_k}, \\ f[t_{k+1}, t_k, \dots, t_1, t_0] &= \frac{f[t_{k+1}, \dots, t_{j+1}, t_{j-1}, \dots, t_0] - f[t_{k+1}, \dots, t_{i+1}, t_{i-1}, \dots, t_0]}{t_i - t_j}. \end{aligned} \quad (3.8)$$

Andere Schreibweisen sind

$$f[t_{k+1}, \dots, t_1, t_0] = [t_{k+1}, \dots, t_1, t_0]_f = [t_{k+1}, \dots, t_1, t_0]f = [t_{k+1}, \dots, t_1, t_0] = \delta^{k+1}f_0.$$

Um die dividierten Differenzen $f[t_0], f[t_1, t_0], \dots, f[t_{k+1}, \dots, t_1, t_0]$ systematisch und geschickt zu berechnen, bietet sich das folgende Schema an.

Schema der dividierten Differenzen ($n = 3$)

$$\begin{array}{ccccccc} & & & x_0 & f_0 & & \\ & & & x_1 - x_0 & f[x_1, x_0] & & \\ & & & x_1 & f_1 & f[x_2, x_1, x_0] & \\ x_3 - x_0 & x_2 - x_0 & & x_2 - x_1 & f[x_2, x_1] & & f[x_3, x_2, x_1, x_0] \\ & & & x_2 & f_2 & f[x_3, x_2, x_1] & \\ & & & x_3 - x_2 & f[x_3, x_2] & & \\ & & & x_3 & f_3 & & \end{array}$$

Bei der Konstruktion des Newtonschen Interpolationspolynoms wird dieses Schema verwendet.

Für das lineare und quadratische Interpolationspolynom erhält man mit den dividierten Differenzen die Darstellungen

$$\begin{aligned}
 p_1(t) &= f[t_0] + f[t_1, t_0](t - t_0), \\
 p_2(t) &= f[t_0] + f[t_1, t_0](t - t_0) + f[t_2, t_1, t_0](t - t_0)(t - t_1),
 \end{aligned}
 \tag{3.9}$$

oder für eine Referenz aus den $n + 1$ Punkten $(t_k, f(t_k)) = (t_k, f_k)$, $k = 0, 1, \dots, n$, die allgemeine Form

$$p_n(t) = \sum_{k=0}^n \delta^k f_0 \prod_{j=0}^{k-1} (t - t_j), \quad \text{wobei} \quad \prod_{j=0}^{-1} = 1.
 \tag{3.10}$$

Wir kommen wieder auf die Differenzen zurück und gehen nun von äquidistanten Stützstellen mit dem Abstand $h > 0$ aus.

Zunächst notieren wir Differenzenschemata mittels der Vorwärtsdifferenzen

$$\begin{aligned}
 \Delta^0 f_k &= f_k, \\
 \Delta^1 f_k &= \Delta f_k = f_{k+1} - f_k, \\
 \Delta^j f_k &= \Delta(\Delta^{j-1} f_k) = \Delta^{j-1}(\Delta f_k) = \Delta^{j-1} f_{k+1} - \Delta^{j-1} f_k,
 \end{aligned}$$

bzw. Rückwärtsdifferenzen

$$\begin{aligned}
 \nabla^0 f(t) &= f(t), \\
 \nabla^1 f(t) &= \nabla f(t) = f(t) - f(t - h) = \Delta f(t - h), \\
 \nabla^j f(t) &= \nabla(\nabla^{j-1} f(t)) = \nabla^{j-1}(\nabla f(t)).
 \end{aligned}$$

Darauf basieren die Interpolationspolynome, die ihrerseits in die Mehrschrittformeln für gDGI eingehen. Die Referenz aus den n Punkten sei (t_{i-k}, f_{i-k}) , $k = n - 1, n - 2, \dots, 1, 0$. Das Schema lautet

		t_k	$f_k = \Delta^0 f_k$					
		$t_{i-(n-1)}$	$f_{i-(n-1)}$					
	h	$t_{i-(n-1)}$	$f_{i-(n-1)}$	$\Delta^1 f_{i-(n-1)}$				
	$2h$	$t_{i-(n-2)}$	$f_{i-(n-2)}$	$\Delta^1 f_{i-(n-2)}$	$\Delta^2 f_{i-(n-2)}$			
	$3h$	h	$t_{i-(n-2)}$	$f_{i-(n-2)}$	$\Delta^1 f_{i-(n-2)}$	$\Delta^2 f_{i-(n-2)}$	$\Delta^3 f_{i-(n-2)}$	
	\dots	$2h$	$t_{i-(n-3)}$	$f_{i-(n-3)}$	$\Delta^1 f_{i-(n-3)}$	$\Delta^2 f_{i-(n-3)}$	$\Delta^3 f_{i-(n-3)}$	\dots
	$3h$	h	$t_{i-(n-3)}$	$f_{i-(n-3)}$	$\Delta^1 f_{i-(n-3)}$	$\Delta^2 f_{i-(n-3)}$	$\Delta^3 f_{i-(n-3)}$	\dots
	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
$(n-1)h$	\dots	\dots	\dots	\dots	\dots	\dots	\dots	$\Delta^{n-1} f_{i-(n-1)}$
	$3h$	h	\dots	\dots	\dots	\dots	\dots	\dots
	\dots	$2h$	t_{i-2}	f_{i-2}	$\Delta^1 f_{i-3}$	$\Delta^2 f_{i-3}$	$\Delta^3 f_{i-4}$	\dots
	$3h$	h	t_{i-2}	f_{i-2}	$\Delta^1 f_{i-2}$	$\Delta^2 f_{i-2}$	$\Delta^3 f_{i-3}$	\dots
	$2h$	h	t_{i-1}	f_{i-1}	$\Delta^1 f_{i-1}$	$\Delta^2 f_{i-2}$	$\Delta^3 f_{i-3}$	\dots
	h	t_i	f_i	$\Delta^1 f_{i-1}$	$\Delta^2 f_{i-2}$	$\Delta^3 f_{i-3}$	\dots	\dots

Es gilt

$$\Delta^j f_{i-(n-1)} = \sum_{k=0}^j (-1)^{j-k} \binom{j}{k} f_{i-(n-1)+k}, \quad j = 0, 1, \dots, n-1, \quad (3.11)$$

und für die dividierte Differenz

$$f[t_{i-(n-1)}, t_{i-(n-2)}, \dots, t_{i-(n-1-j)}] = \frac{\Delta^j f_{i-(n-1)}}{h^j j!}, \quad j = 0, 1, \dots, n-1. \quad (3.12)$$

Daraus folgt das Newtonsche Interpolationspolynom vom Grad $n-1$

$$\begin{aligned} p_{n-1}(t) &= f[t_{i-(n-1)}] + f[t_{i-(n-1)}, t_{i-(n-2)}] (t - t_{i-(n-1)}) + \dots + \\ &\quad + f[t_{i-(n-1)}, t_{i-(n-2)}, \dots, t_i] (t - t_{i-(n-1)})(t - t_{i-(n-2)}) \cdot \dots \cdot (t - t_{i-1}) \\ &= f_{i-(n-1)} + \sum_{j=1}^{n-1} \frac{\Delta^j f_{i-(n-1)}}{h^j j!} (t - t_{i-(n-1)}) \cdot \dots \cdot (t - t_{i-(n-j)}) \\ &= f_{i-(n-1)} + \sum_{j=1}^{n-1} \frac{\Delta^j f_{i-(n-1)}}{j!} \frac{t - t_{i-(n-1)}}{h} \cdot \dots \cdot \frac{t - t_{i-(n-j)}}{h} \\ &= f_{i-(n-1)} + \sum_{j=1}^{n-1} \Delta^j f_{i-(n-1)} \frac{s_i(s_i - 1) \cdot \dots \cdot (s_i - j + 1)}{j!}, \\ p_{n-1}(t) &= f_{i-(n-1)} + \sum_{j=1}^{n-1} \Delta^j f_{i-(n-1)} \binom{s}{j}, \quad s = s_i = \frac{t - t_{i-(n-1)}}{h}. \end{aligned} \quad (3.13)$$

Etwas anschaulicher ist die Situation bei Verwendung von Rückwärtsdifferenzen.

t_k	$f_k = \nabla^0 f_k$				
$t_{i-(n-1)}$	$f_{i-(n-1)}$				
		$\nabla^1 f_{i-(n-2)}$			
$t_{i-(n-2)}$	$f_{i-(n-2)}$		$\nabla^2 f_{i-(n-3)}$		
		$\nabla^1 f_{i-(n-3)}$		$\nabla^3 f_{i-(n-4)}$	
$t_{i-(n-3)}$	$f_{i-(n-3)}$		$\nabla^2 f_{i-(n-4)}$		\dots
		$\nabla^1 f_{i-(n-4)}$		$\nabla^3 f_{i-(n-5)}$	\dots
\dots	\dots	\dots	\dots	\dots	\dots
\dots	\dots	\dots	\dots	\dots	$\nabla^{n-1} f_i$
		$\nabla^1 f_{i-2}$		$\nabla^3 f_{i-1}$	
t_{i-2}	f_{i-2}		$\nabla^2 f_{i-1}$		\dots
		$\nabla^1 f_{i-1}$		$\nabla^3 f_i$	
t_{i-1}	f_{i-1}		$\nabla^2 f_i$		
		$\nabla^1 f_i$			
t_i	f_i				

Mit der dritten Formel in (3.2) bzw. der oberen Diagonale (Zeile) des letzten Schemas ergibt sich das Polynom in der Gestalt

$$p_{n-1}(t) = f_{i-(n-1)} + \sum_{j=1}^{n-1} \nabla^j f_{i-(n-1-j)} \binom{s}{j}, \quad s = s_i = \frac{t - t_{i-(n-1)}}{h}. \quad (3.14)$$

Nimmt man jeweils die unteren Zeilen der Differenzenschemata, so erhält man ebenfalls das Interpolationspolynom.

$$\begin{aligned} p_{n-1}(t) &= f_i + \sum_{j=1}^{n-1} \frac{\Delta^j f_{i-j}}{h^j j!} (t - t_i) \cdot \dots \cdot (t - t_{i-(j-1)}) \\ &= f_i + \sum_{j=1}^{n-1} \Delta^j f_{i-j} \binom{s+j-1}{j}, \quad s = s_i = \frac{t - t_i}{h}, \\ &= f_i + \sum_{j=1}^{n-1} \Delta^j f_{i-j} (-1)^j \binom{-s}{j}, \quad \Delta^j f_{i-j} = \nabla^j f_i, \\ &= f_i + \sum_{j=1}^{n-1} \nabla^j f_i (-1)^j \binom{-s}{j} \\ &= \sum_{j=0}^{n-1} \nabla^j f_i (-1)^j \binom{-s}{j}. \end{aligned} \quad (3.15)$$

Wie schon erwähnt, kann diese Technik bei der Herleitung von expliziten und impliziten Mehrschrittverfahren verwendet werden.

Im Zusammenhang mit der Interpolation gibt es auch den Interpolationsfehler

$$R_{n-1}(t) = f(t) - p_{n-1}(t) = \frac{f^{(n)}(\xi)}{n!} (t - t_{i-(n-1)})(t - t_{i-(n-2)}) \cdot \dots \cdot (t - t_i), \quad (3.16)$$

wobei $\xi \in I = [t_{i-(n-1)}, t_i]$, vorausgesetzt $f \in \mathcal{C}^n(I)$.

Man hofft, dass mit wachsendem n bei hinreichend glatter Funktion das Interpolationspolynom $p_{n-1}(t)$ immer genauer wird, d.h. die Bedingung

$$\|f - p_{n-1}\|_{\infty} = \max_{t \in I} |f(t) - p_{n-1}(t)| \rightarrow 0$$

für $n \rightarrow \infty$ erfüllt ist.

Dabei sind natürlich einige Aspekte zu berücksichtigen:

- Glattheit der zu interpolierenden Funktion und Beschränktheit ihrer Ableitungen,
- Endlichkeit des Intervalls mit den Stützstellen t_j ,
- Verteilung der Stützstellen t_j verbunden mit der Kondition der Interpolation.

Für $t \in I$ und den äquidistanten Knoten t_j gilt

$$|R_{n-1}(t)| \leq \max_{t \in I} |f^{(n)}(t)| \frac{h^n (n-1)!}{n!} = \max_{t \in I} |f^{(n)}(t)| \frac{h^n}{n}, \quad (3.17)$$

sodass eine gute Approximation zu erwarten ist, sofern f beschränkte oder moderate Ableitungen besitzt. Kommt man aber von den gDGI, ist das genau ein Problem.

Deshalb wird zur Beschreibung der Polynominterpolation auch der Begriff ihrer Kondition eingeführt [73], um die Abhängigkeit des Interpolationspolynoms von $f(t)$ und der Referenz zu unterstreichen und zu untersuchen, wie sich Störungen in der Funktion $f(t)$ auf das Interpolationspolynom auswirken.

Insofern ist die Interpolation zwar einfach, aber keine gutartige Aufgabenstellung.

3.3 Approximation von Ableitungen mit Differenzenquotienten

Da es bei der numerischen Lösung von gDGI bzw. SysgDGI auch um die Approximation der darin auftretenden Ableitungen geht, wird dieser Abschnitt der näherungsweise Berechnung von ausgewählten Ableitungen $f^{(n)}(x)$ der Funktion $f(x)$ mittels entsprechender Differenzenquotienten $\delta_p^{(n)} f(x; h)$ der Genauigkeitsordnung $p \geq 1$ gewidmet.

In den Diskretisierungsverfahren werden die Differenzenquotienten meist in abgewandelter Form einbezogen, aber allein die Erläuterung ihrer Eigenschaften zeigt neben der schon genannten Kondition der Interpolation einen weiteren Aspekt zur Numerik gDGI.

Bei der Approximation von Ableitungen bestimmt man natürlich auch Fehlergrößen.

Definition 3.3

Gegeben sind $x \in [a, b]$, die Maschenweite $h = (b - a)/N$ und Stützstellen $x_i = a + ih$.

Absoluter Fehler

$$AF(x; h) = |f^{(n)}(x) - \delta_p^{(n)} f(x; h)|, \quad err1 = \max_{0 \leq i \leq N} AF(x_i; h). \quad (3.18)$$

Absolut-relativer Fehler

$$RF(x; h) = \frac{AF(x; h)}{1 + |f^{(n)}(x)|}, \quad err2 = \max_{0 \leq i \leq N} RF(x_i; h). \quad (3.19)$$

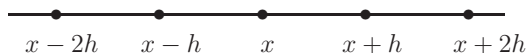
In einigen Übersichten werden ausgewählte und häufig auftretende Differenzenapproximationen zusammengestellt.

3.3.1 Einfache Differenzenquotienten

$$\begin{aligned} \delta^{(2)} f(x; h) \quad \text{zentral} &= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} = f''(x) + \frac{h^2}{12} f^{(4)}(\xi) \\ \delta f(x; h) \quad \text{zentral} &= \frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{6} f^{(3)}(\xi) \\ \delta^+ f(x; h) \quad \text{vorwärts} &= \frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2} f''(\xi) \\ \delta^- f(x; h) \quad \text{rückwärts} &= \frac{f(x) - f(x-h)}{h} = f'(x) - \frac{h}{2} f''(\xi) \end{aligned}$$

3.3.2 Symmetrische Quotienten für die 1. und 2. Ableitung

Symmetrie bzgl. der Stützstellenwahl



$$\delta_2^{(1)} f(x; h) = \frac{1}{2h}(f(x+h) - f(x-h)) = f'(x) + \frac{h^2}{6}f^{(3)}(\xi)$$

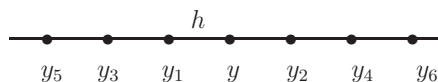
$$\delta_2^{(2)} f(x; h) = \frac{1}{h^2}(f(x+h) - 2f(x) + f(x-h)) = f''(x) + \frac{h^2}{12}f^{(4)}(\xi)$$

$$\delta_4^{(1)} f(x; h) = \frac{1}{12h}(-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)) = f'(x) - \frac{h^4}{30}f^{(5)}(\xi)$$

$$\delta_4^{(2)} f(x; h) = \frac{1}{12h^2}(-f(x+2h) + 16f(x+h) - 30f(x) + 16f(x-h) - f(x-2h)) = f''(x) - \frac{h^4}{90}f^{(6)}(\xi)$$

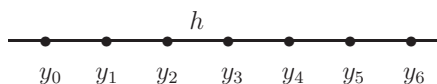
3.3.3 Symmetrische Differenzenquotienten bis zur 6. Ableitung

Symmetrie bzgl. der Stützstellenwahl außer bei einseitigen Differenzenquotienten



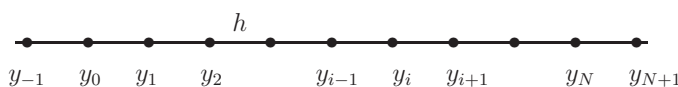
Abl. n	Differenzenquotient $\delta_p^{(n)} f(x; h)$	Ordn. p
1	$\frac{1}{h}(y_2 - y), \frac{1}{h}(y - y_1),$ 2 Stützstellen	1
	$\frac{1}{2h}(y_2 - y_1),$ 3 Stützstellen inklusive Mitte	2
	$\frac{1}{12h}(y_3 - 8y_1 + 8y_2 - y_4),$ 5 Stützstellen	4
	$\frac{1}{60h}(-y_5 + 9y_3 - 45y_1 + 45y_2 - 9y_4 + y_6),$ 7 Stützstellen	6
2	$\frac{1}{h^2}(y_1 - 2y + y_2)$	2
	$\frac{1}{12h^2}(-y_3 + 16y_1 - 30y + 16y_2 - y_4)$	4
	$\frac{1}{180h^2}(2y_5 - 27y_3 + 270y_1 - 490y + 270y_2 - 27y_4 + 2y_6)$	6
3	$\frac{1}{8h^3}(y_5 - 8y_3 + 13y_1 - 13y_2 + 8y_4 - y_6)$	4
4	$\frac{1}{6h^4}(-y_5 + 12y_3 - 39y_1 + 56y - 39y_2 + 12y_4 - y_6)$	4
5	$\frac{1}{2h^5}(-y_5 + 4y_3 - 5y_1 + 5y_2 - 4y_4 + y_6)$	2
6	$\frac{1}{h^6}(y_5 - 6y_3 + 15y_1 - 20y + 15y_2 - 6y_4 + y_6)$	2

3.3.4 Differenzenquotienten für die 1. Ableitung



Abl.	Differenzenquotient $\delta_p^{(n)} f(x; h)$	Ordn. p	Fehler $f^{(n)}(x) - \delta_p^{(n)} f(x; h)$
y'_0	$\frac{1}{h}(y_1 - y_0)$	1	$-\frac{h}{2}f^{(2)}(\xi)$
y'_1	$\frac{1}{h}(y_1 - y_0)$	1	$\frac{h}{2}f^{(2)}(\xi)$
y'_0	$\frac{1}{2h}(-3y_0 + 4y_1 - y_2)$	2	$\frac{h^2}{3}f^{(3)}(\xi)$
y'_1	$\frac{1}{2h}(-y_0 + y_2)$	2	$-\frac{h^2}{6}f^{(3)}(\xi)$
y'_2	$\frac{1}{2h}(y_0 - 4y_1 + 3y_2)$	2	$\frac{h^2}{3}f^{(3)}(\xi)$
y'_0	$\frac{1}{6h}(-11y_0 + 18y_1 - 9y_2 + 2y_3)$	3	$-\frac{h^3}{4}f^{(4)}(\xi)$
y'_1	$\frac{1}{6h}(-2y_0 - 3y_1 + 6y_2 - y_3)$	3	$\frac{h^3}{12}f^{(4)}(\xi)$
y'_2	$\frac{1}{6h}(y_0 - 6y_1 + 3y_2 + 2y_3)$	3	$-\frac{h^3}{12}f^{(4)}(\xi)$
y'_3	$\frac{1}{6h}(-2y_0 + 9y_1 - 18y_2 + 11y_3)$	3	$\frac{h^3}{4}f^{(4)}(\xi)$
y'_0	$\frac{1}{12h}(-25y_0 + 48y_1 - 36y_2 + 16y_3 - 3y_4)$	4	$\frac{h^4}{5}f^{(5)}(\xi)$
y'_1	$\frac{1}{12h}(-3y_0 - 10y_1 + 18y_2 - 6y_3 + y_4)$	4	$-\frac{h^4}{20}f^{(5)}(\xi)$
y'_2	$\frac{1}{12h}(y_0 - 8y_1 + 8y_3 - y_4)$	4	$\frac{h^4}{30}f^{(5)}(\xi)$
y'_3	$\frac{1}{12h}(-y_0 + 6y_1 - 18y_2 + 10y_3 + 3y_4)$	4	$-\frac{h^4}{20}f^{(5)}(\xi)$
y'_4	$\frac{1}{12h}(3y_0 - 16y_1 + 36y_2 - 48y_3 + 25y_4)$	4	$\frac{h^4}{5}f^{(5)}(\xi)$

3.3.5 Differenzenquotienten für die 2. und höhere Ableitungen



Abl.	Differenzenquotient $\delta_p^{(n)} f(x; h)$	Ordn. p	Fehler $f^{(n)}(x) - \delta_p^{(n)} f(x; h)$
y_i''	$\frac{1}{h^2}(y_i - 2y_{i+1} + y_{i+2})$	1	$\frac{25h}{6} f^{(3)}(\xi)$
y_i''	$\frac{1}{h^2}(y_{i-1} - 2y_i + y_{i+1})$	2	$-\frac{h^2}{12} f^{(4)}(\xi)$
y_i''	$\frac{1}{12h^2}(11y_{i-1} - 20y_i - 6y_{i+1} + 4y_{i+2} - y_{i+3})$	3	$\frac{h^3}{12} f^{(5)}(\xi)$
y_i''	$\frac{1}{12h^2}(-y_{i-2} + 16y_{i-1} - 30y_i + 16y_{i+1} - y_{i+2})$	4	$\frac{h^4}{12} f^{(6)}(\xi)$
y_i''	$\frac{1}{12h^2}(5y_{i-1} + 10y_i - 54y_{i+1} + 64y_{i+2} - 31y_{i+3} + 6y_{i+4})$	4	$-\frac{251h^4}{720} f^{(6)}(\xi)$
y_i'''	$\frac{1}{2h^3}(-y_{i-2} + 2y_{i-1} - 2y_{i+1} + y_{i+2})$	2	$-\frac{h^2}{4} f^{(5)}(\xi)$
y_0'''	$\frac{1}{2h^3}(-3y_{-1} + 10y_0 - 12y_1 + 6y_2 - y_3)$	2	$\frac{h^2}{4} f^{(5)}(\xi)$
y_N'''	$\frac{1}{2h^3}(y_{N-3} - 6y_{N-2} + 12y_{N-1} - 10y_N + 3y_{N+1})$	2	$\frac{h^2}{4} f^{(5)}(\xi)$
y_i''''	$\frac{1}{h^4}(y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2})$	2	$-\frac{h^2}{6} f^{(6)}(\xi)$
y_0''''	$\frac{1}{h^4}(y_{-1} - 4y_0 + 6y_1 - 4y_2 + y_3)$	1	$-h f^{(5)}(\xi)$
y_N''''	$\frac{1}{h^4}(y_{N-3} - 4y_{N-2} + 6y_{N-1} - 4y_N + y_{N+1})$	1	$h f^{(5)}(\xi)$

3.3.6 Berechnung von Ableitungen

Für die Berechnung der Näherungen von Ableitungen mit den äquidistanten Stützstellen $x_i = x_0 + ih$ und entsprechenden Funktionswerten macht man den Ansatz

$$f^{(n)}(x_0) = \sum_{i=-m}^k \alpha_i f(x_i) - R, \quad R = \mathcal{O}(h^p), \quad (3.20)$$

$$f(x_i) = f(x_0) + \frac{f'(x_0)}{1!}(x_i - x_0) + \frac{f''(x_0)}{2!}(x_i - x_0)^2 + \dots$$

Die Bestimmung der Größen m, k, α_i, p, R erfolgt durch Koeffizientenabgleich [66].

Beispiel 3.1

Einige interessante Beispiele für Ableitungen ergeben sich mit den Testfunktionen $f(x)$

$$\frac{2}{3}|x|^{3/2}, \quad \sin(x), \quad \frac{x^6}{720}, \quad \frac{1}{x}, \quad -x + \sin(x^2 + e^{-x}), \quad \frac{1}{x^2 + 10^{-4}}, \quad \frac{4970x - 4923}{4970x^2 - 9799x + 4830}.$$

Betrachten wir die Berechnung des Differenzenquotienten $\delta_2^{(2)}f(x; h)$ für die zweite Ableitung der rationalen Funktion [76]

$$f(x) = \frac{4970x - 4923}{4970x^2 - 9799x + 4830} \quad (3.21)$$

mit der Nullstelle $x_1 = 4923/4970 = 0.9905432595\dots$ und den nahe beieinander liegenden Polstellen $p_1 = 0.9857142823\dots$, $p_2 = 0.9859154963\dots$.

Zunächst untersuchen wir den Funktionsgraphen, den wir z.B. mit dem CAS *Mathematica* oder dem interaktiven Programm GNUPLOT erzeugen können. Eine grobe Grafik mittels

```
Plot [f[x], {x, 0, 1.2}]
bzw. plot [0:1.2] [-150:75] f(x)
```

lässt zwar die Nullstelle gut erkennen, aber die Polstellen fehlen.

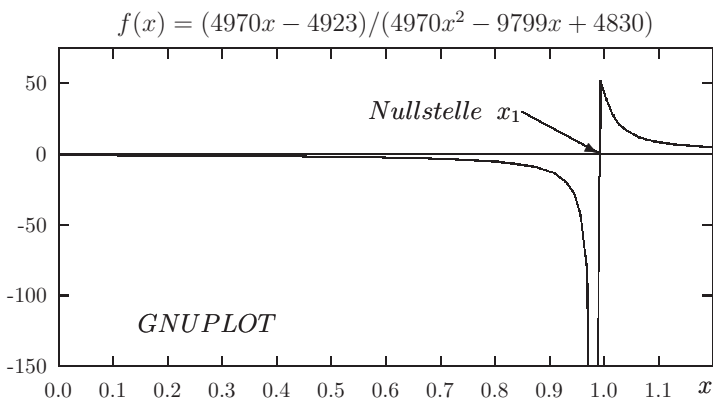
Geht man mittels Fenstertechnik langsam in den interessanten Bereich, z.B. mit

```
Plot [f[x], {x, 0.985, 0.987}]
bzw. plot [0.9854:0.9864] [-3e6:3e6] f(x)
```

so werden die Sprungstellen (bei eingezeichneten Asymptoten) sichtbar. Aber über diese Stellen mit ihren beidseitig betragsgroßen Werten wird "interpoliert". Die Nullstelle ist nun rechts außerhalb des Fensters. Im letzten Schritt

```
plot [0.98565:0.98600] [-3e6:3e6] f(x)
```

sind die Asymptoten ebenfalls zusätzlich eingezeichnet worden.



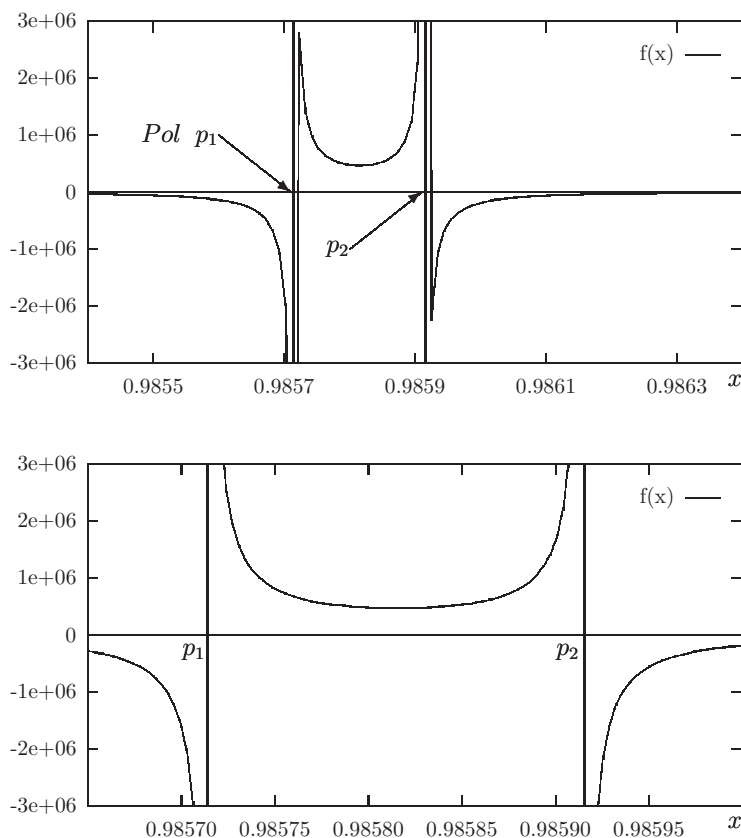


Abb. 3.1 Dateien *cg11.tex*, *cg12.tex*, *cg13.tex* (GNUPLOT: LaTeX picture),
Grafik der Funktion $f(x)$ in verschiedenen Bereichen und Fenstern

Als nächstes interessieren wir uns für die näherungsweise Berechnung und Darstellung der zweiten Ableitung von $f(x)$. Es gilt

$$f(1) = 47, \quad f'(1) = -1657, \quad f''(1) = 94, \quad f'''(1) = -27851401752,$$

$$\Delta_h^2 f(x) = \delta_2^{(2)} f(x; h) = \frac{1}{h^2} (f(x+h) - 2f(x) + f(x-h)), \quad (3.22)$$

zentraler Differenzenquotient 2. Ordnung,

$$\lim_{h \rightarrow 0} \Delta_h^2 f(x) = f''(x).$$

Wir untersuchen die grafische Darstellung von $\Delta_h^2 f(x)$ als Funktion von h bei beliebigem aber festem x .

Bei der Berechnung der notwendigen Differenzen in der Gleitkommaarithmetik (GKA) mit der Mantissenlänge t weiß man, dass die Stellenauslöschung bei $h \approx \sqrt{10^{-t}}$ beginnt. Dies steht dem Wunsch entgegen, mit möglichst sehr kleiner Schrittweite h eine gute

Genauigkeit der Approximation der zweiten Ableitung zu erreichen.

Diese Zwickmühle ist in [90] für die erste Ableitung beschrieben worden. Das Auftreten der Stellenauslöschung bei der GKA mit Mantissengenauigkeit $\varepsilon = 10^{-t}$ erfolgt für die erste Ableitung mit $\delta^\pm f(x; h)$ bei

$$h \approx 2\sqrt{\varepsilon} \sqrt{\frac{f}{f''}} \quad (3.23)$$

und kleiner.

In unserem Fall der zweiten Ableitung stehen sich

- der relative Diskretisierungsfehler $\frac{h^2 f''''}{12 f''}$
 - und der Auslöschungsfehler $\varepsilon \frac{3f}{h^2 f''}$, $\varepsilon = 10^{-t}$,
- gegenüber.

Vorausgesetzt man kann $f(x)$ mit einem relativen Fehler nahe der Mantissengenauigkeit 10^{-t} berechnen, dann erhalten wir für die Schrittweite h bei einem ausgewogenen Verhältnis der beiden Kontrahenten

$$h = \sqrt[4]{\varepsilon \frac{48f}{f''''}} = 2\sqrt[4]{\varepsilon} \sqrt[4]{\frac{3f}{f''''}}, \quad (3.24)$$

$$h \approx \sqrt[4]{\varepsilon 10^{-7}}, \quad \text{falls } x = 1.$$

Grafische Darstellungen von $\Delta_h^2 f(1)$ machen den Sachverhalt anschaulich.

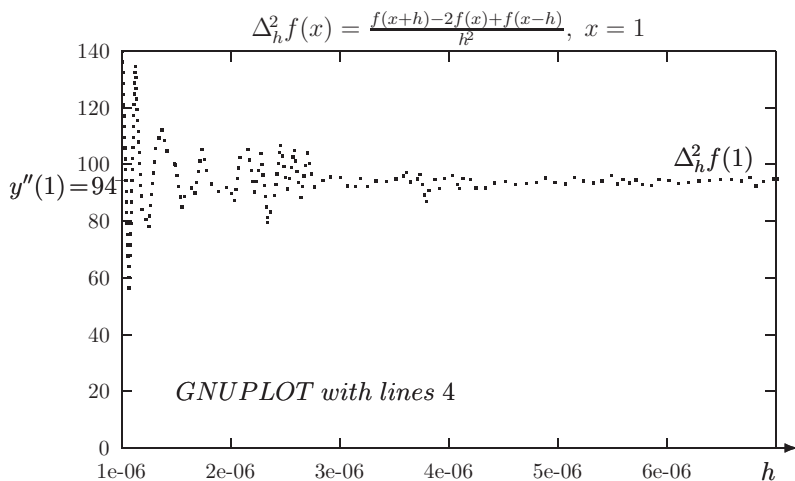


Abb. 3.2 Datei *cg21.tex*, GNUPLOT-Grafik

Für die verschiedenen GK-Formate ergeben sich aus der folgenden Tabelle zulässige Bereiche für die Schrittweite h .

GK-Format	t	h -Bereich
<i>extended</i>	19...20	$2e-7 \dots 1e-5$
<i>double</i>	15...16	$5e-7 \dots 1e-5$
<i>real</i>	11...12	$3e-5 \dots 4e-5$
<i>single</i>	7...8	versagt wegen ungenauer Berechnung von $f(x)$

Tab. 3.1 Zulässige Schrittweiten h für die Approximation der zweiten Ableitung

Jetzt wird die Punktfolge $(h, \Delta_h^2 f(1))$ für die 2. Ableitung mit einem Pascal-Programm bei verschiedenen GK-Formaten erzeugt und als Text-File `*.dat` bereitgestellt. Dieses Datenfile kann man in GNUPLOT grafisch darstellen und in analoger Weise auch als \TeX -File `*.tex` aufbereiten, um es wie hier in einen Text als Input-File einzubinden.

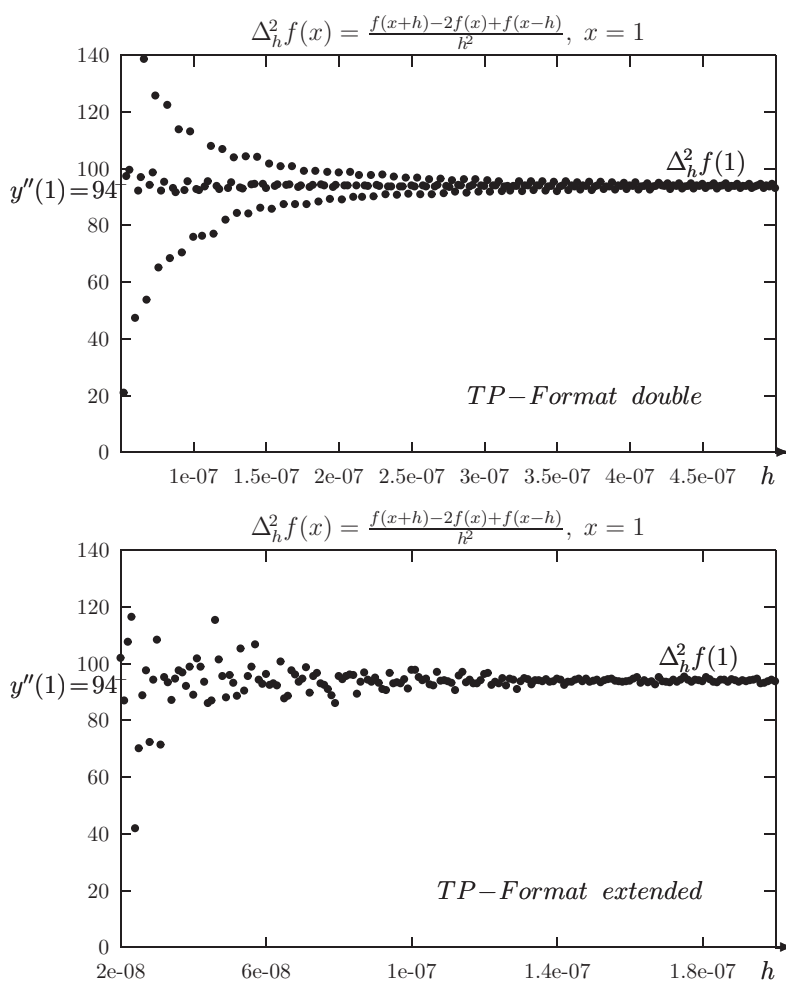


Abb. 3.3 Dateien `cg22.tex`, `cg23.tex`, GNUPLOT-Grafik von Pascal-Datei

Rechnungen in Maple

Grafische Darstellung von Funktionen mit Unstetigkeitsstellen

Definition einer rationalen Funktion mit Nullstellen und Polstellen

```
> Digits:=10:
  x:='x':
  f:=x->(4970*x-4923)/(4970*x^2-9799*x+4830):      # Funktion
  f(x):                                             # Ausdruck, Formel

> 'Nullstelle';
  ns := solve(f(x));
  evalf(ns);

'Polstellen';
  ps := [solve(4970*x^2-9799*x+4830)];
  evalf(ps);
```

Nullstelle

$$ns := \frac{4923}{4970}$$

0.9905432596

Polstellen

$$ps := \left[\frac{70}{71}, \frac{69}{70} \right]$$

[0.9859154930, 0.9857142857]

Ableitungen und Ableitungswerte

```
> Diff(f(x),x)=diff(f(x),x);
  D(f);      # = unapply(diff(f(x),x),x);
  D(f)(x);   # = diff(f(x),x);
  D(D(f));   # = (D@@2)(f);

> f(1), D(f)(1), (D@@2)(f)(1), (D@@3)(f)(1), (D@@4)(f)(1);
```

47, -1657, 94, 49371978, -27851401752

Die grobe Grafik der Funktion ist problematisch, deshalb sollte man in das kritische Gebiet mit Null- und Polstellen hineinschauen (zoomen).

```
> plot(f(x),x=0..1.2,y=-150..75,thickness=3,
  title='f(x)=(4970*x-4923)/(4970*x^2-9799*x+4830), x in [0,1.2]');

> plot(f(x),x=0.9858..1.05,y=-150..75,thickness=3,
  title='f(x), x in [0.9858,1.05]');

> plot(f(x),x=0.9854..0.9864,y=-3e6..3e6,thickness=3,
  title='f(x)=(4970*x-4923)/(4970*x^2-9799*x+4830),
  x in [0.9854,0.9864]');

> plot(f(x),x=0.98565..0.98600,y=-3e6..3e6,thickness=3,
  title='f(x), x in [0.98565,0.98600]');
```

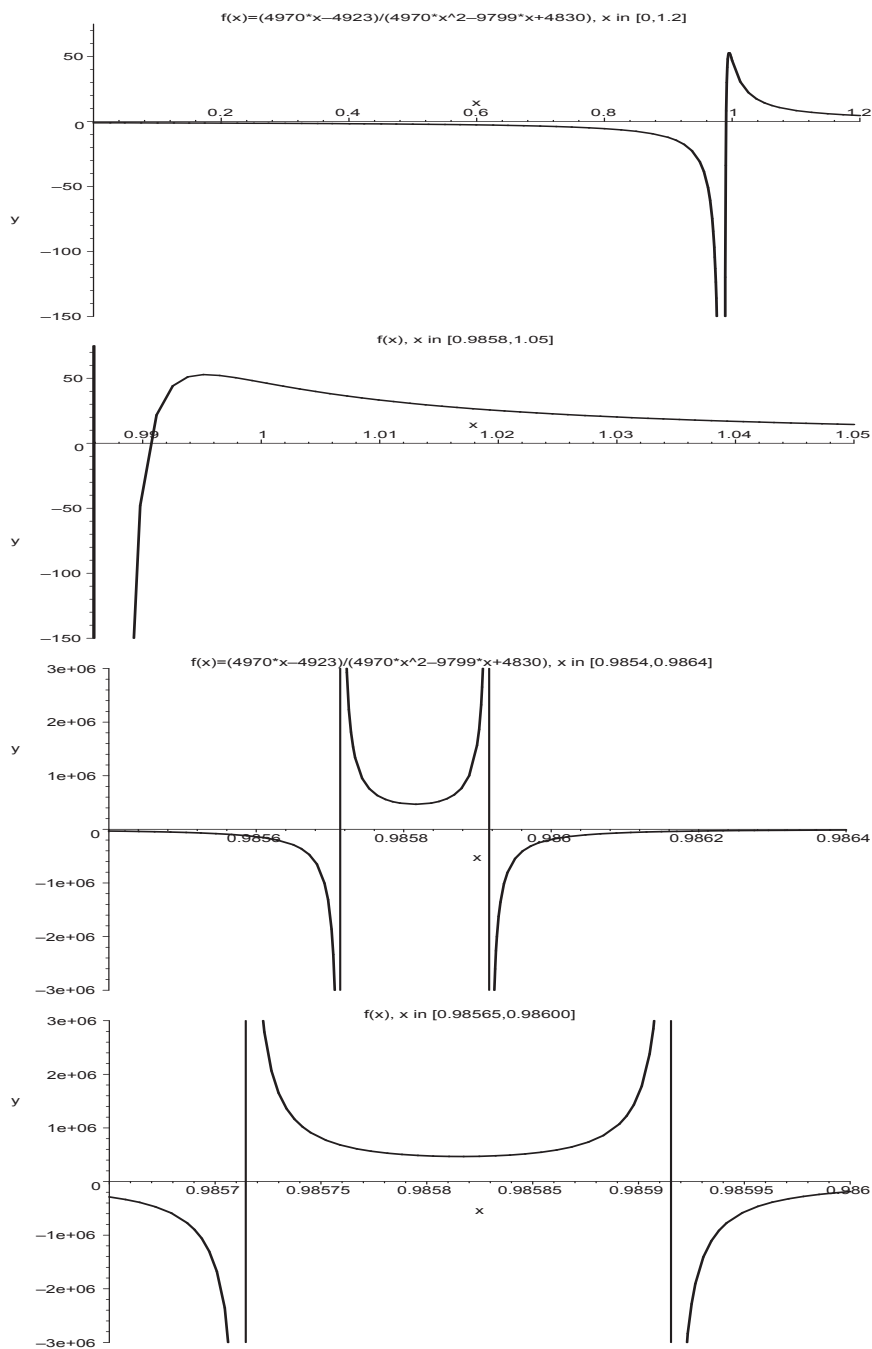



Abb. 3.4 Dateien *fabl_01.eps*, *fabl_02.eps*, *fabl_03.eps*, *fabl_04.eps*,
 Funktion $f(x) = \frac{4970x - 4923}{4970x^2 - 9799x + 4830}$ mit x aus vier Bereichen

Zweite Ableitung $f''(x)$ und ihre Approximation mit dem zentralen Differenzenquotienten
 $d2(x, h) = \Delta_h^2 f(x)$, $x = 1$, $h > 0$

```
> (D@@2)(f); (D@@2)(f)(1);
```

$$x \rightarrow -\frac{9940(9940x - 9799)}{(4970x^2 - 9799x + 4830)^2} + \frac{2(4970x - 4923)(9940x - 9799)^2}{(4970x^2 - 9799x + 4830)^3} - \frac{9940(4970x - 4923)}{(4970x^2 - 9799x + 4830)^2}$$

94

```
> Digits:=16:
```

```
d2:=(x,h)->(f(x+h)-2*f(x)+f(x-h))/h^2; # f''(1)-d2(1,h)=-h^2/12 f^(4)
d2(1,1e-5);
```

$$d2 := (x, h) \rightarrow \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

93.767900000000000

```
> # Test von Digits=5,6,... bei fester Schrittweite h=1e-5>0
```

```
n:=29:
printf(' Digits   Approximation fuer f''(1)\n'):
for i from 5 by 1 to n do
  Digits:=i:
  printf(' %2d     %25.16f \n',i,d2(1,1e-5)):
end do:
```

Digits	Approximation fuer f''(1)
5	0.0000000000000000
6	0.0000000000000000
7	0.0000000000000000
8	460000.0000000000000000
9	467000.0000000000000000
10	467300.0000000000000000
11	-2730.0000000000000000
12	-2726.0000000000000000
13	93.8000000000000000
14	93.7700000000000000
15	93.7680000000000000
16	93.7679000000000000
17	93.7679100000000000
18	93.7679050000000000
19	93.7679047000000000
20	93.7679047600000000
21	93.7679047550000000
22	93.7679047547000000
23	93.7679047546500000
24	93.7679047546510000
25	93.7679047546509000
26	93.7679047546509500
27	93.7679047546509550
28	93.7679047546509551
29	93.7679047546509551

Grafische Darstellung der Approximation mit wachsender Genauigkeit

```
> h:='h': y:='y':
p2:=plot([[5e-7,94],[5e-6,94]],color=black,thickness=3):
p3:=textplot([[5e-7,94,'94'],[5e-7,97,'_']],align=LEFT):

Digits:=10:
p11:=plot(d2(1,h),h=5e-7..5e-6,y=0..140,color=blue):
plots[display]([p2,p11,p3],labels=['h',''],title=
'Approximation von f''(1)=94 mit zentr. DQ d2(1,h) bei h->0,D=10..14');

Digits:=15:
p12:=plot(d2(1,h),h=5e-7..5e-6,y=0..140,color=blue):
plots[display]([p2,p12,p3],labels=['h',''],title=
'Approximation von f''(1)=94 mit zentr. DQ d2(1,h) bei h->0,D=15');

Digits:=16:
p13:=plot(d2(1,h),h=5e-7..5e-6,y=0..140,color=blue):
plots[display]([p2,p13,p3],labels=['h',''],title=
'Approximation von f''(1)=94 mit zentr. DQ d2(1,h) bei h->0,D=16');

Digits:=17:
p14:=plot(d2(1,h),h=5e-7..5e-6,y=0..140,color=blue):
plots[display]([p2,p14,p3],labels=['h',''],title=
'Approximation von f''(1)=94 mit zentr. DQ d2(1,h) bei h->0,D=17');

Digits:=18:
p14:=plot(d2(1,h),h=5e-7..5e-6,y=0..140,color=blue):
plots[display]([p2,p14,p3],labels=['h',''],title=
'Approximation von f''(1)=94 mit zentr. DQ d2(1,h) bei h->0,D=18..19');
```

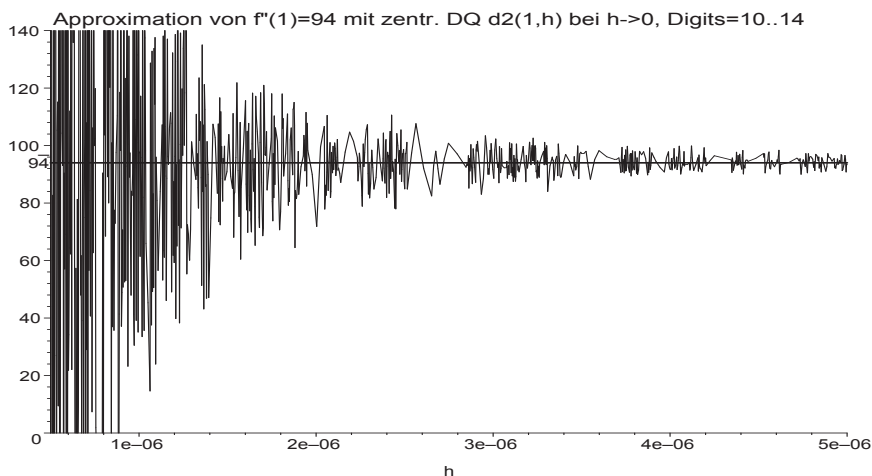


Abb. 3.5 Datei *fabl_05.eps*,

Approximation von $f''(1) = 94$ mit dem zentralen Differenzenquotienten

$\text{Digits}=10..14$,

$$d2(1,h) = \Delta_h^2 f(x) = \frac{1}{h^2}(f(x+h) - 2f(x) + f(x-h)), \quad h \in [h_0, h_1], \quad x = 1$$

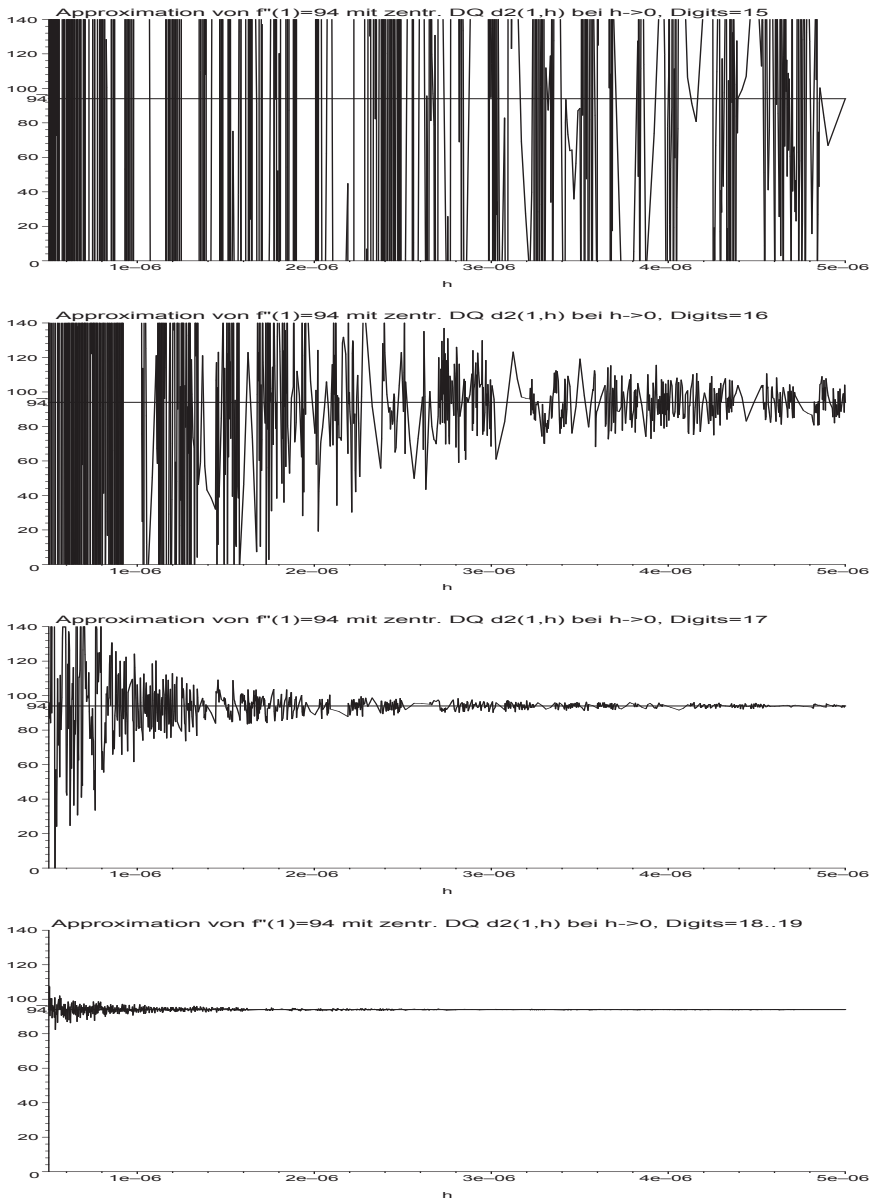


Abb. 3.6 Dateien *fabl_06.eps*, *fabl_07.eps*, *fabl_08.eps*, *fabl_09.eps*,

Approximation von $f''(1) = 94$ mit dem zentralen Differenzenquotienten,
 Digits=15,16,17,18..19 (v.o.n.u.)

Es scheint so, dass für numerische Auswertungen in Maple die standardmäßig eingestellte GKA mit `Digits=10` (entspricht dem Format *real*) relativ gut funktioniert. Zumindest liefern in den Rechnungen die Einstellungen `Digits=11,12,13,14` keine besseren Ergebnisse, unverständlicherweise manchmal sogar schlechtere, wie bei `Digits=15`. Erst mit dem Übergang zum Format *double* (`Digits=16`) sind Verbesserungen zu erwarten.

3.4 Taylor-Reihen und Ableitungen

Hier soll nur eine Reihe von Formeln aufgelistet werden.

Taylor-Reihen im \mathbb{R}^1

$$\begin{aligned} f(x) &= f(a) + \frac{f'(a)}{1!} (x-a) + \frac{f''(a)}{2!} (x-a)^2 + \dots, \\ f(x \pm h) &= f(x) \pm \frac{f'(x)}{1!} h + \frac{f''(x)}{2!} h^2 \pm \dots, \\ f(x_i) &= f(x_0) + \frac{f'(x_0)}{1!} (x_i - x_0) + \frac{f''(x_0)}{2!} (x_i - x_0)^2 + \dots \end{aligned} \quad (3.25)$$

Taylor-Polynom

$$S_n(x) = f(a) + \frac{f'(a)}{1!} (x-a) + \frac{f''(a)}{2!} (x-a)^2 + \dots + \frac{f^{(n)}(a)}{n!} (x-a)^n. \quad (3.26)$$

Fehler der Taylor-Reihe

$$\begin{aligned} R_n(x) &= f(x) - S_n(x) \\ &= \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-a)^{n+1}, \quad \xi \in (a, x) \\ &= \frac{1}{n!} \int_{t=a}^x (x-t)^n f^{(n+1)}(t) dt. \end{aligned} \quad (3.27)$$

Taylor-Reihen im \mathbb{R}^2

$$\begin{aligned} f(x+h, y+k) &= f(x, y) + \left(\frac{\partial f(x, y)}{\partial x} h + \frac{\partial f(x, y)}{\partial y} k \right) + \\ &\quad + \frac{1}{2!} \left(\frac{\partial^2 f(x, y)}{\partial x^2} h^2 + 2 \frac{\partial^2 f(x, y)}{\partial x \partial y} hk + \frac{\partial^2 f(x, y)}{\partial y^2} k^2 \right) + \\ &\quad + \frac{1}{3!} \left(\frac{\partial^3 f(x, y)}{\partial x^3} h^3 + 3 \frac{\partial^3 f(x, y)}{\partial x^2 \partial y} h^2 k + \right. \\ &\quad \left. + 3 \frac{\partial^3 f(x, y)}{\partial x \partial y^2} h k^2 + \frac{\partial^3 f(x, y)}{\partial y^3} k^3 \right) + \dots \\ &= f(x, y) + \frac{1}{1!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right) f(x, y) + \\ &\quad + \frac{1}{2!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^2 f(x, y) + \frac{1}{3!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^3 f(x, y) + \\ &\quad + \dots + \frac{1}{n!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^n f(x, y) + R_n(x, y) \\ &= \sum_{j=0}^n \frac{1}{j!} \tilde{D}^j f(x, y) + R_n(x, y), \quad \tilde{D} = h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y}, \\ f(x + \alpha h, y + \beta h) &= \sum_{j=0}^n \frac{h^j}{j!} \hat{D}^j f(x, y) + R_n(x, y), \quad \hat{D} = \alpha \frac{\partial}{\partial x} + \beta \frac{\partial}{\partial y}. \end{aligned} \quad (3.28)$$

Fehler der Taylor-Reihe

$$R(x, y) = \frac{1}{(n+1)!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^{n+1} f(x + \theta_1 h, y + \theta_2 k), \quad 0 < \theta_1, \theta_2 < 1,$$

bzw.

$$R(x, y) = \frac{h^{n+1}}{(n+1)!} \left(\alpha \frac{\partial}{\partial x} + \beta \frac{\partial}{\partial y} \right)^{n+1} f(x + \theta_1 h, y + \theta_2 h). \quad (3.29)$$

Für die gDGI $y' = f(x, y)$ mit der Funktion $y = y(x)$ ergeben sich vereinfachte Darstellungen der weiteren Ableitungen y'', y''', \dots unter Verwendung des symbolischen Differentialoperators $D = \frac{\partial}{\partial x} + f \frac{\partial}{\partial y}$.

$$y' = f(x, y(x)),$$

$$y'' = \frac{d}{dx} y' = \frac{d}{dx} f(x, y(x))$$

$$= f_x + f_y y' = f_x + f_y f = \left(\frac{\partial}{\partial x} + f \frac{\partial}{\partial y} \right) f = Df,$$

$$y''' = \frac{d}{dx} y'' = \frac{d^2}{dx^2} f(x, y(x))$$

$$= \frac{d}{dx} (f_x + f_y f)$$

$$= f_{xx} + f_{xy} y' + (f_x + f_y y') f_y + f (f_{xy} + f_{yy} y')$$

$$= f_{xx} + f_{xy} f + f_x f_y + f_y^2 f + f_{xy} f + f_{yy} f^2$$

$$= f_{xx} + 2f f_{xy} + f^2 f_{yy} + f_y (f_x + f f_y)$$

$$= \underbrace{\left(\frac{\partial^2}{\partial x^2} + 2f \frac{\partial^2}{\partial x \partial y} + f^2 \frac{\partial^2}{\partial y^2} \right)}_{D^2} f + f_y \underbrace{\left(\frac{\partial}{\partial x} + f \frac{\partial}{\partial y} \right)}_D f \quad (3.30)$$

$$= D^2 f + f_y Df,$$

$$y'''' = \frac{d}{dx} y''' = \frac{d^3}{dx^3} f(x, y(x))$$

$$= \frac{d}{dx} [f_{xx} + 2f f_{xy} + f^2 f_{yy} + f_y (f_x + f f_y)]$$

$$= D^3 f + f_y D^2 f + f_y^2 Df + 3Df Df_y,$$

$$y''''' = \frac{d}{dx} y'''' = \frac{d^4}{dx^4} f(x, y(x))$$

$$= D^4 f + 6Df D^2 f_y + 4D^2 f Df_y + D^2 f f_y^2 +$$

$$+ D^2 f f_y^3 + 3(Df)^2 f_{yy} + D^3 f f_y + 7f_y Df Df_y.$$

Kapitel 4

Einführung in die Numerik

4.1 Grundbegriffe

Im Weiteren befassen wir uns mit der numerischen Lösung gewöhnlicher Differentialgleichungen. Diese sind allgemein wie folgt definiert.

Definition 4.1

(a) Gegeben seien ein endliches (Zeit)Intervall $I = [a, b]$, ein Gebiet $\Omega \subseteq I \times \mathbb{R} \subseteq \mathbb{R}^2$ und eine hinreichend glatte Funktion $f : \Omega \rightarrow \mathbb{R}$. Eine skalare Gleichung der Form

$$x'(t) = f(t, x(t)) \quad (4.1)$$

oder kurz $x' = f(t, x)$, heißt explizite gewöhnliche Differentialgleichung 1. Ordnung. Die Gleichung

$$F(t, x(t), x'(t)) = 0 \quad (4.2)$$

heißt implizite gewöhnliche Differentialgleichung 1. Ordnung.

(b) Gegeben seien ein Gebiet $\Omega \subseteq I \times \mathbb{R}^n \subseteq \mathbb{R}^{n+1}$ und eine Funktion $f : \Omega \rightarrow \mathbb{R}$. Eine Gleichung der Form

$$x^{(n)} = f(t, x, x', \dots, x^{(n-1)}) \quad (4.3)$$

heißt explizite gewöhnliche Differentialgleichung n -ter Ordnung. Die Gleichung

$$F(t, x, x', \dots, x^{(n-1)}, x^{(n)}) = 0 \quad (4.4)$$

heißt implizite gewöhnliche Differentialgleichung n -ter Ordnung.

(c) Im Folgenden repräsentiere ein fett gedruckter Buchstabe einen Vektor der Länge n (n fest), z.B. $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$. Gegeben seien ein Gebiet $\Omega \subseteq I \times \mathbb{R}^n \subseteq \mathbb{R}^{n+1}$ und eine vektorwertige Funktion $\mathbf{f} : \Omega \rightarrow \mathbb{R}^n$, $\mathbf{f}(t, \mathbf{z}) = (f_1(t, \mathbf{z}), \dots, f_n(t, \mathbf{z}))^T$. Das Gleichungssystem

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t)) \quad (4.5)$$

heißt System expliziter gewöhnlicher Differentialgleichungen 1. Ordnung für die vektorwertige Funktion $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$.

Seine implizite Form ist

$$\mathbf{F}(t, \mathbf{x}(t), \mathbf{x}'(t)) = \mathbf{0}. \quad (4.6)$$

(d) Ist f bzw. \mathbf{f} von t unabhängig, so heißt die zugehörige Gleichung bzw. das System autonom.

Anstelle von $x'(t) = f(t, x(t))$ findet man auch andere Notationen wie $\dot{x}(t) = f(t, x(t))$, $y'(x) = f(x, y(x))$. Die Glattheit der rechten Seite notieren wir als $f \in \mathcal{C}^k(\mathbb{I} \times \mathbb{R})$.

Bemerkung 4.1 (a) Eine Funktion $x(t)$ (bzw. $\mathbf{x}(t)$) auf einem Intervall \mathbb{I} ist Lösung bzw. Trajektorie der gDGL (des SysgDGL), falls für jedes $t \in \mathbb{I}$ die entsprechende Gleichung (4.1), (4.2), ..., (4.6) erfüllt ist.

(b) Mit der Lösung $x(t)$ der gDGL bzw. $\mathbf{x}(t)$ des SysgDGL ist die Phasenkurve (Bahnkurve, Integralkurve) $(t, x(t))$ bzw. $(t, \mathbf{x}(t))$ definiert.

Das Gebiet Ω wird dann auch als Phasenraum bezeichnet.

(c) GDGL höherer Ordnung können stets als SysgDGL geschrieben werden, indem die Gleichung (4.3) umformuliert wird. Hierzu definieren wir

$$\mathbf{z}(t) = (z_1(t), z_2(t), \dots, z_n(t))^T = (x(t), x'(t), \dots, x^{(n-1)}(t))^T.$$

Die Lösung der Gleichung (4.3) ist äquivalent zur Lösung des Systems

$$\mathbf{z}'(t) = \begin{pmatrix} z_2(t) \\ z_3(t) \\ z_4(t) \\ \vdots \\ z_n(t) \\ f(t, \mathbf{z}(t)) \end{pmatrix} = \begin{pmatrix} z_2(t) & & & & & \\ & z_3(t) & & & & \\ & & z_4(t) & & & \\ & & & \ddots & & \\ & & & & z_n(t) & \\ f(t, z_1(t), z_2(t), z_3(t), \dots, z_n(t)) & & & & & \end{pmatrix} = \tilde{\mathbf{f}}(t, \mathbf{z}(t)). \quad (4.7)$$

(d) Nicht autonome gDGL und entsprechend SysgDGL können stets als autonome Systeme formuliert werden. Sie lassen sich autonomisieren. Hierzu definieren wir zu der skalaren Funktion $x(t)$ die erweiterte vektorwertige Funktion $\tilde{\mathbf{x}}(t) = (t, x(t))^T = (\tilde{x}_1(t), \tilde{x}_2(t))^T$. Die Lösung von (4.1) ist dann äquivalent zur Lösung des SysgDGL

$$\tilde{\mathbf{x}}' = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}) \quad \text{mit} \quad \tilde{\mathbf{f}}(\tilde{\mathbf{x}}) = (1, f(x_1, x_2))^T. \quad (4.8)$$

Manchmal findet man auch die Notation von $\tilde{\mathbf{x}}$ in der Reihenfolge $(x(t), t)^T$. Damit vereinfachen sich einige theoretische Betrachtungen, wenn $\tilde{\mathbf{f}}$ nicht von t abhängig ist.

Beispiel 4.1

(a) Die explizite gDGL 1. Ordnung

$$x' = -\lambda x \quad \text{mit} \quad \lambda > 0 \quad (4.9)$$

beschreibt ein exponentiell abklingendes Verhalten, wie es z.B. beim Zerfall eines radioaktiven Stoffes der Fall ist. Die allgemeine Lösung ist

$$x(t) = C e^{-\lambda t} \quad (4.10)$$

mit einer frei wählbaren Konstante C .

(b) Die implizite gDGL

$$m x''(t) + b x'(t) + D x(t) = 0 \quad (4.11)$$

beschreibt die Bewegung eines gedämpften Schwingungssystems. Ist $m \neq 0$, so lässt sie sich umformulieren als explizite gDGL zweiter Ordnung gemäß

$$x''(t) = -\frac{b}{m} x'(t) - \frac{D}{m} x(t).$$

Für kleine Dämpfungen b ist die allgemeine Lösung gegeben durch

$$x(t) = C e^{-\delta t} \sin(\omega_d t + \phi) \quad (4.12)$$

mit $\delta = b/(2m)$, $\omega_d = \sqrt{D/m - \delta^2}$ und den frei zu wählenden Konstanten C und ϕ .

(c) Ein beliebtes Anschauungsobjekt in der Theorie der SysgDGL ist das sogenannte Räuber-Beute-Modell. Dies beschreibt das Zusammenleben zweier biologischer Spezies y ("Räuber") und x ("Beute", welche von y gejagt wird). Das Zusammenleben dieser Gattungen wird beschrieben durch die beiden gDGL

$$x' = ax - bxy \quad \text{und} \quad y' = -cy + dxy \quad (4.13)$$

mit den positiven Konstanten a, b, c, d . Mit $\mathbf{x} = (x_1, x_2)^T = (x, y)^T$ kann dies zusammengefasst werden zu dem System $\mathbf{x}' = \mathbf{f}(\mathbf{x})$ mit

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} x_1(a - b x_2) \\ -x_2(c - d x_1) \end{pmatrix}.$$

(d) Gemäß Bemerkung 4.1(c) kann die gDGL zweiter Ordnung aus Teil (b) umformuliert werden in ein System 1. Ordnung. Hierzu definieren wir $\mathbf{x} = (x_1, x_2)^T$ mit $x_1(t) = x(t)$ und $x_2(t) = x'(t)$. Das lineare SysgDGL für \mathbf{x} lautet

$$\mathbf{x}'(t) = \begin{pmatrix} 0 & 1 \\ -D/m & -b/m \end{pmatrix} \mathbf{x}(t) \quad (4.14)$$

mit der rechten Seite $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$.

Wie wir in den bisherigen Beispielen gesehen haben, sind Lösungen von gDGL in der Regel nicht eindeutig. Aus einer gDGL können wir lediglich die folgende Information ablesen. Verläuft eine Lösung $x(t)$ von (4.1) zur Zeit t_0 durch einen Punkt c , also $x(t_0) = c$, so hat ihre Steigung $x'(t_0)$ dort den Wert $f(t_0, c)$. Die Tangente an die Lösung im Punkt $(t_0, x(t_0))^T$ ist damit durch den Tangentenvektor $(1, f(t_0, c))^T$ gegeben. Diese Information erlaubt es uns, den ungefähren Verlauf der Lösung mit Hilfe des Richtungsfeldes grafisch zu skizzieren und motiviert die folgende Definition.

Definition 4.2 Das zur gDGL $x' = f(t, x)$ definierte Vektorfeld

$$R(t, x) = \frac{1}{\sqrt{1 + f^2(t, x)}} \begin{pmatrix} 1 \\ f(t, x) \end{pmatrix} \quad (4.15)$$

heißt Richtungsfeld der gDGL.

Um das Richtungsfeld in jedem Gebiet $\Omega_1 \subseteq I \times \mathbb{R}$ grafisch anschaulich darzustellen, genügt die Normierung des Richtungsvektors mit seiner euklidischen Länge nicht. Man definiert sogenannte skalierte Linienelemente in einem geeigneten Gitter auf Ω_1 .

Ihre Gesamtheit ist das Richtungsfeld.

Eine Lösungskurve entsteht durch das Aneinanderreihen von Linienelementen bzw. Richtungselementen. Sie ist eindeutig durch die Vorgabe eines Punktes $(t_0, x(t_0))$, der Anfangsbedingung, bestimmt.

Aus dem Richtungsfeld gelangt man u.a. zu folgenden Erkenntnissen:

- besondere Lösungskurven,
- Polstellen oder andere besondere Stellen,
- Isoklinen, das sind Kurven, wo x' konstant ist
(bei autonomen gDGL sind die Isoklinen horizontale Linien),
- Zerfall des Gebiets Ω_1 in verschiedene Lösungsgebiete,
- orthogonale Trajektorien, das sind Kurvenscharen, die Lösungskurven orthogonal in jedem Punkt schneiden. Die orthogonale Trajektorie genügt der gDGL

$$-\frac{1}{x'} = f(t, x).$$

Als erste Variante zur Darstellung des Richtungsfeldes in MATLAB sei die Funktion `dfield` als MATLAB-Datei `dfield.m` definiert.

```
function dfield(f,xv,yv)
[x,y] = meshgrid(xv,yv);
[n,m] = size(x);
mag = min((xv(m)-xv(1))/m,(yv(n)-yv(1))/n); % fuer Skalierung
hold on
for i=1:n
    for j=1:m
        v = [1,feval(f,x(i,j),y(i,j))]' ;
        v = 0.4*mag*v/norm(v); % Normierung und Skalierung
        line([x(i,j)-v(1),x(i,j)+v(1)], [y(i,j)-v(2),y(i,j)+v(2)]);
    end;
end;
if xv(m)*xv(1)<=0 % fuer Achseneintragung
    plot([0,0], [yv(1),yv(n)], 'k-');
end;
if yv(n)*yv(1)<=0
    plot([xv(1),xv(m)], [0,0], 'k-');
end;
hold off
```

Dazu haben wir die Funktion `f0` als MATLAB-Datei `f0.m` sowie ihre Aufrufe gemäß

```
function y = f0(t,x)
y = x;

% Variante 1
dfield('f0',0:0.1:2,-1.0:0.1:2);
title('Richtungsfeld zu x'=x in [0,2]x[-1,2]')
print r_f0.ps -dpsc
```

Wir betrachten folgende gDGL mit ihren Lösungen und Richtungsfeldern.

$$\begin{aligned}
 x' &= x, & x(t) &= Ce^t, \\
 x' &= 10(x - t^2/(1+t^2)) + 2t/(1+t^2)^2, & x(t) &= Ce^{10t} + t^2/(1+t^2), \\
 x' &= \sin(1/x) - 2, \\
 x' &= x(1-x), & x(t) &= Ce^t/(1+C(e^t-1)).
 \end{aligned}
 \tag{4.16}$$

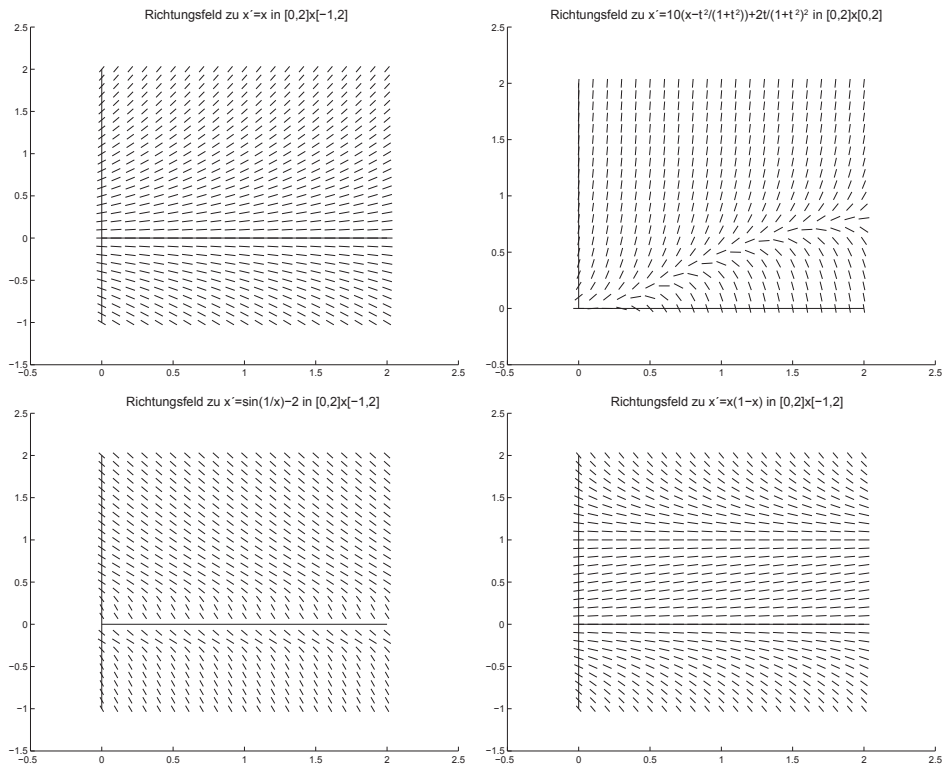


Abb. 4.1 Dateien *r_f0.ps*, *r_f1.ps*, *r_f2.ps*, *r_f4.ps*, Richtungsfelder zu den 4 gDGL, Variante `dfield`

Natürlich kann man auch das MATLAB-Kommando `quiver` verwenden.

Wir notieren es im Rahmen von zwei weiteren Versionen für die gDGL $x' = x$.

```

% Variante 2
t = 0:0.1:2;
x = -1:0.1:2;
for i = 1:length(x)
    for j = 1:length(t)
        DT(i,j) = 1;
        DX(i,j) = f0(t(j),x(i));
    end;
end;
quiver(t,x,DT,DX);
title('Richtungsfeld zu x'=x in [0,2]x[-1,2]');

```

```

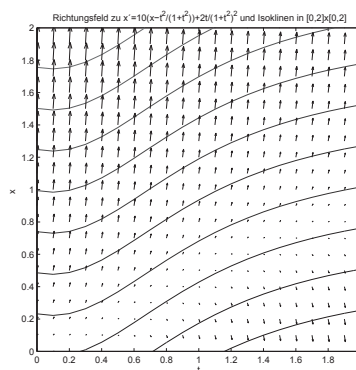
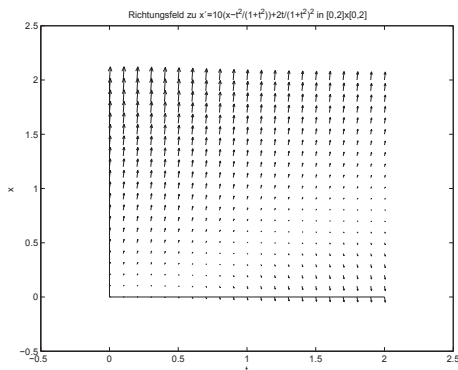
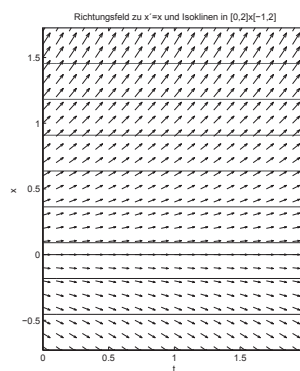
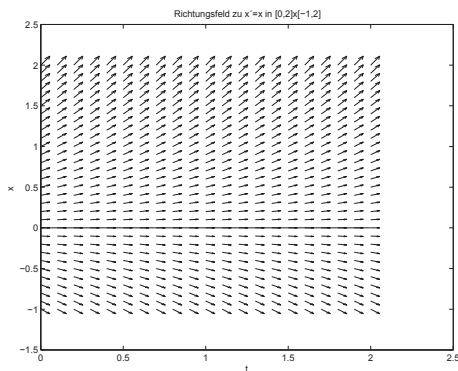
xlabel('t'); ylabel('x');
hold on
plot([0 2],[0 0],'k-',[0 0],[-1 2],'k-');
print r_f0b.ps -dpsc
hold off

% -----
% Variante 3
[tt,xx] = meshgrid(t,x);
DX = f0v(tt,xx);
DT = ones(size(DX));
contour(tt,xx,DX,10);
title('Richtungsfeld zu x'=x und Isoklinien in [0,2]x[-1,2]');
xlabel('t');
ylabel('x');
axis image
hold on
quiver(tt,xx,DT,DX);
plot([0 2],[0 0],'k-',[0 0],[-1 2],'k-');
print r_f0c.ps -dpsc
hold off

```

Die Grafik der Variante 3 hat bzgl. der Achsen den gleichen Maßstab und enthält zusätzlich zehn Isoklinien als Konturen von x' .

Dazu braucht man noch die Funktionsprozedur in vektorieller Form als MATLAB-Datei `f0v.m`, die sich wegen der Einfachheit der rechten Seite nicht von `f0.m` unterscheidet.



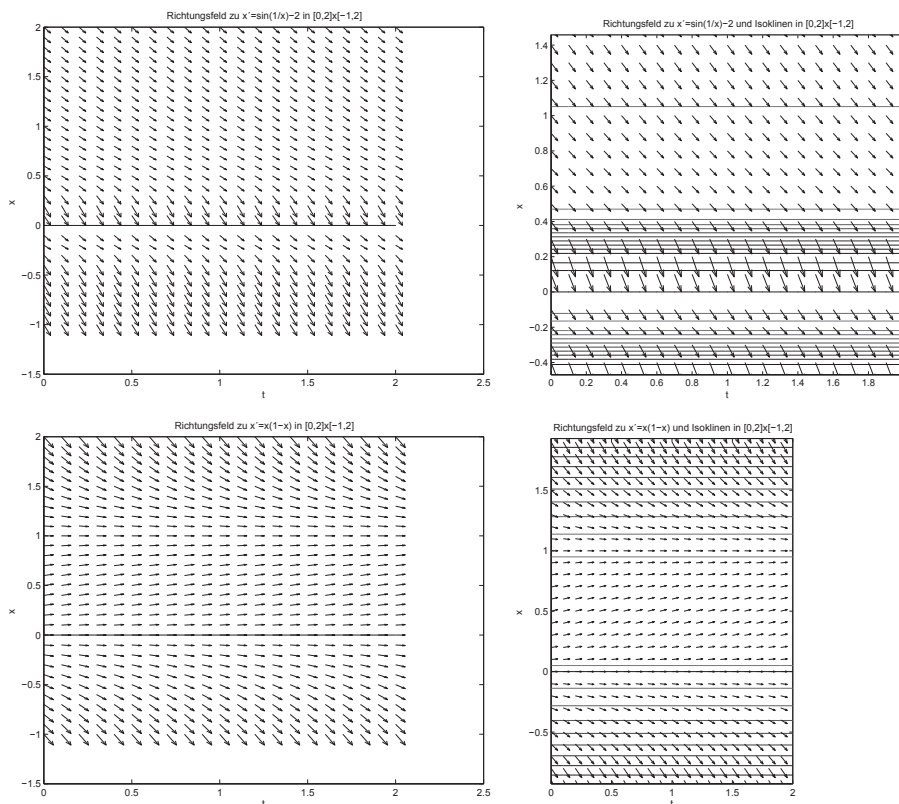


Abb. 4.2 Dateien *r_f0b.ps*, *r_f0c.ps*, *r_f1bb.ps*, *r_f1cc.ps*,
r_f2b.ps, *r_f2c.ps*, *r_f4b.ps*, *r_f4c.ps*,
 Richtungsfelder (mit Isoklinen) zu den 4 gDGL (4.16), Variante **quiver**

Wenn die Anstiege der Linienelemente sehr unterschiedlich sind, hilft auch der mögliche Skalierungsparameter in **quiver** nicht. Die eigene Definition **dfield** ist dann teilweise anschaulicher.

Ähnlich kann man in Maple vefahren.

Richtungsfeld $(1, x')^T$ zur Lösung $x(t)$ der gDGL $x' - x + t^3 - 3t + 2 = 0$, also

$$x' = f(t, x) = x - t^3 + 3t - 2.$$

```
> v:=[1,x-t^3+3*t-2]:
pp:=fieldplot(v,t=-1.5..2, x=-2..5,
  title='Richtungsfeld zu x'=x-t^3+3t-2, [-1.5,2]x[-2,5]'):
pp;
> datei1:='C:/D/neundorf/nwptexte/gdgl/maple/richt1.ps':
interface(plotdevice=ps, plotoutput=datei1,
  plotoptions='color,portrait,width=640,height=480');
plots[display](pp);
interface(plotdevice=win);
```

Andere Vorgehensweise mit Kommandos aus dem Paket DEtools (Datei *richt2.ps*)

```
> F:=diff(x(t),t)-x(t)+t^3-3*t+2:
  pts:=[[0,1/2],[0,1],[0,3/2]]: # Anfangsbedingungen
> pp:=dfieldplot({F=0},[x],t=-1.5..2,x=-2..5,
  title='Richtungsfeld zu x'=x-t^3+3t-2, [-1.5,2]x[-2,5]'):
  pp;
```

Richtungsfeld mit Lösungen (Datei *richt3.ps*)

```
> p1:=DEplot({F=0},[x(t)],t=-1.5..2,
  pts,linicolor=[red,green,blue],axes=frame,title=
  'Richtungsfeld zu x'=x-t^3+3t-2, [-1.5,2]x[-2,5], mit Loesungen'):
  p2:=plot({[-1.5,0],[2,0]],[0,-2],[0,5]},style=line,color=black):
  plots[display](p1,p2);
```

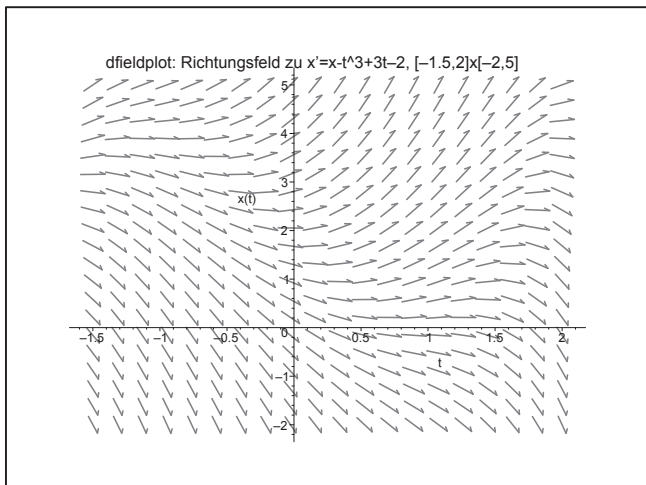
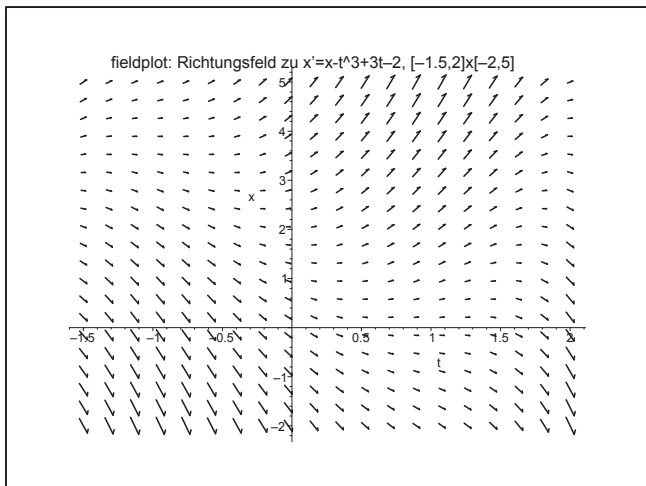


Abb. 4.3

Dateien

richt1.ps, richt2.ps,

Richtungsfelder zur gDGL

$$x' = f(t, x)$$

$$= x - t^3 + 3t - 2,$$

Maple-Kommandos

`fieldplot` (oben),

`dfieldplot` (unten)

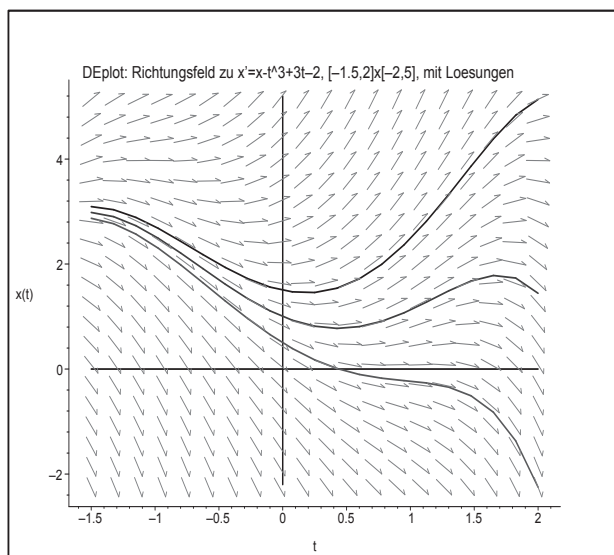


Abb. 4.4

Datei *richt3.ps*,

Richtungsfeld zur gDGL

$$x' = f(t, x)$$

$$= x - t^3 + 3t - 2$$

sowie Näherungslösungen

mit den AB $x(0) = \frac{1}{2}, 1, \frac{3}{2}$,

Maple-Kommando

DEplot

Da gDGL Spezialfälle von SysgDGL sind, wollen wir uns bei den folgenden Überlegungen auf letztere beschränken.

Eine Lösung $\mathbf{x}(t)$ von (4.5) ist unter gewissen Bedingungen an \mathbf{f} eindeutig, wenn sie an einem Punkt t_0 vorgegeben ist. Dies führt auf die folgende Definition.

Definition 4.3 *Unter einem Anfangswertproblem (AWP) verstehen wir ein SysgDGL der Form (4.5), welches ergänzt ist durch eine Bedingung der Form*

$$\mathbf{x}(t_0) = \mathbf{c} = (c_1, c_2, \dots, c_n)^T \text{ für ein } (t_0, \mathbf{c})^T \in \Omega \subseteq I \times \mathbb{R}^n \quad (4.17)$$

in einem Punkt $t_0 \in I$. Man nennt diese Anfangsbedingung (AB).

Eine gDGL mit AB generiert also einen Prozess, bei dem der jeweilige Zustand die Tendenz seiner Veränderung bestimmt.

Der folgende Satz ist ein zentrales Ergebnis der klassischen Theorie der gDGL, welches besagt, dass Lösungen von AWP eindeutig sind, sofern die rechten Seiten Lipschitz¹-stetig sind. Für einen Beweis des Satzes verweisen wir auf die Standard-Lehrbücher.

Satz 4.1 *von Picard²-Lindelöf³ über die Existenz und Eindeutigkeit von Lösungen von SysgDGL*

¹Rudolf Otto Sigismund Lipschitz (1832-1903), deutscher Mathematiker, Arbeitsgebiete waren Zahlentheorie, Theorie der Bessel-Funktionen und Fourier-Reihen, gDGL und partielle DGL

²Jean-Felix Picard (1620-1682), französischer Priester und Astronom, Arbeiten zur Messung der Länge eines Grades eines Meridians, 1665 Professor am Collège de France

³Ernst Leonard Lindelöf (1870-1946), finnischer Mathematiker, Arbeiten zur Existenz von Lösungen von DGL, analytischen Funktionen, Asymptotik von Taylor- und Fourier-Reihen sowie konforme Abbildungen

Die rechte Seite \mathbf{f} des SysgDGl (4.5) sei im folgenden Sinne Lipschitz-stetig.

Es gibt eine Konstante $L > 0$ derart, dass für alle $t \in \mathbb{R}$ und alle $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ gilt:

Ist $(t, \mathbf{x}_1)^T, (t, \mathbf{x}_2)^T \in \Omega$, so folgt die Abschätzung

$$\|\mathbf{f}(t, \mathbf{x}_1) - \mathbf{f}(t, \mathbf{x}_2)\| \leq L \|\mathbf{x}_1 - \mathbf{x}_2\|. \quad (4.18)$$

Hierbei ist $\|\cdot\|$ eine beliebige Vektornorm in \mathbb{R}^n .

Dann hat das AWP (4.5), (4.17) eine eindeutige Lösung $\mathbf{x}(t)$. Diese lässt sich bis zum Rand von I fortsetzen.

Im Folgenden sei, um die Eindeutigkeit des AWP zu gewährleisten, die Lipschitz-Stetigkeit (4.18) immer vorausgesetzt. Das SysgDGl (4.5) erzeugt eine ganze Schar von Lösungen, welche der zu (4.5) gehörige Fluss genannt wird und welche wir mit dem Symbol Φ bzw. $\Phi^{\cdot\cdot}$ bezeichnen wollen. Hierbei beschreibt

$$\mathbf{x}(t) = \mathbf{x}(t; \tau, \tilde{\mathbf{x}}) = \Phi^{t, \tau} \tilde{\mathbf{x}} \quad (4.19)$$

diejenige spezielle Lösung (Trajektorie), welche zum (Zeit)Punkt τ den Wert $\tilde{\mathbf{x}}$ annimmt, also

$$\mathbf{x}(\tau) = \Phi^{\tau, \tau} \tilde{\mathbf{x}}.$$

Insbesondere notiert man die Lösung des AWP (4.5) mit (4.17) in der Kurzform

$$\mathbf{x}(t) = \Phi^{t, t_0} \mathbf{c}. \quad (4.20)$$

Wir wollen den Fluss im skalaren Fall (4.1) grafisch illustrieren.

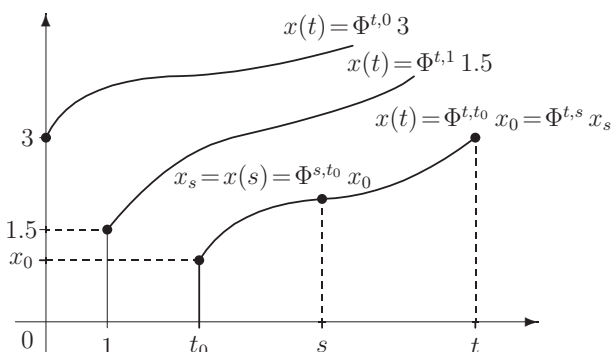


Abb. 4.5 Datei *fluss1.pic*, Lösungen $x(t)$ eines AWP als $\Phi^{t, \tau} \tilde{x}$ (4.19)

Wie sieht der Fluss eines SysgDGl aus? Sei

$$\mathbf{x}'(t) = A \mathbf{x}(t), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (4.21)$$

ein lineares homogenes SysgDGl mit konstanter Matrix A .

Seine exakte Lösung ist

$$\mathbf{x}(t) = e^{A(t-t_0)} \mathbf{x}_0,$$

was sich leicht überprüfen lässt, denn $\mathbf{x}(t_0) = \mathbf{x}_0$ und $\mathbf{x}'(t) = A e^{A(t-t_0)} \mathbf{x}_0 = A \mathbf{x}(t)$.

Der Fluss ist also die Matrixfunktion

$$\Phi^{t,t_0} = e^{A(t-t_0)}. \quad (4.22)$$

Besonders einfach ist die Darstellung, wenn A eine Diagonalmatrix ist.

Aus $A = \text{diag}(a_1, a_2, \dots, a_n)$ und bei $\mathbf{x}(t_0) = \mathbf{x}_0 = \mathbf{c} = (c_1, c_2, \dots, c_n)^T$ folgen

$$\Phi^{t,t_0} = e^{A(t-t_0)} = \text{diag}(e^{a_1(t-t_0)}, e^{a_2(t-t_0)}, \dots, e^{a_n(t-t_0)})$$

und

$$\mathbf{x}(t) = \Phi^{t,t_0} \mathbf{c} = (c_1 e^{a_1(t-t_0)}, c_2 e^{a_2(t-t_0)}, \dots, c_n e^{a_n(t-t_0)})^T.$$

Satz 4.2 *Der Fluss Φ in der Lösung $\mathbf{x}(t) = \mathbf{x}(t; t_0, \mathbf{x}_0) = \Phi^{t,t_0} \mathbf{x}_0$ besitzt offenbar folgende Eigenschaften.*

$$(a) \quad \Phi^{t_0,t_0} = I, \quad (4.23)$$

$$(b) \quad \Phi^{t,s} \Phi^{s,t_0} = \Phi^{t,t_0}. \quad (4.24)$$

Beweis.

(a) Die Formel (4.23) ist der formale Ausdruck dafür, dass die Lösung des AWP in t_0 den Wert \mathbf{x}_0 annimmt.

(b) Sei $\mathbf{x}(t; t_0, \mathbf{x}_0)$ die durch (t_0, \mathbf{x}_0) verlaufende sowie $\mathbf{x}(t; s, \mathbf{x}_s)$ die durch (s, \mathbf{x}_s) verlaufende Trajektorie.

Es sei

$$\mathbf{x}_s = \mathbf{x}(s) = \Phi^{s,t_0} \mathbf{x}_0.$$

Wegen der Eindeutigkeit der Lösung des SysgDgl gilt, dass beide eine Lösung des AWP

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t)), \quad \mathbf{x}(s) = \mathbf{x}_s,$$

darstellen, d.h. ihre Werte stimmen für $t \geq s$ überein. Somit folgt

$$\mathbf{x}(t; t_0, \mathbf{x}_0) = \mathbf{x}(t; s, \mathbf{x}_s),$$

$$\Phi^{t,t_0} \mathbf{x}_0 = \Phi^{t,s} \mathbf{x}_s = \Phi^{t,s} \Phi^{s,t_0} \mathbf{x}_0, \quad \text{also}$$

$$\Phi^{t,t_0} = \Phi^{t,s} \Phi^{s,t_0}. \quad \square$$

Bemerkung 4.2 Dass Existenz und Eindeutigkeit der Lösung nicht für allgemeine rechte Seiten erfüllt sind, wollen wir an mehreren Beispielen demonstrieren.

(a) Gegeben sei auf $\Omega = \mathbb{R}^2$ das AWP

$$x'(t) = x(t) - t^3 + 3t - 2, \quad x(t_0) = x_0 = c. \quad (4.25)$$

In den Abb. 4.3, 4.4 haben wir das Richtungsfeld der gDgl dargestellt. Wir erkennen, dass die Lösung überall oder "ewig" existiert, also global ist, und das AWP lösbar ist.

Als allgemeine Lösung der gDGI erhält man

$$x(t) = Ce^t + t^3 + 3t^2 + 3t + 5. \quad (4.26)$$

Mit der AB $x(0) = 1$ folgt $C = -4$ und die zugehörige Trajektorie lautet

$$x(t) = -4e^t + t^3 + 3t^2 + 3t + 5. \quad (4.27)$$

Damit lässt sich auch das Verhalten der Funktion für betragsgroße Werte t deuten.

(b) Gegeben sei auf $\Omega = \mathbb{R}^2$ die gDGI

$$x'(t) = x^2(t). \quad (4.28)$$

Einerseits hat man die einzige globale Lösung $x(t) \equiv 0$. Diese stellt aber eine instabile Gleichgewichtslage dar. Wenn x nicht Null ist, findet man gemäß $x'/x^2 = 1$ durch Integration die allgemeine Lösung

$$x(t) = -\frac{1}{t+C} = \frac{1}{-C-t}. \quad (4.29)$$

In der Tat sind durch diese Formel zwei Lösungen bestimmt. Die eine ist auf dem Intervall $(-\infty, -C)$, die andere auf $(-C, +\infty)$ definiert. Diese Lösungen sind maximal fortgesetzt, aber eben keine globalen Lösungen.

Wenn man mit der AB $x(t_0) = c \neq 0$ rechnet, so ist $C = -(t_0 + \frac{1}{c})$ und die Trajektorie

$$x(t) = \frac{1}{t_0 + \frac{1}{c} - t}.$$

Die Lösung existiert dann im Intervall $[t_0, t_0 + \frac{1}{c})$ der Länge $\frac{1}{c}$, sie kann nach links fortgesetzt werden und an der rechten Grenze gilt

$$\lim_{t \nearrow t_0 + \frac{1}{c}} |x(t)| = +\infty.$$

Man sagt, die Lösung “explodiert“ nach endlicher Zeit. Es macht also keinen Sinn, sie in einem größeren Intervall $[t_0, T]$ zu suchen.

Gegeben sei (4.28) auf $I = [a, b] = [0, 2]$ als AWP

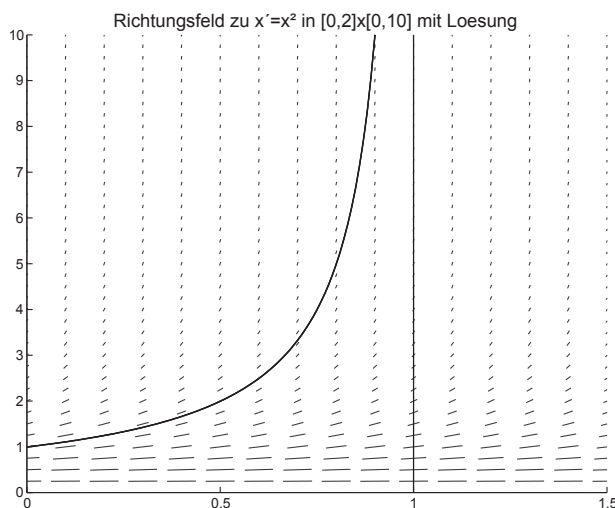
$$x'(t) = x^2(t), \quad x(0) = 1. \quad (4.30)$$

Die eindeutige Lösung

$$x(t) = \frac{1}{1-t} \quad (4.31)$$

existiert auf $[0, 1)$, auch auf $(-\infty, 1)$, aber nicht auf dem Intervall $[0, 2]$.

Für einen noch so kleinen positiven Startwert $x(t_0) = c$ explodiert die Lösung immer nach langer, aber endlicher Zeit.

**Abb. 4.6**

Datei *r-f6.ps*,
 Richtungsfeld zur gDGI
 $x' = f(t, x) = x^2$
 sowie bei $t = 1$
 explodierende Lösung
 $x(t) = 1/(1 - t)$
 zur AB $x(0) = 1$

(c) Gegeben sei auf $I = [a, b] = [0, 1]$ das AWP

$$x'(t) = 2\sqrt{x(t)}, \quad x(0) = 0. \quad (4.32)$$

Die allgemeine Lösung, die auch die triviale Nulllösung enthält, ist gegeben durch

$$x(t) = \begin{cases} (t - s)^2 & \text{für } t \geq s \geq a, \quad a = 0, \\ 0 & \text{sonst,} \end{cases} \quad (4.33)$$

mit einem frei wählbaren Parameter $s \geq a$. Die Lösung ist also nicht eindeutig.

(d) Gegeben sei auf $I = [a, b] = [0, T]$ das AWP

$$x'(t) = -\frac{1}{\sqrt{x(t)}}, \quad x(0) = 1. \quad (4.34)$$

Es beschreibt die Bewegung eines Satelliten im Gravitationsfeld der Erde unter Berücksichtigung der atmosphärischen Reibung. $x(t)$ bedeutet dabei den Abstand zum Gravitationszentrum.

Die Lösung lautet

$$x(t) = \left(1 - \frac{3}{2}t\right)^{2/3}. \quad (4.35)$$

Diese kann nach links beliebig fortgesetzt werden.

Nach rechts endet sie bei $t = T = \frac{2}{3}$, denn

$$\lim_{t \nearrow T} x(t) = 0, \quad \lim_{t \nearrow T} x'(t) = -\infty,$$

was bei einer glatten Fortsetzung über T hinaus zu unerlaubten negativen Funktionswerten bzw. ins Komplexe führen würde.

Der Gültigkeitsbereich der gDGI ist das Intervall $(-\infty, T)$.

Man sagt, die Lösung als Phasenkurve $(t, x(t))$ “kollabiert“ (fällt in sich zusammen) nach endlicher Zeit am Rand $\partial\Omega$ des erweiterten Phasenraums und notiert das als

$$\lim_{t \nearrow T} \text{dist}((t, x(t)), \partial\Omega) = 0.$$

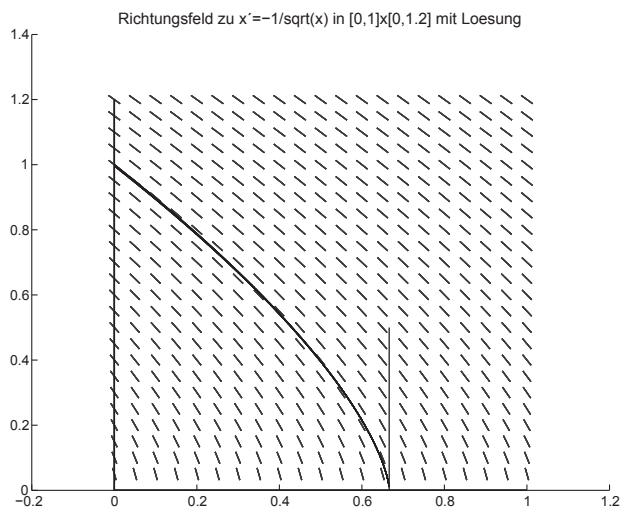


Abb. 4.7

Datei *r_f5.ps*,

Richtungsfeld zur gDGI

$$x' = f(t, x) = -\frac{1}{\sqrt{x(t)}}$$

sowie bei $t = \frac{2}{3}$

kollabierende Lösung

$$x(t) = (1 - \frac{3}{2}t)^{2/3}$$

mit der AB $x(0) = 1$

Rechnung in Maple mit DEplot

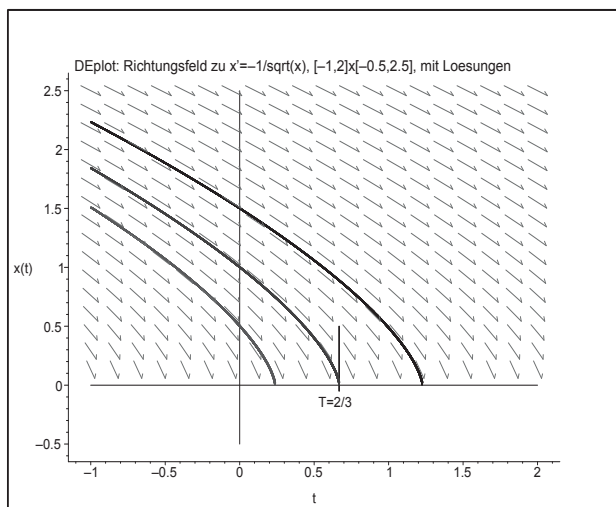


Abb. 4.8

Datei *r_f52.ps*,

Richtungsfeld zur gDGI

$$x' = f(t, x) = -\frac{1}{\sqrt{x(t)}}$$

sowie bei $t = T$

kollabierende Lösungen $x(t)$

mit den AB $x(0) = \frac{1}{2}, 1, \frac{3}{2}$

Die Rechnung kann nicht im Bereich $x(t) < 0$ ausgeführt werden und man erhält beim Aufruf von DEplot die Information

```
Warning, plot may be incomplete, the following error(s) were issued:
fractional power of a negative
```

```

> F2:=diff(x(t),t)+1/sqrt(x(t));
> pts:=[[0,1/2],[0,1],[0,3/2]]: # Anfangsbedingungen

> p1:=DEplot({F2=0},[x(t)],t=-1..2, pts, x=-0.5..2.5,
  linecolor=[red,green,blue],axes=frame,stepsize=0.0001,
  title='DEplot: Richtungsfeld zu x'=-1/sqrt(x),
  [-1,2]x[-0.5,2.5], mit Loesungen'):
p2:=plot({[[-1,0],[2,0]],[[0,-0.5],[0,2.5]],
  [[0.667,-0.05],[0.667,0.5]]}, style=line,color=black):
p3:=textplot([[0.65,-0.15,'T=2/3']]):
plots[display](p1,p2,p3);

```

(e) Betrachten wir die gDGI

$$x'(t) = \frac{x(t) - 2}{2t^2 - t}. \quad (4.36)$$

Die allgemeine Lösung ist

$$x(t) = C \frac{2t - 1}{t} + \frac{1}{t}. \quad (4.37)$$

Der Definitionsbereich der rechten Seite von (4.36) enthält nicht die t -Werte 0 und $\frac{1}{2}$. Er zerfällt somit in drei Teilgebiete. Also macht es keinen Sinn, eine AB $x(t_0) = x_0$ mit $t_0 = 0$ oder $\frac{1}{2}$ zu wählen. Aber bei der allgemeinen Lösung stellt das Argument $t = \frac{1}{2}$ plötzlich kein Problem mehr dar.

Außerdem erkennt man die triviale konstante Lösung $x(t) = 2$, die mit jeder AB $x(t_0) = 2$, $t_0 \neq 0, \frac{1}{2}$, erhalten wird und überall definiert ist.

Bei $t = \frac{1}{2}$ und speziell im Punkt $P(\frac{1}{2}, 2)$ tritt eine Singularität auf. Im Punkt P führt dies zur Situation "0/0". Das bedeutet zugleich eine Lösungsverzweigung und die Mehrdeutigkeit der Lösung.

Zum AWP (4.36) mit $x(t_0) = x_0$ ergibt sich die Lösung

$$x(t) = -\frac{2t - 1}{t} \frac{1 - t_0 x_0}{2t_0 - 1} + \frac{1}{t}, \quad (4.38)$$

wo mit der AB bei t_0 sich der Quotient $\frac{2t-1}{2t_0-1}$ einfach "wegkürzen" würde sowie bei $t_0 \neq \frac{1}{2}$ stets der Lösungswert $x(\frac{1}{2}) = 2$ entsteht.

Einige AB zu (4.36) liefern

$$\begin{aligned}
x(1) &= 1, & x(t) &= \frac{1}{t}, & x\left(\frac{1}{2}\right) &= 2, \\
x(1) &= \frac{1}{2}, & x(t) &= \frac{3}{2t} - 1, \\
x(1) &= 2, & x(t) &= 2, \\
x\left(\frac{1}{4}\right) &= 3, & x(t) &= \frac{1}{2t} + 1, \\
x\left(\frac{1}{4}\right) &= 1, & x(t) &= -\frac{3}{2t} + 3.
\end{aligned}$$

Die besondere Situation der AB bei $t_0 = \frac{1}{2}$ wird durch das Maple-Kommando `dsolve` ignoriert und nicht als Fehler angezeigt.

```

> dg11:=diff(x(t),t)-(x(t)-2)/(2*t^2-t):
  fcn:=x(t):
> init3:=x(1/2)=2:
  loes3:=dsolve({dg11,init3},{fcn});

```

$$loes3 := x(t) = \frac{(2t-1) \cdot C1}{t} + \frac{1}{t}$$

Bei der numerischen Lösung des AWP im Intervall $I=[a, b]$, wobei $\frac{1}{2} \in I$, wird durch die Diskretisierung und Rechnung im Reellen i. Allg. über die Stelle hinweg integriert. Nur wenn man zufällig diesen Wert treffen sollte, wird das Verfahren wegen der Division durch Null abbrechen.

Betrachten wir einige Ergebnisse der verschiedenen Maple-Kommandos zur Darstellung des Richtungsfeldes der gDGI sowie zusätzlich von Näherungstrajektorien für gegebene AB, wobei nur wenige Optionen der Kommandos getestet werden.

Das Kommando `dfieldplot` zeigt das Richtungsfeld mit den markanten Stellen, den Singularitäten und einigen Lösungstrajektorien mit Zusatzzweigen.

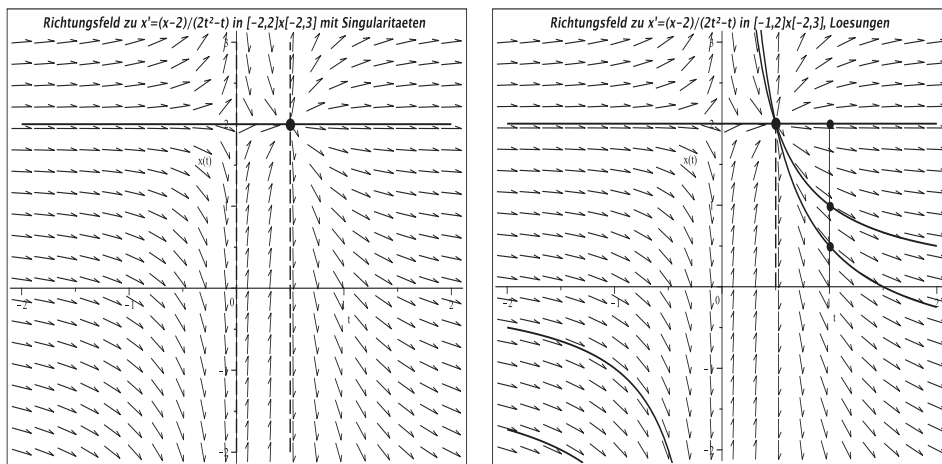


Abb. 4.9 Dateien `r_f80.ps`, `r_f81.ps`, gDGI (4.36) und AB mit Richtungsfeld, Singularitäten, Trajektorien mit Zusatzzweigen, Kommandos `dfieldplot`, `plot`, `pointplot`,

```

> p11:=dfieldplot(dg11,x(t),t=-2..2,x=-2..3,arrows=SMALL,
  titlefont=[HELVETICA,BOLD,12],
  title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [-2,2]x[-2,3]
  mit Singularitaeten'):
  p12:=plot(2,t=-2..2,color=black,thickness=2,linestyle=solid):
  p13:=plot({[[0,-2],[0,3]],[[1/2,-2],[1/2,3]]},
  color=black,thickness=2,linestyle=dash):
  p14:=pointplot([1/2,2],symbol=solidcircle,symbolsize=20,color=black):
> display(p11,p12,p13,p14);

```

```

> x10:=unapply(1/t,t):
  x11:=unapply(2,t):
  x12:=unapply(3/(2*t)-1,t):

> q11:=dfieldplot(dg11,x(t),t=-2..2,x=-2..3,
  titlefont=[HELVETICA,BOLD,12],
  title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [-1,2]x[-2,3],
  Loesungen'):
  q12:=plot([x10(t),x11(t),x12(t)],t=-2..2,
  color=blue,thickness=2,discont=true):
  q13:=pointplot([[1,1/2],[1,1],[1,2]],
  symbol=solidcircle,symbolsize=16,color=blue):
  q14:=pointplot([1/2,2],symbol=solidcircle,symbolsize=20,color=black):
  q15:=plot([[1/2,0],[1/2,2]],color=black,thickness=2,linestyle=dash):
  q16:=plot([[1,0],[1,2]],color=black,thickness=1):
> plots[display](q11,q12,q13,q14,q15,q16);

```

Das Kommando `phaseportrait`, das ebenfalls ein Richtungsfeld liefert, erfordert die zusätzliche Angabe von `AB` für Lösungstrajektorien (Näherungen), ansonsten erhält man beim Aufruf die Information

```
Error, (in DEtools/phaseportrait) initial conditions cannot be empty
```

Man vergleiche die Ergebnisse der folgenden Aufrufe bei Veränderung des t -Intervalls. Es können weitere Optionen gewisse Auswirkungen zeigen, wie z.B. `stepsize`. Und nicht zu vergessen sind die Unterschiede zwischen den Maple-Versionen, wobei die neueren zumeist besser funktionieren.

```

> phaseportrait(dg11,x(t),t=0.0001..2,[x(1)=1],color=red,linecolor=black,
  view=[0..2,0..1e4],
  title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [0,2]x[0,1e4], Naehering');
> phaseportrait(dg11,x(t),t=0.0001..2,[x(1)=1],color=red,linecolor=black,
  view=[0..2,0..1e5],
  title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [0,2]x[0,1e5], Naehering');
# Maple 12
phaseportrait(dg11,x(t),t=0.000001..2,[x(1)=1],color=red,linecolor=black,
  view=[0..2,-1.2e9..1e3],
  title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [0,2]x[-1.2e9,3], Naehering');
# Maple 14
phaseportrait(dg11,x(t),t=0.000001..2,[x(1)=1],color=red,linecolor=black,
  view=[0..2,0..1e6],
  title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [0,2]x[-1.2e9,3], Naehering');
> # Maple 12
phaseportrait(dg11,x(t),t=0..2,[x(1)=1],color=red,linecolor=black,
  view=[0..2,-1.8e8..1e3],
  title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [0,2]x[-1.8e8,1e3],Naehering');
# Maple 14
phaseportrait(dg11,x(t),t=0..2,[x(1)=1],color=red,linecolor=black,
  view=[0..2,0..25],
  title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [0,2]x[0,25], Naehering');

```

Der zweite Aufruf `phaseportrait(dg11,x(t),t=0.0001..2,...)` liefert in Maple 12 und 14 die nachfolgende Abbildung. Der `view`-Bereich wurde angepasst.

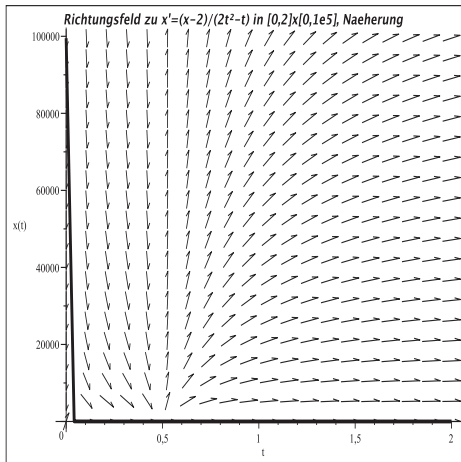


Abb. 4.10 Datei *r_f82.ps*,
Kommando `phaseportrait`
in Maple 12 und 14,
gDGI (4.36) und AB $x(1) = 1$,
Richtungsfeld und
Näherungstrajektorie

Die fünften und sechsten Aufrufe liefern in Maple 12 bzw. 14 verschiedene Bilder. In Maple 14 wird die Singularität bei $t = 0$ bemerkt und eine Warnung ausgegeben.

Warning, plot may be incomplete, the following error(s) were issued:
cannot evaluate the solution further left of .33023600e-152,
probably a singularity

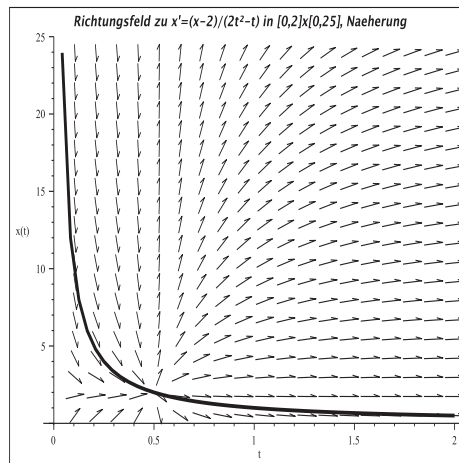
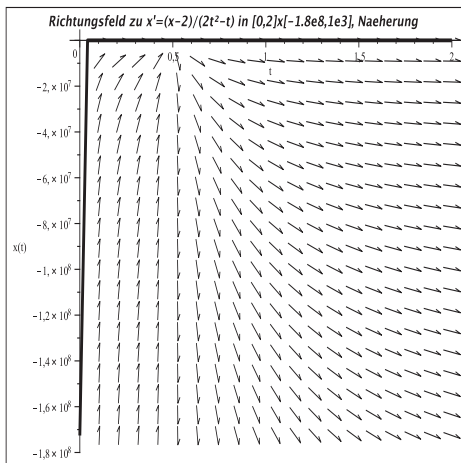


Abb. 4.11 Dateien *r_f83.ps*, *r_f83a.ps*, Kommando `phaseportrait` in Maple 12/14,
AWP (4.36), $x(1) = 1$, mit Richtungsfeld und Näherungstrajektorie

```
> pl7:=phaseportrait(dg11,x(t),t=0.5..2,[x(1)=1/2,x(1)=1,x(1)=2],
  color=red,linecolor=[red,blue,black],
  view=[0.5..2,-0.5..2],titlefont=[HELVETICA,BOLD,12],
  title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [0.5,2]x[-0.5,2],
  Naerungen'):
pl8:=pointplot([[1,1/2],[1,1],[1,2]],symbol=solidcircle,symbolsize=16,
  color=[red,blue,black]):
```



```

> p19:=plot([[1/2,0],[1/2,2]],color=black,thickness=2,linestyle=dash):
p110:=plot([[1,0],[1,2]],color=black,thickness=1):
p111:=pointplot([1/2,2],symbol=solidcircle,symbolsize=20,color=black):
> display(p17,p18,p19,p110,p111);

```

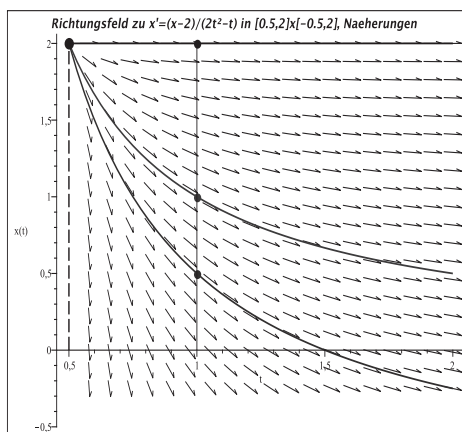


Abb. 4.12 Datei *r_f84.ps*,
Kommando `phaseportrait`,
gDgl (4.36) und AB $x(1) = \frac{1}{2}, 1, 2$,
Richtungsfeld und
Näherungstrajektorien

```

> p112:=phaseportrait(dg11,x(t),t=-0.5..2,
[x(1)=1/2,x(1)=1,x(1)=2,x(1/4)=1,x(1/4)=3],
color=red,linewidth=[red,blue,black,green,magenta], # stepsize=0.1
view=[-0.5..2,-50..80], # Maple 12
# view=[-0.5..2,-20..70] # Maple 14
titlefont=[HELVETICA,BOLD,12],
title='Richtungsfeld zu x'=(x-2)/(2t^2-t) in [-0.5,2]x[-50,80],
Naeherungen'):
> display(p112); # Option stepsize beachten

```

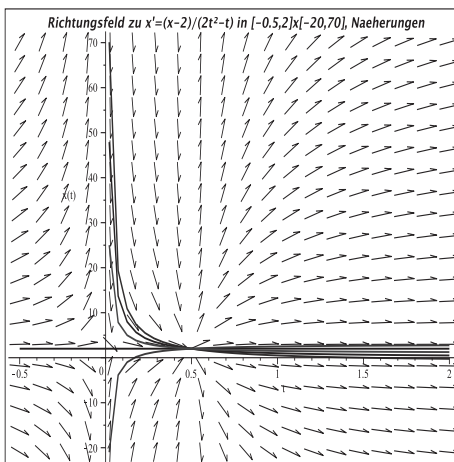
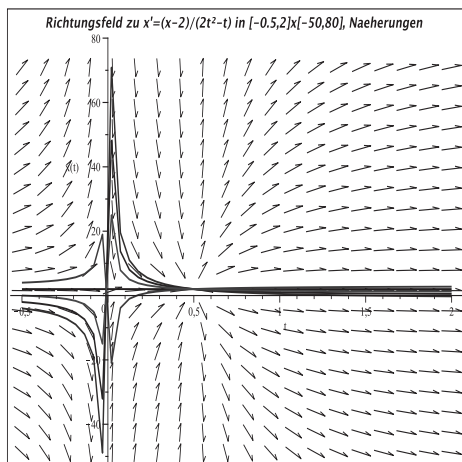


Abb. 4.13 Dateien *r_f85.ps*, *r_f85a.ps*, Kommando `phaseportrait` in Maple 12/14,
AWP (4.36), $x(1) = \frac{1}{2}, 1, 2$, $x(\frac{1}{4}) = 1, 3$, Richtungsfeld und Nähe-
rungstrajektorien, Maple 14 rechnet nicht für $t < 0$, außer bei $x(1) = 2$

Zu empfehlen ist das Kommando

```
DEplot(dg11,x(t),t=-0.5..2,x=-0.5..3,[x(1)=1/2,x(1)=1,x(1)=2,x(1/4)=1,...
```

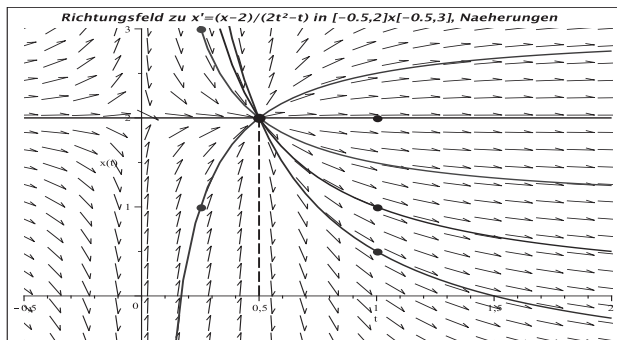


Abb. 4.14 Datei r_f86.ps, Kommando DEplot, gDGI (4.36) und AB $x(1) = \frac{1}{2}, 1, 2, x(\frac{1}{4}) = 1, 3,$ Richtungsfeld und Näherungstrajektorien

(f) In der Abb. 4.1 zeigten wir das Richtungsfeld der autonomen gDGI

$$x' = \sin\left(\frac{1}{x}\right) - 2 \tag{4.39}$$

auf dem Gebiet $\Omega_1 = [0, 2] \times [-1, 2]$. Dabei ist zu beachten, dass bei $x = 0$ keine Linienelemente gezeichnet werden können.

Es gilt $x' \in [-3, -1]$. Für $x \rightarrow 0$ hat die Ableitung x' keinen Grenzwert, weiter ist $\lim_{x \rightarrow \pm\infty} x' = -2$.

Die rechte Seite der gDGI ist auf dem Gebiet $\Omega = [0, +\infty) \times (0, +\infty)$ beliebig oft differenzierbar. Bei einer allgemeinen AB $x(t_0) = c$ in Ω folgt wegen $x' \leq -1$, dass die Lösung $x(t)$ unterhalb der Geraden $g(t) = c - (t - t_0)$ verläuft, die die t -Achse im Punkt $t_s = t_0 + c$ schneidet. Die Lösung muss sich also in einem Punkt $T \leq t_s$ der Null nähern.

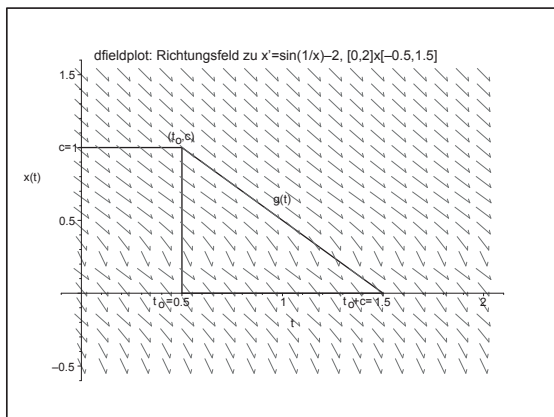


Abb. 4.15 Datei r_f71.ps, Richtungsfeld zur gDGI (4.39) sowie obere Grenze $g(t) = c - (t - t_0)$ für eine durch den Punkt (t_0, c) verlaufenden Lösung $x(t)$

Für die AB $x(0) = 1$ heißt das $x(t) \leq 1 - t$ und $T \leq t_s = 1$.

Die Grenze T bestimmt man gemäß

$$\frac{dx}{dt} = \sin\left(\frac{1}{x}\right) - 2,$$

$$\begin{aligned}
 dt &= \frac{dx}{\sin(1/x) - 2}, \\
 \int_0^T dt &= \int_1^0 \frac{dx}{\sin(1/x) - 2}, \\
 T &= \int_0^1 \frac{dx}{2 - \sin(1/x)} = 0.767\,410\,604.
 \end{aligned} \tag{4.40}$$

Aber an der kritischen Stelle $x = 0$ passiert der “Kollaps“ und die exakte Lösung ist nicht stetig fortsetzbar auf den größeren Bereich $[0, +\infty) \times (-\infty, +\infty)$.

Man erkennt dies auch am Verhalten der Linienelemente, wenn x nahe Null liegt. Ihre Anstiege “zittern“ unkontrolliert im Wertebereich $[-3, -1]$, sodass sich schwer eine Lösung aus dem Aneinanderfügen solcher Elemente erzeugen lässt.

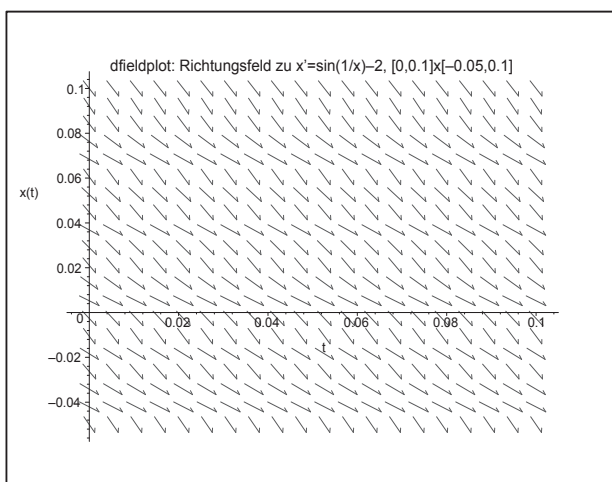


Abb. 4.16
Datei *r_f72.ps*,
Richtungsfeld
zur gDGI (4.39)
nahe $x = 0$

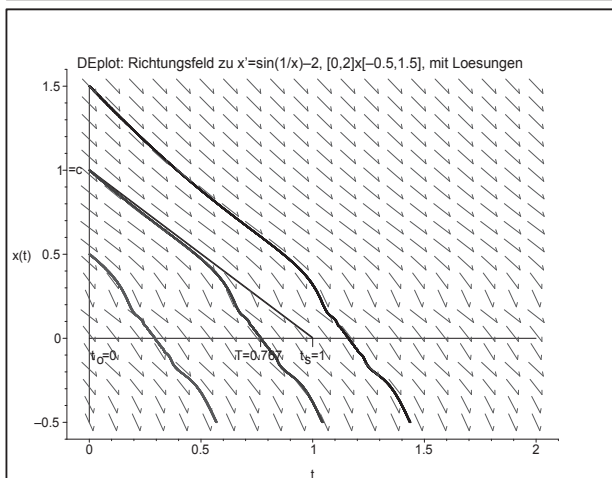


Abb. 4.17
Datei *r_f73.ps*,
Richtungsfeld
zur gDGI (4.39)
sowie bei $t = T$
eigentlich kollabierende
Näherungslösungen
mit den AB $x(0) = \frac{1}{2}, 1, \frac{3}{2}$

Dass die Näherungstrajektorien über die kritische Linie $x = 0$ hinaus berechnet werden, wie in Abb. 4.17 zu sehen ist, liegt an dem im Tool DEplot implementierten Einschrittverfahren.

Es ist eher unwahrscheinlich, dass das Näherungsverfahren genau auf einen Zeitpunkt t_i trifft, wo die Näherung η_i für $x(t_i)$ Null und damit eine Nulldivision provoziert wird. So kommt es unbeschadet über die Stelle hinweg und rechnet dann normal weiter.

Wir wollen noch einen Blick auf Ergebnisse der Schrittweitensteuerung mittels der Option `stepsize` werfen, um das wacklige Verhalten der nicht kollabierenden Näherungslösungen nahe Null zu bemerken.

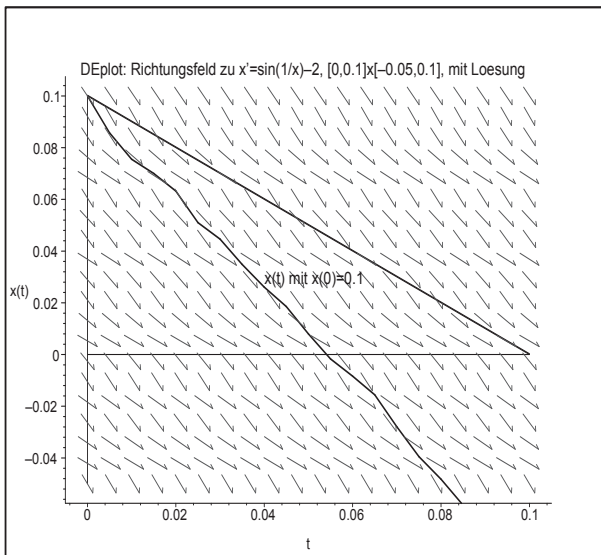


Abb. 4.18

Datei `r_f74.ps`,
Richtungsfeld
zur gDGl (4.39)
sowie Näherungslösung
zur AB $x(0) = 0.1$,
ohne `stepsize` in DEplot

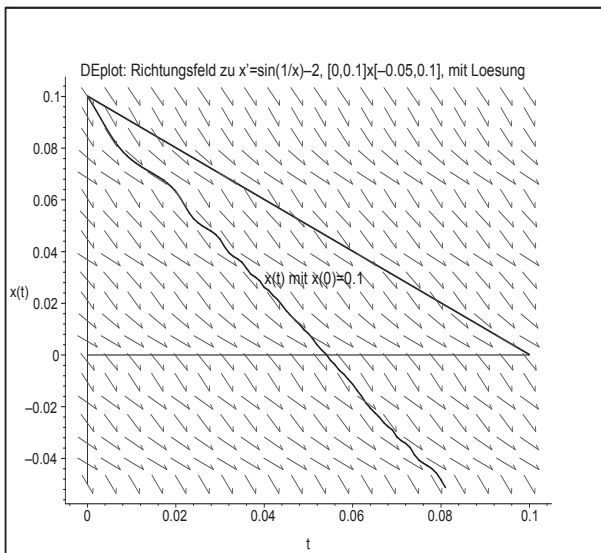


Abb. 4.19

Datei `r_f75.ps`,
Richtungsfeld
zur gDGl (4.39)
sowie Näherungslösung
zur AB $x(0) = 0.1$,
`stepsize=0.001` in DEplot

Was die Darstellung von Richtungsfeldern ohne/mit Trajektorien bzw. Näherungslösungen betrifft, bieten die heutigen CAS bequeme Lösungen an.

Dabei ist es aber noch nicht allzu lange her, dass man selbst umfangreiche Implementierungen in verschiedenen Programmiersprachen entwarf.

So gibt es für das Richtungsfeld und AWP für gDGI 1. Ordnung $y' = f(x, y)$, $y(x_0) = y_0$, das Programm RICHT_DG.PAS in der Sprache Turbo Pascal V.6.0,7.0, Borland Pascal V.7.0, dazu die Unit FOR_COM8.PAS für Funktionsparser $f(x, y)$.

Das Programm beinhaltet die Grafik des Richtungsfeldes der gDGI im auswählbaren Darstellungsbereich $[x_a, x_a + (m - 1)\Delta x] \times [y_a, y_a + (n - 1)\Delta y]$, nach Wahl die Lösung eines AWP im Bereich mit zulässigem Startpunkt $y(x_0) = y_0$ sowie den Druck von ausgewählten Bereichen des Richtungsfeldes.

Kurzcharakteristik der Oberfläche des Programms

- Menüorientiert, wahlweise Ablaufsteuerung, benutzerfreundliche Oberfläche, Fenstertechnik (4 kleine oder 1 großes Fenster), Ausdünnen des Punktgitters für Darstellung (fein, normal, grob)
- Demonstrationsvariante
- Auswahl der gDGI (Anzahl = 20) oder Funktionseingabe und -parser für die rechte Seite
- Dateiarbeit, Datei für (x, y, y') , vorhandene Beispieldateien
- Tabellarische und grafische Ausgaben
- Druck von ausgewählten Bereichen des Richtungsfeldes
- Lösung des AWP mittels klassischen Runge-Kutta-Verfahren mit dualer Schrittweitensteuerung, Auswahl der Parameter, Informationen zum Lösungsablauf und zu Ergebnissen, Grafik der Näherungslösung

Die Programmierung war schon sehr aufwendig, denn der Code des Pascal-Programms RICHT_DG.PAS umfasst ca. 2500 Zeilen, der der Unit FOR_COM8.PAS ca. 350 Zeilen.

Ähnlich wurden große Pascal-Programme STAB_ESV.PAS zur Stabilität von Einschrittverfahren bei AWP für gDGI sowie S_GEBIET.PAS zu Stabilitätsgebieten von Mehrschrittverfahren entwickelt.

Dabei ist man gezwungen, auf sehr viele Aspekte der Programmierung im Zusammenhang mit Parameterwahl, Verfahren, Skalierung, Grafik, Dateiarbeit, Darstellungsvarianten u.ä. zu achten. Der intensive Umgang mit der Materie führt jedoch zu nicht zu unterschätzenden Lerneffekten.

Übung 4.1 Weisen Sie nach, dass die in (4.33) und (4.35) definierten Funktionen Lösungen der AWP (4.32) bzw. (4.34) sind. Zeigen Sie, dass die rechten Seiten der Differentialgleichungen auf $[a, b]$ nicht Lipschitz-stetig sind und damit nicht die Voraussetzungen des Satzes 4.1 erfüllen.

4.2 Das Eulersche Polygonzugverfahren

Die Formel

$$\mathbf{x}(t) = \Phi^{t,\tau} \mathbf{x}_\tau \quad (4.41)$$

bezeichnet diejenige Trajektorie (Lösung) des SysgDGL, welche zu einem fest vorgegebenen (Zeit)Punkt τ durch einen festen Punkt \mathbf{x}_τ geht. Damit lässt sich $\mathbf{x}(t)$ in der Nähe von τ leicht durch eine Gerade approximieren. Wegen der Stetigkeit von \mathbf{f} gilt nämlich

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t)) \approx \mathbf{f}(\tau, \mathbf{x}_\tau)$$

und damit

$$\mathbf{x}(t) \approx \mathbf{x}_\tau + (t - \tau) \mathbf{f}(\tau, \mathbf{x}_\tau). \quad (4.42)$$

Wir stellen die Situation im skalaren Fall dar.

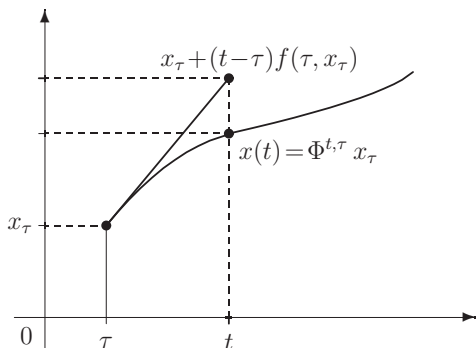


Abb. 4.20 Datei *fluss2.pic*, Approximation von $x(t)$ durch $x_\tau + (t - \tau) f(\tau, x_\tau)$

Ist \mathbf{f} in einer Umgebung von (τ, x_τ) stetig differenzierbar, so ist $\mathbf{x}(t)$ in einer Umgebung $(\tau - s, \tau + s)$ zweimal stetig differenzierbar, und es gilt

$$\mathbf{x}''(t) = \frac{d}{dt} \mathbf{f}(t, \mathbf{x}(t)) = \mathbf{f}_t(t, \mathbf{x}(t)) + \mathbf{f}_x(t, \mathbf{x}(t)) \mathbf{x}'(t) = \mathbf{f}_t(t, \mathbf{x}(t)) + \mathbf{f}_x(t, \mathbf{x}(t)) \mathbf{f}(t, \mathbf{x}(t)).$$

Nach der Taylor⁴-Formel lässt sich $\mathbf{x}(t)$ schreiben in der Form

$$\begin{aligned} \mathbf{x}(t) &= \mathbf{x}(\tau) + (t - \tau) \mathbf{f}(\tau, \mathbf{x}(\tau)) + \frac{(t - \tau)^2}{2} \mathbf{x}''(\tau + \theta(t - \tau)) \\ &= \mathbf{x}(\tau) + (t - \tau) \mathbf{f}(\tau, \mathbf{x}(\tau)) + \mathcal{O}((t - \tau)^2) \end{aligned} \quad (4.43)$$

für ein $\theta \in (0, 1)$. Diese Ideen können wir zur Konstruktion einer ersten Näherung der Lösung des AWP (4.5), (4.17) für $t \geq t_0$ benutzen. Hierzu definieren wir einen kleinen Schritt bzw. eine Schrittweite $h = \Delta t$ sowie äquidistante Knoten (Stützstellen)

$$t_i = t_0 + i h, \quad i = 0, 1, 2, \dots, \quad (4.44)$$

und approximieren die Lösung durch die folgende stetige, stückweise lineare Funktion.

⁴Brook Taylor (1685-1731), englischer Mathematiker, arbeitete auf den Gebieten der Funktionenreihen, DGL, projektiven Geometrie

Initialisierung: $\eta_0 = \mathbf{x}_h(t_0) = \mathbf{c}$,

Iteration, Rekursion: Ist $\eta_i = \mathbf{x}_h(t_i)$ gegeben, so definiere man $\mathbf{x}_h(t)$ in $[t_i, t_{i+1}]$ durch $\mathbf{x}_h(t) = \eta_i + (t - t_i) \mathbf{f}(t_i, \eta_i)$, insbesondere ist $\eta_{i+1} = \eta_i + h \mathbf{f}(t_i, \eta_i)$.

Sind wir nur an Näherungen an den Knoten t_i interessiert, so ergibt sich hieraus das schrittweise Verfahren bzw. Diskretisierungsverfahren.

Algorithmus 4.1 Eulersches Polygonzugverfahren

Man wähle eine hinreichend kleine Schrittweite h und definiere die zugehörigen Knoten $t_i = t_0 + i h$. Die Werte η_i an den Knoten t_i sind rekursiv definiert durch

$$\eta_0 = \mathbf{c}, \quad (4.45)$$

$$\eta_{i+1} = \eta_i + h k, \quad k = k_1 = \mathbf{f}(t_i, \eta_i), \quad i = 0, 1, 2, \dots \quad (4.46)$$

Es ist das einfachste explizite Einschrittverfahren (ESV). Es ist invariant bzgl. der Stelle t_i , d.h. die rechte Seite hat an jeder Stelle t_i immer dieselbe Gestalt.

Folgende Bezeichnungen für das Verfahren findet der Leser in der Literatur:

PZV, Euler-Verfahren, explizites Euler-Verfahren, Tangentenverfahren, Euler-Cauchy-Verfahren. Dieses bereits 1768 von Leonard Euler⁵ benutzte Verfahren wurde durch Augustin-Louis Cauchy⁶ 1840 theoretisch begründet.

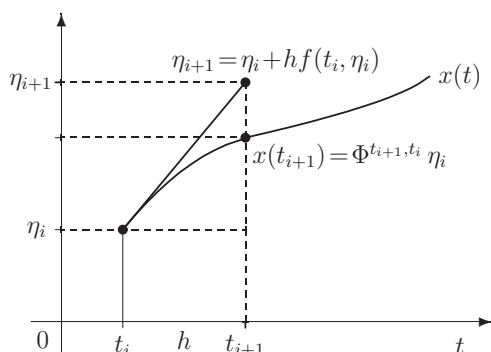


Abb. 4.21 Datei *pzv1.pic*, PZV mit Schrittformel $\eta_{i+1} = \eta_i + h f(t_i, \eta_i)$ zum AWP $x'(t) = f(t, x(t))$, $x(t_i) = \eta_i$

Beispiel 4.2

(a) Betrachten wir als AWP die lineare homogene gDGL

$$x' = \lambda x, \quad x(0) = 1, \quad \lambda \in \mathbb{C}. \quad (4.47)$$

Die Lösung des AWP ist $x(t) = e^{\lambda t}$. Sie soll mit dem PZV im Intervall $[0, T]$ approximiert werden. Hierzu wählen wir eine Anzahl $N \in \mathbb{N} \setminus \{0\}$ und die Schrittweite $h = T/N$.

⁵Leonard Euler (1707-1783), schweizer Mathematiker, wirkte in Berlin und St. Petersburg, wo er auf fast allen Gebieten der Mathematik arbeitete

⁶Augustin-Louis Cauchy (1789-1857), französischer Mathematiker, Arbeiten zur Trigonometrie, Determinanten- und Reihenlehre sowie Funktionentheorie

Mit $f(t, x) = \lambda x$ liefert der Algorithmus 4.1

$$\begin{aligned}\eta_0 &= 1, \\ \eta_{i+1} &= \eta_i + h \lambda \eta_i = (1 + h \lambda) \eta_i.\end{aligned}$$

Durch Induktion folgt

$$\eta_i = (1 + h \lambda)^i, \quad i = 0, 1, 2, \dots \quad (4.48)$$

Nun bemerkt man Folgendes.

(1) Aus der Analysis kennt man den Grenzwert

$$\left(1 + \frac{1}{r}\right)^r \rightarrow e \quad \text{für } r \rightarrow \infty.$$

Damit gilt

$$\begin{aligned}\eta_N &= \left(1 + \lambda \frac{T}{N}\right)^N = \left(\left(1 + \frac{1}{N/(\lambda T)}\right)^{N/(\lambda T)}\right)^{\lambda T} \\ &\rightarrow e^{\lambda T} = x(T) \quad \text{für } N \rightarrow \infty.\end{aligned} \quad (4.49)$$

Die Näherung η_N konvergiert damit für wachsende N gegen den exakten Wert $x(T)$.

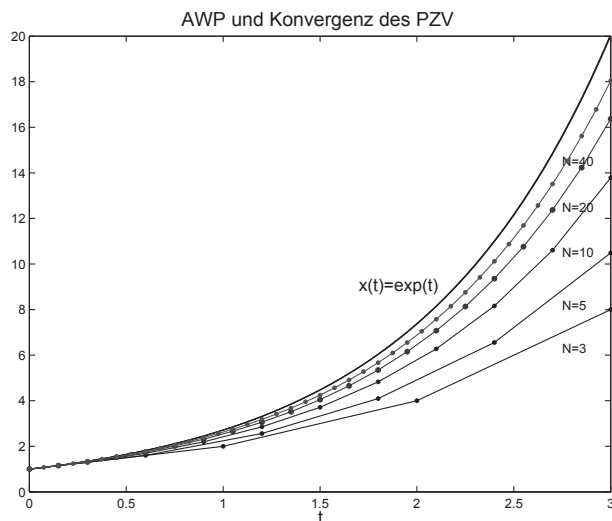


Abb. 4.22 Datei *pzv_1.ps*,
AWP $x' = \lambda x$, $x(0) = 1$,
 $0 \leq t \leq 3$, $\lambda = 1$,
mit exakter Lösung $x(t) = e^t$
und Konvergenz
des PZV $\eta_{i+1} = (1 + h \lambda) \eta_i$,
 $h = 3/N$, $N = 3, 5, 10, 20, 40$

(2) Ist λ reell und < 0 , so gilt $x(t) = e^{\lambda t} \rightarrow 0$.

Um dieses qualitative Verhalten mit dem PZV zu erhalten, darf die Schrittweite h nicht zu groß gewählt werden.

Es gilt immer $1 + h \lambda < 1$. Ist nämlich $h > 2/|\lambda|$, so ist $1 + h \lambda < -1$. In diesem Fall oszilliert die Folge η_i und es gilt

$$|\eta_i| = |1 + h \lambda|^i \rightarrow \infty \quad \text{für } i \rightarrow \infty. \quad (4.50)$$

Als notwendiges Kriterium für die Konvergenz gegen Null für $N \rightarrow \infty$ ergibt sich damit die Bedingung

$$h < \frac{2}{|\lambda|}. \quad (4.51)$$

Will man vermeiden, dass sogar $1 + h\lambda < 0$ wird, also keine Oszillationen auftreten, sondern sich ein monoton abnehmender Verlauf der Näherungslösung ergibt, so erhält man die strengere Beschränkung der Schrittweite $h < \frac{1}{|\lambda|}$.

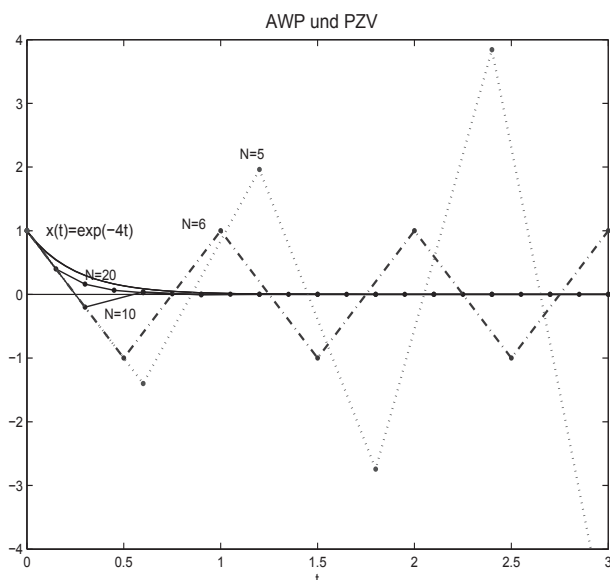


Abb. 4.23 Datei *pzv_2.ps*,
AWP $x' = \lambda x$, $x(0) = 1$,
 $0 \leq t \leq 3$, $\lambda = -4$,
mit exakter Lösung $x(t) = e^{-4t}$
und PZV $\eta_{i+1} = (1 + h\lambda)\eta_i$,
 $h = 3/N$, $N = 5, 6, 10, 20$

Für $h = 3/5 > 2/4$ oszilliert und divergiert die Folge η_i , für $2/4 > h = 3/10 > 1/4$ oszilliert und konvergiert die Folge, während für $h = 3/20 < 1/4 = 1/|\lambda|$ monotone Konvergenz vorliegt. Im Grenzfall $h = 1/2$ ist $\eta_i = (-1)^i$. Das sind Werte, die zwar nicht divergieren, aber als Näherungslösung sich eher nicht eignen.

(b) Gegeben sei das lineare homogene SysgDGL für $\mathbf{x}(t) = (x_1(t), x_2(t))^T$ als AWP

$$\mathbf{x}' = \begin{pmatrix} -\alpha & \beta \\ \beta & -\alpha \end{pmatrix} \mathbf{x} = A \mathbf{x}, \quad \mathbf{x}(t_0) = \mathbf{x}(0) = \mathbf{x}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \quad (4.52)$$

Hier ist $\mathbf{f}(t, \mathbf{x}) = A \mathbf{x}$. Die exakte Lösung $\mathbf{x}(t) = \Phi^{t, t_0} \mathbf{x}_0$ bestimmt man mittels der Eigenwerte und Eigenvektoren der symmetrischen Matrix A .

Das charakteristische Polynom ist $p_2(\lambda) = \det(A - \lambda I) = (\alpha + \lambda)^2 - \beta^2$. Die beiden Eigenwerte sind

$$\lambda_1 = -\alpha + \beta \quad \text{und} \quad \lambda_2 = -\alpha - \beta, \quad (4.53)$$

die zugehörigen orthogonalen Eigenvektoren $\mathbf{v}_1 = (1, 1)^T$ und $\mathbf{v}_2 = (1, -1)^T$.

Damit ist wegen $\Lambda = \text{diag}(\lambda_1, \lambda_2)$, $V = (\mathbf{v}_1, \mathbf{v}_2)$ als Matrix der Eigenvektoren, $AV = V\Lambda$ und

$$\begin{aligned}\mathbf{x}' &= A\mathbf{x} = V\Lambda V^{-1}\mathbf{x}, \quad \mathbf{z} = V^{-1}\mathbf{x}, \\ \mathbf{z}' &= \Lambda\mathbf{z} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{z},\end{aligned}$$

oder komponentenweise geschrieben als

$$\begin{aligned}z_1' &= \lambda_1 z_1, \\ z_2' &= \lambda_2 z_2.\end{aligned}$$

Dies sind zwei skalare lineare homogene gDGL mit der Lösung

$$\mathbf{z} = (s_1 e^{\lambda_1 t}, s_2 e^{\lambda_2 t})^T.$$

Die allgemeine Lösung ist $\mathbf{x} = V\mathbf{z}$ bzw.

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} s_1 e^{\lambda_1 t} \\ s_2 e^{\lambda_2 t} \end{pmatrix} = s_1 e^{\lambda_1 t} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + s_2 e^{\lambda_2 t} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} s_1 e^{\lambda_1 t} + s_2 e^{\lambda_2 t} \\ s_1 e^{\lambda_1 t} - s_2 e^{\lambda_2 t} \end{pmatrix}.$$

Mit der AB $\mathbf{x}(0) = (1, 0)^T$ ergeben sich die Koeffizienten $s_1 = s_2 = \frac{1}{2}$ und die Beziehung

$$\mathbf{x}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{2} \left[\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right] = \frac{1}{2}(\mathbf{v}_1 + \mathbf{v}_2).$$

Die exakte Lösung von (4.52) ist mit (4.53)

$$\mathbf{x}(t) = \frac{1}{2} \begin{pmatrix} e^{(-\alpha+\beta)t} + e^{(-\alpha-\beta)t} \\ e^{(-\alpha+\beta)t} - e^{(-\alpha-\beta)t} \end{pmatrix}. \quad (4.54)$$

Das PZV liefert zur gewählten Schrittweite h mit $\eta_0 = \mathbf{x}(0)$

$$\begin{aligned}\eta_i &= (I + hA)\eta_{i-1} = (I + hA)^i \eta_0 = (I + hA)^i \frac{1}{2}(\mathbf{v}_1 + \mathbf{v}_2) \\ &= \begin{pmatrix} 1 - h\alpha & h\beta \\ h\beta & 1 - h\alpha \end{pmatrix}^i \begin{pmatrix} 1 \\ 0 \end{pmatrix}.\end{aligned}$$

Die Matrix $I + hA$ bzw. ihre Eigenwerte $1 + h\lambda$ charakterisieren das Wachstum der Näherungslösung η_i . Deshalb wird $E(A, h) = I + hA$ als Wachstums-, Transformations- bzw. Übergangsfunktion bezeichnet. Wir wollen ihre Eigenwerte allgemein bestimmen. Es sind die Größen

$$\tilde{\lambda}_1 = 1 - h(\alpha - \beta), \quad \tilde{\lambda}_2 = 1 - h(\alpha + \beta).$$

Die zugehörigen Eigenvektoren sind wiederum $\mathbf{v}_1, \mathbf{v}_2$.

Es gilt

$$\eta_i = \frac{1}{2}(I + hA)^i (\mathbf{v}_1 + \mathbf{v}_2) = \frac{1}{2}((I + hA)^i \mathbf{v}_1 + (I + hA)^i \mathbf{v}_2) = \frac{1}{2}(\tilde{\lambda}_1^i \mathbf{v}_1 + \tilde{\lambda}_2^i \mathbf{v}_2),$$

also

$$\eta_i = \frac{1}{2} \begin{pmatrix} [1 + h(-\alpha + \beta)]^i + [1 + h(-\alpha - \beta)]^i \\ [1 + h(-\alpha + \beta)]^i - [1 + h(-\alpha - \beta)]^i \end{pmatrix}, \quad i = 0, 1, 2, \dots \quad (4.55)$$

Damit erklärt sich die Approximation von $\mathbf{x}(t_i)$ gemäß (4.54) durch die Größen η_i (4.55) wegen des Zusammenhangs von

$$\begin{aligned} e^{(-\alpha+\beta)t_i} &= e^{(-\alpha+\beta)ih} && \iff [1 + h(-\alpha + \beta)]^i \\ &= 1 + (-\alpha + \beta)ih && = 1 + (-\alpha + \beta)ih \\ &\quad + \frac{1}{2}(-\alpha + \beta)^2 i^2 h^2 + \mathcal{O}((ih)^3), && + \frac{1}{2}(-\alpha + \beta)^2 i(i-1)h^2 + \mathcal{O}((ih)^3), \\ \\ e^{(-\alpha-\beta)t_i} &= e^{(-\alpha-\beta)ih} && \iff [1 + h(-\alpha - \beta)]^i \\ &= 1 + (-\alpha - \beta)ih && = 1 + (-\alpha - \beta)ih \\ &\quad + \frac{1}{2}(-\alpha - \beta)^2 i^2 h^2 + \mathcal{O}((ih)^3), && + \frac{1}{2}(-\alpha - \beta)^2 i(i-1)h^2 + \mathcal{O}((ih)^3). \end{aligned}$$

Man hat also ein vergleichbares Wachstumsverhalten.

Seien nun $\alpha > 0$ und $\beta > 0$. Wenn zusätzlich $\alpha > \beta$ gilt, ist die exakte Lösung $\mathbf{x}(t)$ die Summe zweier abnehmender Funktionen $e^{-(\alpha-\beta)t}$ und $e^{-(\alpha+\beta)t}$, im umgekehrten Fall die Summe der zunehmenden Komponente $e^{(\beta-\alpha)t}$ und der abnehmenden $e^{-(\alpha+\beta)t}$. Diese Eigenschaft sollte in der Näherungslösung η_i durch die Wahl einer geeigneten Schrittweite h erhalten bleiben. Dabei werden die abnehmenden Bestandteile auf eine Forderung für h führen.

Fallunterscheidung

(1) $\alpha > \beta > 0$

Die Bedingungen sind

$$\begin{aligned} |1 + h(-\alpha + \beta)| &< 1, \quad \text{d.h.} \quad -1 < 1 - h(\alpha - \beta) < 1, \\ |1 + h(-\alpha - \beta)| &< 1, \quad \text{d.h.} \quad -1 < 1 - h(\alpha + \beta) < 1. \end{aligned}$$

Ausschlaggebend hierbei ist die Forderung $-1 < 1 - h(\alpha + \beta)$ bzw.

$$0 < h < \frac{2}{\alpha + \beta} = \frac{2}{|\lambda_2|}. \quad (4.56)$$

Will man auch Oszillation vermeiden und monotones Abklingverhalten erzeugen, so fordert man $0 < 1 - h(\alpha + \beta)$ bzw. als schärfere Beschränkung $0 < h < \frac{1}{\alpha + \beta}$.

(2) $\beta > \alpha > 0$

Die Bedingung ist

$$|1 + h(-\alpha - \beta)| < 1, \quad \text{d.h.} \quad -1 < 1 - h(\alpha + \beta) < 1$$

mit den Ergebnissen wie im Teil (1).

Führt man eine Testrechnung mit der Schrittweite $h = 0.1$ für $\alpha = \beta = 1$ durch, so erhält man die in Tabelle 4.1 aufgeführten numerischen Ergebnisse, welche zumindest den qualitativen Verlauf der exakten Lösung widerspiegeln.

t	numerisch, $h=0.1$		exakt $\mathbf{x}(t)$	
0.0	1	0	1	0
0.1	0.9000	0.1000	0.9094	0.0906
0.2	0.8200	0.1800	0.8352	0.1648
0.3	0.7560	0.2440	0.7744	0.2256
0.4	0.7048	0.2952	0.7247	0.2753
0.5	0.6638	0.3362	0.6839	0.3161
1.0	0.5537	0.4463	0.5677	0.4323

Tab. 4.1 Näherungslösung $\eta_i = (\eta_{1i}, \eta_{2i})$ für $h = 0.1$, $\alpha = \beta = 1$

Testrechnungen für $\alpha = \beta = 20$ sind aus Tabelle 4.2 abzulesen. Es zeigt sich, dass die Schrittweite $h = 0.1$ hier völlig unbrauchbar ist. Die Methode ist offenbar instabil. Wie lässt sich dies erklären?

t	num., $h=0.1$		numerisch, $h = 0.01$		exakt $\mathbf{x}(t) = (x_1(t), x_2(t))$	
0.0	1	0	1	0	1	0
0.1	-1	2	0.503023308800	0.496976691200	0.509157819444	0.490842180555
0.2	5	-4	0.500018280792	0.499981719207	0.500167731313	0.499832268686
0.3	-13	14	0.500000110536	0.499999889463	0.500003072106	0.499996927893
0.4	41	-40	0.500000000668	0.499999999331	0.500000056267	0.499999943732
0.5	-121	122	0.500000000004	0.499999999995	0.500000001030	0.499999998969
1.0	29525	-29524	0.500000000000	0.500000000000	0.500000000000	0.500000000000

Tab. 4.2 Näherungslösung $\eta_i = (\eta_{1i}, \eta_{2i})$ für $h = 0.1, 0.01$, $\alpha = \beta = 20$

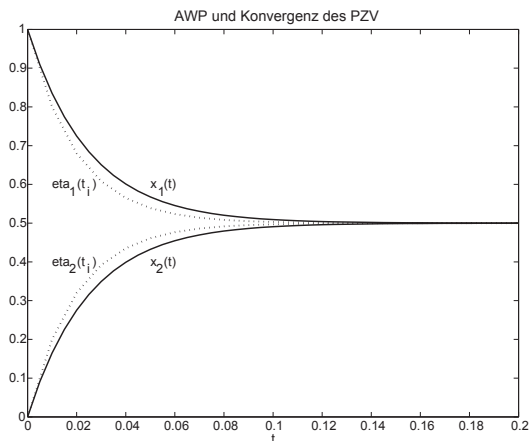


Abb. 4.24

Datei *pzv_4.ps*,
 AWP (4.52) mit der Lösung $\mathbf{x}(t)$
 und dem PZV η_i (4.55) bei
 $\alpha = \beta = 20$,
 $h = 1/N$, $N = 100$,
 in $t \in [0, 0.2]$

Der Grund für unbrauchbare Näherungslösungen sind die Schrittweite h sowie die Eigenwerte $\lambda_{1,2}$ von A , die wir schon berechnet haben.

Im Fall $\alpha = \beta$ lauten die Eigenwerte $\lambda_1 = 0$ und $\lambda_2 = -2\alpha$. Die zugehörigen Eigenvektoren sind natürlich $(1, 1)^T$ und $(1, -1)^T$.

Durch die Transformation mit den Eigenvektoren entstehen die zwei skalaren linearen homogenen gDGI

$$z'_k = \lambda_k z_k, \quad k = 1, 2.$$

Aus Teil (a) (vergl. (4.51)) bzw. (4.56) wissen wir, dass zur numerischen Lösung mit Hilfe des PZV eine Schrittweite gewählt werden muss, für die $h < \frac{2}{|\lambda_k|}$ gilt. Im Fall $\alpha = \beta = 1$ folgt nach diesem Kriterium als maximal zulässige Schrittweite $h < \frac{1}{\alpha} = 1$. Im Fall $\alpha = \beta = 20$ ist dies $h < \frac{1}{20}$.

(c) Wir kehren kurz zur gDGI $x' = \sin(1/x) - 2$ aus der Bemerkung 4.2 Teil (f) zurück und wenden das PZV darauf an.

Zunächst werfen wir noch einen Blick auf das Richtungsfeld, das erst bei feiner Auflösung und nahe $x = 0$ das ‐Zittern‐ der Linienelemente deutlich machen lässt. Dies wird mit der MATLAB-Funktion `dfield` aus Abschnitt 4.1 erzeugt.

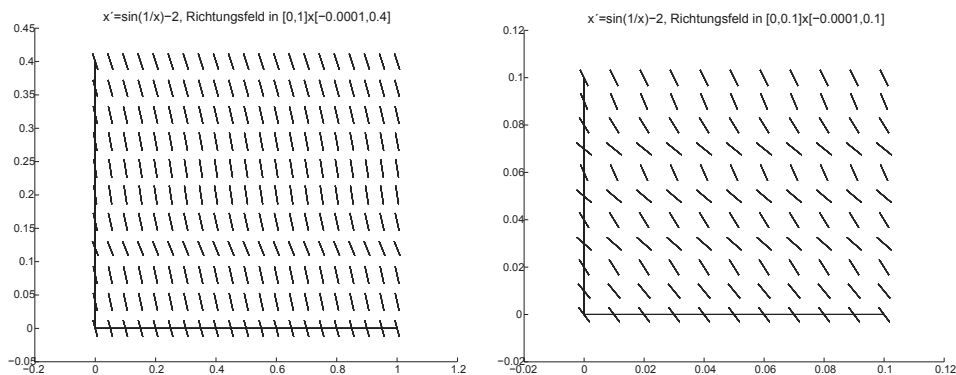


Abb. 4.25 Dateien `pzv_51.ps`, `pzv_52.ps`, Richtungsfelder zur gDGI $x' = \sin(1/x) - 2$ in den Bereichen $\Omega_1 = [0, 1] \times [-10^{-4}, 0.4]$ und $\Omega_2 = [0, 0.1] \times [-10^{-4}, 0.1]$

Nun verfolgen wir den Verlauf der Näherungslösung mit dem PZV (4.45), (4.46) zu AWP bei ausgewählter Schrittweite h . Dabei interessieren uns die Näherungswerte η_i nahe der Null bzw. der ‐Nulldurchgang‐. Da die Stelle des Nulldurchgangs keine rationale Zahl ist, für die AB $x(0) = c = 1$ beträgt dieser Wert (4.40)

$$t = T = \int_0^c \frac{dx}{2 - \sin(1/x)} = 0.767\,410\,604,$$

ist eher unwahrscheinlich, dass das PZV mit irgendeiner festen Schrittweite h genau auf den Zeitpunkt t_i trifft, wo die Näherung $\eta_i = 0$ wird und damit eine Nulldivision entsteht.

So kommt es unbeschadet, ohne Kollaps, über die Stelle hinweg und rechnet dann normal weiter. Die Verläufe der Näherungslösungen für verschiedene Schrittweiten h zeichnen wir vor dem Hintergrund eines groben Richtungsfeldes.

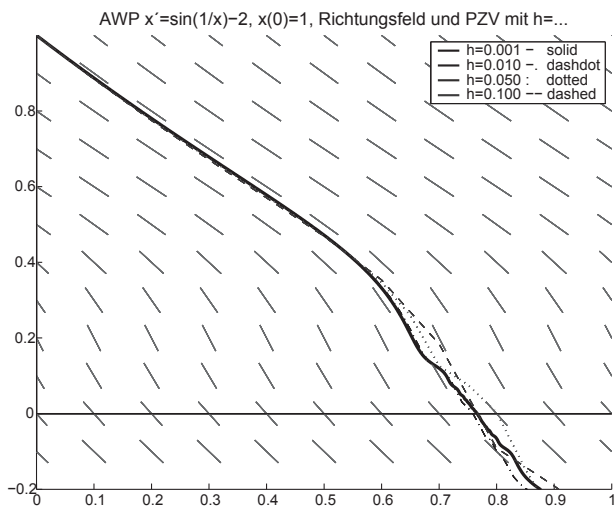


Abb. 4.26

Datei *pzv_53.ps*,
Richtungsfeld zur gDGl
 $x' = \sin(1/x) - 2$
sowie Näherungslösungen
mit dem PZV, AB $x(0) = 1$,
bei $h = 0.001, 0.01, 0.05, 0.1$

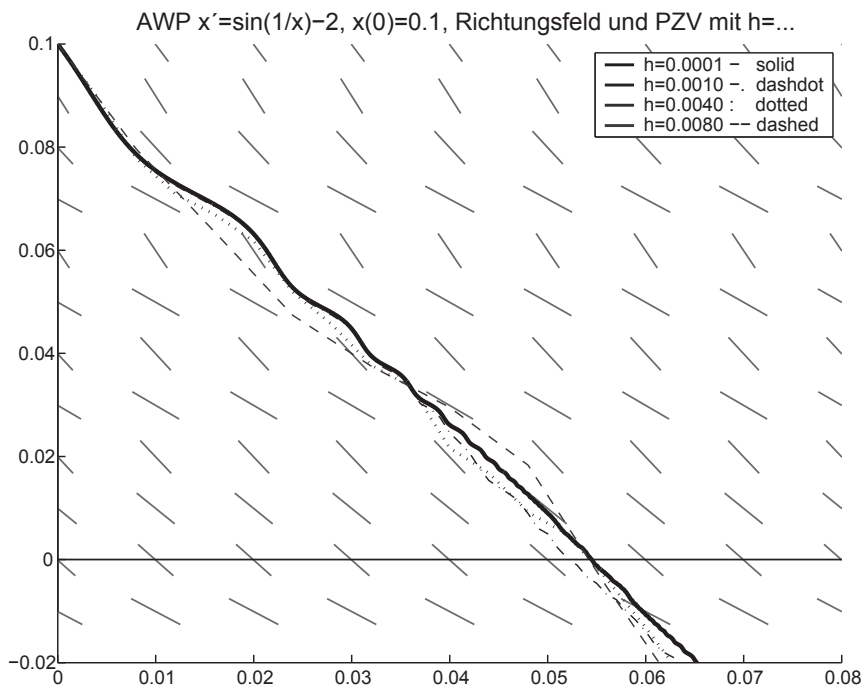


Abb. 4.27 Datei *pzv_54.ps*, Richtungsfeld zur gDGl $x' = \sin(1/x) - 2$
sowie Näherungslösungen mit dem PZV, AB $x(0) = 0.1$,
bei $h = 0.0001, 0.001, 0.004, 0.008$

Wir betrachten noch die Nulldurchgänge des AWP bei T und die Näherungswerte η_i des PZV in dieser Umgebung, wo der Vorzeichenwechsel auftritt.

(1) $x(0) = 1$, $[a, b] = [0, 1]$, $T = 0.767\,410\,604$

h	$(t_i, \eta_i)^T, i = k, k+1, \dots, k+5$					
0.00001	0.76717	0.76718	0.76719	0.76720	0.76721	0.76722
	0.000028	0.000015	0.000004	-0.000005	-0.000023	-0.000053
0.0001	0.7664	0.7665	0.7666	0.7667	0.7668	0.7669
	0.000349	0.000245	0.000102	-0.000166	-0.000374	-0.000492
0.001	0.764	0.765	0.766	0.767	0.768	0.769
	0.0048	0.0023	0.0009	-0.0021	-0.0049	-0.0070
0.01	0.73	0.74	0.75	0.76	0.77	0.78
	0.0590	0.0295	0.0152	-0.0031	-0.0308	-0.0593
0.05	0.70	0.75	0.80	0.85	0.90	0.95
	0.1263	0.0762	0.0029	-0.1469	-0.2720	-0.3464
0.1	0.5	0.6	0.7	0.8	0.9	1.0
	0.4688	0.3534	0.1840	-0.0910	-0.1910	-0.3044

(2) $x(0) = 0.1$, $[a, b] = [0, 0.1]$, $T = 0.055\,308\,494$

h	$(t_i, \eta_i)^T, i = k, k+1, \dots, k+5$					
0.000001	0.055229	0.055230	0.055231	0.055232	0.055233	0.055234
	0.0000045	0.0000035	0.0000005	-0.0000011	-0.0000021	-0.0000045
0.00001	0.05502	0.05503	0.05504	0.05505	0.05506	0.05507
	0.000038	0.000017	0.000001	-0.000020	-0.000045	-0.000057
0.0001	0.0542	0.0543	0.0544	0.0545	0.0546	0.0547
	0.000578	0.000460	0.000173	-0.000004	-0.000304	-0.000588
0.001	0.050	0.051	0.052	0.053	0.054	0.055
	0.004992	0.002289	0.000288	-0.001120	-0.003341	-0.004615
0.004	0.044	0.048	0.052	0.056	0.060	0.064
	0.016782	0.009193	0.004893	-0.003749	-0.013162	-0.023341
0.008	0.032	0.040	0.048	0.056	0.064	0.072
	0.037492	0.029488	0.018302	-0.005241	-0.027205	-0.036737

Tab. 4.3 PZV mit h , Näherungswerte η_i der Lösung $x(t)$ nahe Null

Man bemerke, dass die Nulldurchgänge des PZV teilweise in Intervallen liegen, insbesondere bei sehr kleinen Schrittweiten h , die nicht den Nulldurchgang T der exakten Lösung enthalten.

(d) Ein Rechenbeispiel ist das AWP der Riccati⁷-DGI

$$x' = t - x^2, \quad x(0) = 0, \quad t \in [0, 1], \tag{4.57}$$

das unter Verwendung der Airyschen Funktionen exakt lösbar ist.

Man löse dieses näherungsweise mit dem PZV bei $N = 5, 10, 20, \dots$ Schritten.

Die Ergebnisse η_i , $i = 0, 1, \dots, N$, und entsprechende Fehler sind in Tabellen zusammengestellt. Man bemerke, dass sich bei Schrittweithalbung ungefähr der Fehler halbiert.

t_i	η_i			$x(t_i)$
	$N = 5$ $h = 0.2$	$N = 10$ $h = 0.1$	$N = 20$ $h = 0.05$	
0.0	0	0	0	0
0.2	0	0.0100	0.0150	0.0199 8401
0.4	0.0040	0.0599	0.0697	0.0794 9206
0.6	0.1197	0.1486	0.1626	0.1762 1416
0.8	0.2368	0.2721	0.2887	0.3046 0043
1.0	0.3856	0.4228	0.4397	0.4555 4452
e_N	-0.0699	-0.0327	-0.0158	

Tab. 4.4 Näherungswerte aus dem Verlauf des PZV und Genauigkeit im Vergleich mit $x(1) = 0.4555 4452 6081$

N	h	η_N ($t_N = 1$)	Glob. Fehler $e_N(h)$	$\frac{ e_N(h) }{ e_{2N}(h/2) }$
5	0.2	0.3855 9903	-0.0699 4548	2.1362
10	0.1	0.4228 0221	-0.0327 4230	2.0690
20	0.05	0.4397 1998	-0.0158 2454	2.0345
40	0.025	0.4477 6657	-0.0077 7795	2.0172
80	0.0125	0.4516 8879	-0.0038 5573	2.0086
160	0.0062 5	0.4536 2492	-0.0019 1960	2.0043
320	0.0031 25	0.4545 8678	-0.0009 5774	2.0021
640	0.0015 625	0.4550 6617	-0.0004 7835	-
	$x(1)$	0.4555 4452		2

Tab. 4.5 Ergebnisse η_N bei $t_N = 1$ für $N = 5, 10, 20, 40, 80, 160, 320, 640$ sowie Relation der globalen Fehler $e_N = \eta_N - x(t_N)$ bei Schrittweithalbung

⁷Jacopo Francesco Riccati (1676-1754), Vincenzo Riccati (1707-1775), italienische Mathematiker, Differentialgleichungen

Das PZV ist leicht zu programmieren. In MATLAB erstellt man für die rechte Seite, das Verfahren und den Rahmen entsprechende Dateien. Dabei soll das Anliegen sein, die Programme eher gut verständlich als in ausgefeilter Version (z.B. durch speziellen Umgang mit Feldern) zu notieren.

MATLAB für den skalaren Fall

Die Implementation ist für das Beispiel 4.2 (a) $x' = \lambda x$, $x(0) = 1$, mit $\lambda = 1$ gemacht worden, woraus sich die Abbildung 4.22 ergibt.

```
%-----
% Datei f0.m
function y = f0(t,x)
    y = x;
%-----
% Datei euler1.m
function [t,x] = euler1(f,t0,t1,x0,N)
    h = (t1-t0)/N;
    t(1) = t0;
    x(1) = x0;
    for i = 1:N
        t(i+1) = t0+h*i;
        x(i+1) = x(i)+h*feval(f,t(i),x(i));
    end;
%-----
% Datei r_euler1.m
clear all
figure(1)
ta=0:0.05:3;
xa=exp(ta);
plot(ta,xa,'k-',ta,0.998*xa,'k-',ta,0.996*xa,'k-');
axis([0.0 3.0 0 20.0]);
title('AWP und Konvergenz des PZV');
xlabel('t');
text(1.7,9,'x(t)=exp(t)');
text(2.75,14.5,'N=40');
text(2.75,12.5,'N=20');
text(2.75,10.5,'N=10');
text(2.75,8.2,'N=5');
text(2.75,6.3,'N=3');
hold on
[t,x] = euler1('f0',0,3,1,3);
plot(t,x,'k-');
plot(t,x,'k. ');
[t,x] = euler1('f0',0,3,1,5);
plot(t,x,'k-');
plot(t,x,'k. ');
[t,x] = euler1('f0',0,3,1,10);
plot(t,x,'r-');
plot(t,x,'r. ');
[t,x] = euler1('f0',0,3,1,20);
plot(t,x,'b:');
plot(t,x,'b. ');
[t,x] = euler1('f0',0,3,1,40);
plot(t,x,'g:');
plot(t,x,'g. ');
print pzv_1.ps -dpsc
hold off
```

MATLAB für den vektoriellen Fall

Die Implementation ist für Beispiel 4.2 (b), Tabelle 4.2 und Abbildung 4.24.

```

%-----
% Datei f3s.m
function y = f3s(t,x)
    alpha = 20;
    beta = 20;
    y = zeros(size(x));
    y(1) = -alpha*x(1)+beta*x(2);
    y(2) = beta*x(1)-alpha*x(2);
%-----
% Datei euler.m
function [t,x] = euler(f,t0,t1,x0,N)
    h = (t1-t0)/N;
    n = size(x0,1);
    x = zeros(n,N+1);
    t(1) = t0;
    x(:,1) = x0;
    for i = 1:N
        t(i+1) = t0+h*i;
        x(:,i+1) = x(:,i)+h*feval(f,t(i),x(:,i));
    end;
%-----
% Datei r_euler.m
% Polygonzugverfahren fuer SysgDGl
% --> euler.m --> f3s.m
clear all
clc
format long

alpha = 20;
beta = 20;
t0 = 0
T = 1
x0 = [1,0]'
N = 100 % N = 10
[t,x] = euler('f3s',t0,T,x0,N);

disp('Exakte Loesung des AWP fuer 0<=t<=0.2');
tt = 0:0.005:0.2
x1 = 0.5*(exp((-alpha+beta)*tt)+exp((-alpha-beta)*tt));
x2 = 0.5*(exp((-alpha+beta)*tt)-exp((-alpha-beta)*tt));
plot(tt,x1,'k',tt,x2,'k');
axis([0.0 0.2 0 1.0]);
title('AWP und Konvergenz des PZV');
xlabel('t');
text(0.05,0.6,'x (t)');
text(0.052,0.58,'1');
text(0.05,0.4,'x (t)');
text(0.052,0.38,'2');
hold on

disp('Naehrungsloesung mit PZV fuer 0<=t<=0.2');
plot(t,x(1,:), 'b:');
text(0.01,0.6,'eta (t)');
text(0.017,0.58,'1 i');

```

```

plot(t,x(2,:), 'r:');
text(0.01,0.4, 'eta (t)');
text(0.017,0.38, '2 i');
print pzv_4.ps -dpsc
hold off

disp('Naeherungsloesung mit PZV fuer t=0(0.1)0.5, 1');
x(:,1)'
x(:,11)'
x(:,21)'
x(:,31)'
x(:,41)'
x(:,51)'
x(:,101)'

disp('Exakte Loesung fuer t=0(0.1)0.5, 1');
for t = 0:0.1:0.5
    x1 = 0.5*(exp((-alpha+beta)*t)+exp((-alpha-beta)*t))
    x2 = 0.5*(exp((-alpha+beta)*t)-exp((-alpha-beta)*t))
end
t = 1.0
x1 = 0.5*(exp((-alpha+beta)*t)+exp((-alpha-beta)*t))
x2 = 0.5*(exp((-alpha+beta)*t)-exp((-alpha-beta)*t))
pause

```

Die beiden Beispiele 4.2 (a), (b) sollen die folgende allgemeine Merkregel motivieren.

Bemerkung 4.3

A sei eine $(n \times n)$ -Matrix mit den Eigenwerten $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R}$.

λ_{max} sei der betragsgrößte negative Eigenwert. Zur numerischen Lösung des SysgDGL

$$\mathbf{x}' = A\mathbf{x}$$

mit Hilfe des PZV muss die Schrittweite

$$h < \frac{1}{|\lambda_{max}|} \quad (4.58)$$

gewählt werden.

Lineare SysgDGL, bei denen (komplexe) Eigenwerte mit sehr verschiedenen negativen Realteilen auftreten, heißen auch steife Differentialgleichungssysteme. Hier verursacht der betragsgrößte negative Realteil eine obere Schranke für die Wahl der Schrittweite.

Übung 4.2

(a) Das Beispiel 4.1 (b) soll für sehr kleine Werte von b und D mit Hilfe des PZV gelöst werden. Geben Sie eine Oberschranke für die Schrittweite an.

(b) Das Beispiel 4.1 (d) soll mittels des PZV gelöst werden. Geben Sie eine obere Schranke für die Schrittweitenwahl an.

(c) Wie verhält sich das PZV bei großen Schrittweiten im Fall positiver Eigenwerte?

4.3 Erste Fehlerbetrachtungen

Gegeben sei das skalare AWP

$$x' = f(t, x), \quad x(t_0) = c, \quad t \in [t_0, T]. \quad (4.59)$$

Die exakte Lösung kann mit Hilfe des Flusses in der Form

$$x(t) = \Phi^{t, t_0} c. \quad (4.60)$$

beschrieben werden. Zu gegebener Schrittweite $h > 0$ und den zugehörigen diskreten Zeiten oder Stützstellen $t_i = t_0 + i h$ liefert ein numerisches Verfahren, z.B. das PZV, Näherungen η_i zu den gesuchten Werten $x(t_i)$. Für den hierbei entstehenden Approximationsfehler können zwei Arten betrachtet werden.

Definition 4.4 Fehlerarten

(a) Ausgehend von einem Näherungswert η_i für $\Phi^{t_i, t_0} c$ wird η_{i+1} als Näherungswert der Trajektorie durch (t_i, η_i) an der nächsten Stelle t_{i+1} konstruiert, also

$$\eta_{i+1} \approx x(t_{i+1}; t_i, \eta_i) = \Phi^{t_{i+1}, t_i} \eta_i.$$

Wir notieren hier $x(t_{i+1}; t_i, \eta_i)$ zur Unterscheidung von $x(t_{i+1}) = \Phi^{t_{i+1}, t_0} x_0$.

Die skalierte Abweichung

$$\tau_{i+1} = \tau(t_i, \eta_i, h) = \frac{1}{h} (\Phi^{t_i+h, t_i} \eta_i - \eta_{i+1}) \quad (4.61)$$

nennt man den lokalen Diskretisierungsfehler. Zu seiner Abschätzung bedient man sich üblicherweise der Taylor-Entwicklung des Flusses $\Phi^{t_i+h, t_i} \eta_i$ um den Punkt (t_i, η_i) .

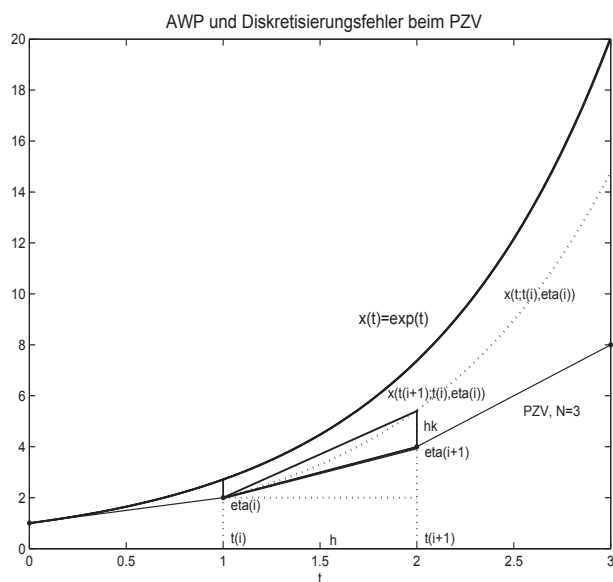


Abb. 4.28 Datei *pzv_3.ps*,
Lokaler Diskretisierungsfehler
 $\tau(t_i, \eta_i, h)$
zum PZV $\eta_{i+1} = (1 + h\lambda)\eta_i$,
 $h = 3/N$, $N = 3$,
für das AWP $x' = \lambda x$,
 $x(0) = 1$, $0 \leq t \leq 3$, $\lambda = 1$,
mit exakter Lösung $x(t) = e^t$

Welche geometrische Interpretation kann man dem lokalen Diskretisierungsfehler (4.61) $\tau(t_i, \eta_i, h)$ geben?

Gemäß Abb. 4.28 findet man $\tau(t_i, \eta_i, h)$ als Differenz der Anstiege der Sekanten zwischen $(\eta_i, x(t_{i+1}; t_i, \eta_i))$ (exakte Lösung) und (η_i, η_{i+1}) (Näherungslösung), denn es gilt

$$\begin{aligned} \tau_{i+1} = \tau(t_i, \eta_i, h) &= \frac{1}{h} (\Phi^{t_{i+1}, t_i} \eta_i - \eta_{i+1}) = \frac{1}{h} (\Phi^{t_{i+1}, t_i} \eta_i - \eta_i + \eta_i - \eta_{i+1}) \\ &= \frac{1}{h} (x(t_{i+1}; t_i, \eta_i) - \eta_i) - \frac{1}{h} (\eta_{i+1} - \eta_i). \end{aligned} \quad (4.62)$$

Der Wert $hk = h\tau(t_i, \eta_i, h)$ ist demnach ein Maß für die lokale Abweichung der Näherungslösung von der exakten, durch (t_i, η_i) verlaufenden, Lösung der gDGL. Das Diskretisierungsverfahren sollte diese Abweichung möglichst "klein halten".

(b) Der Startpunkt η_i zur Berechnung von η_{i+1} ist (außer für $i = 0$) ebenfalls schon mit einem Fehler behaftet, sodass η_{i+1} mit $\Phi^{t_{i+1}, t_i} \eta_i$ anstelle von $\Phi^{t_{i+1}, t_i} x(t_i)$ die "falsche" Trajektorie approximiert. Mit dem lokalen Diskretisierungsfehler als "Keimzelle" addieren sich die in jedem Schritt neu entstehenden Anteile zum globalen Diskretisierungsfehler

$$e_i = e(t_i, h) = \eta_i - x(t_i; t_0, x_0) = \eta_i - \Phi^{t_i, t_0} x_0 \quad (4.63)$$

an der Stelle t_i auf, sodass unter Umständen die Teiltrajektorien von der gesuchten Lösungstrajektorie immer weiter "wegdriften". Diesen globalen Fehler versucht man mit sogenannten Stabilitätsuntersuchungen in den Griff zu bekommen.

Wir untersuchen den lokalen Diskretisierungsfehler für das PZV.

Hierzu nehmen wir an, dass $f(t, x)$ hinreichend glatt ist, hier mindestens zweimal stetig differenzierbar bzgl. t und x ist.

Ist (t_i, η_i) gegeben, so bezeichne

$$z(t) = \Phi^{t, t_i} \eta_i \quad (4.64)$$

die Trajektorie, welche durch (t_i, η_i) verläuft. Damit ist $z(\cdot)$ Lösung des AWP

$$z' = f(t, z), \quad z(t_i) = \eta_i.$$

Wir werten nun die Taylor-Entwicklung

$$z(t) = z(t_i) + (t - t_i) z'(t_i) + \frac{(t - t_i)^2}{2!} z''(t_i) + \dots$$

von $z(t)$ um den Punkt t_i aus. Zunächst gilt

$$z'(t_i) = f(t_i, z(t_i)) = f(t_i, \eta_i).$$

Ferner liefert die Kettenregel die Beziehung

$$\begin{aligned} z''(t_i) &= \left. \frac{d}{dt} z'(t) \right|_{t=t_i} = \left. \frac{d}{dt} f(t, z(t)) \right|_{t=t_i} \\ &= f_t(t_i, z(t_i)) + z'(t_i) f_x(t_i, z(t_i)) = f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i). \end{aligned}$$

Hierbei bezeichnen f_t und f_x die partiellen Ableitungen von f bzgl. der ersten und zweiten Variable. Aus

$$z(t_{i+1}) = \eta_i + h f(t_i, \eta_i) + \frac{h^2}{2} [f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i)] + \mathcal{O}(h^3) \quad (4.65)$$

und

$$\eta_{i+1} = \eta_i + h f(t_i, \eta_i) \quad (4.66)$$

ergibt sich der lokale Diskretisierungsfehler

$$\begin{aligned} \tau_{i+1} = \tau(t_i, \eta_i, h) &= \frac{1}{h} [z(t_{i+1}) - \eta_{i+1}] \\ &= \frac{h}{2} [f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i)] + \mathcal{O}(h^2) \\ &\doteq \frac{h}{2} [f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i)] \\ &= \mathcal{O}(h). \end{aligned} \quad (4.67)$$

Diese Fehlerordnung möchte man auch für den globalen Fehler $e(t_i, h)$ erhalten. Das Symbol \doteq bedeutet, dass nur der Term der führenden Ordnung in h berücksichtigt wurde.

Man erhält den lokalen Diskretisierungsfehler auch dadurch, dass man in das Diskretisierungsverfahren anstelle von η_j die exakten Lösungswerte $x(t_j)$ einsetzt. Das bedeutet

$$\begin{aligned} \eta_{i+1} &= \eta_i + h f(t_i, \eta_i), \\ 0 &= \frac{1}{h} [\eta_{i+1} - \eta_i] - f(t_i, \eta_i), \\ \tau_{i+1} &= \frac{1}{h} [x(t_{i+1}) - x(t_i)] - f(t_i, x(t_i)) \\ &= \frac{1}{h} [x(t_i) + hx'(t_i) + \frac{h^2}{2} x''(t_i) + \mathcal{O}(h^3) - x(t_i)] - f(t_i, x(t_i)) \\ &= x'(t_i) - f(t_i, x(t_i)) + \frac{h}{2} x''(t_i) + \mathcal{O}(h^2) \\ &= \frac{h}{2} x''(t_i) + \mathcal{O}(h^2) \\ &= \mathcal{O}(h). \end{aligned}$$

Beispiel 4.3

Im Fall der gDGL $x'(t) = tx(t)$ hat man $f(t, x) = tx$, $f_t(t, x) = x$ und $f_x(t, x) = t$. Der lokale Diskretisierungsfehler des PZV ist daher gleich

$$\tau(t_i, \eta_i, h) \doteq \frac{h}{2} (\eta_i + t_i^2 \eta_i) = \frac{h}{2} \eta_i (1 + t_i^2).$$

Kapitel 5

Explizite Einschrittverfahren

5.1 Das Konzept der Einschrittverfahren

Der lokale Diskretisierungsfehler des PZV ist von der Ordnung $\mathcal{O}(h)$.

Mit dem Verfahren werden Näherungslösungen von gDGI konstruiert, indem in jedem kleinen Teilintervall eine lineare Approximation konstruiert wird, wobei als Steigung die Steigung am linken Randpunkt gewählt wird.

Ziel ist es nun, Alternativen zum PZV zu entwickeln, welche günstiger sind bzgl. des lokalen Diskretisierungsfehlers. Die folgenden Beispiele versuchen, einen Wert für die Steigung zu finden, welcher dem Verlauf im gesamten Teilintervall besser angepasst ist. Wie vorher sei $t_i = t_0 + i h$ mit einer vorgegebenen kleinen Schrittweite h . Zur Berechnung des lokalen Diskretisierungsfehlers benötigen wir die Taylor-Entwicklung von $f(\cdot, \cdot)$ um den Punkt (t_i, η_i) .

Für

$$x(t) - \eta_i = (t - t_i) x'(\xi) = \mathcal{O}(h) \quad \text{und} \quad \mathcal{O}(t - t_i) = \mathcal{O}(h)$$

ist diese gegeben durch

$$\begin{aligned} f(t, x) &= f(t_i, \eta_i) + (t - t_i) f_t(t_i, \eta_i) + (x - \eta_i) f_x(t_i, \eta_i) \\ &\quad + \frac{1}{2} (t - t_i)^2 f_{tt}(t_i, \eta_i) + (t - t_i)(x - \eta_i) f_{tx}(t_i, \eta_i) \\ &\quad + \frac{1}{2} (x - \eta_i)^2 f_{xx}(t_i, \eta_i) + \mathcal{O}(h^3) \\ &= f(t_i, \eta_i) + (t - t_i) f_t(t_i, \eta_i) + (x - \eta_i) f_x(t_i, \eta_i) + \mathcal{O}(h^2). \end{aligned} \tag{5.1}$$

Beispiel 5.1

(a) **Modifiziertes Polygonzugverfahren** (MPZV) von C. Runge (1895)

Dieses wird auch modifiziertes Euler-Verfahren, Collatz-Verfahren oder Tangententrapezregel genannt. Hier wird als Steigung im Intervall $[t_i, t_{i+1}]$ der Funktionswert von $f(t, x)$ im Punkt $(t_i + \frac{h}{2}, \eta_i + \frac{h}{2} f(t_i, \eta_i))$ angenommen. Welche geometrisch anschauliche Bedeutung hat dieser Punkt?

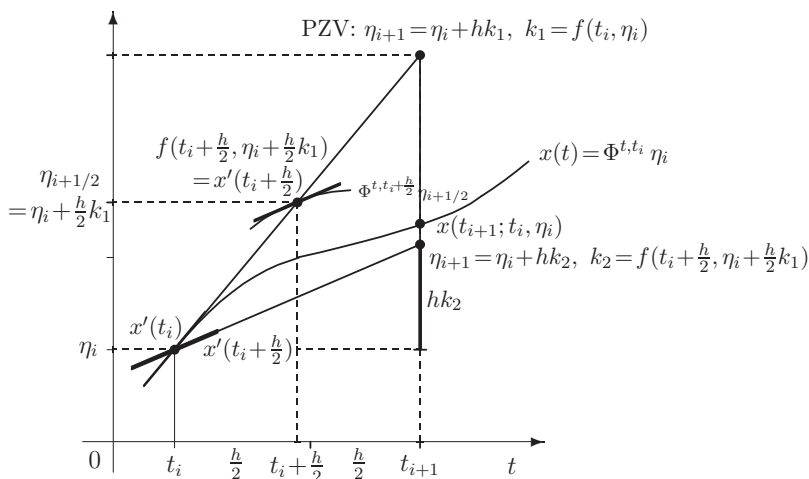


Abb. 5.1 Datei *mpzv1.pic*, Wahl des Anstiegs von einer Trajektorie “an der Mitte des Intervalls“ und Integrationschritt beim MPZV

Der Schritt ist gegeben durch die Vorschrift

$$\eta_{i+1} = \eta_i + h f\left(t_i + \frac{h}{2}, \eta_i + \frac{h}{2} f(t_i, \eta_i)\right) \quad (5.2)$$

oder in der Form mit den Anstiegen (Steigungswerte) k_1, k_2 bzw. Zuwächsen (Korrekturen) hk_1, hk_2

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f\left(t_i + \frac{h}{2}, \eta_i + \frac{h}{2} k_1\right), \\ \eta_{i+1} &= \eta_i + hk, \quad k = k_2. \end{aligned} \quad (5.3)$$

Aus (5.2) und der Taylor-Entwicklung (5.1) folgt

$$\begin{aligned} \eta_{i+1} &= \eta_i + h\left[f(t_i, \eta_i) + \frac{h}{2} f_t(t_i, \eta_i) + \frac{h}{2} f(t_i, \eta_i) f_x(t_i, \eta_i) + \mathcal{O}(h^2)\right] \\ &= \eta_i + hf(t_i, \eta_i) + \frac{h^2}{2} [f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i)] + \mathcal{O}(h^3). \end{aligned} \quad (5.4)$$

Ein Vergleich mit der Taylor-Entwicklung (4.65) für $z(t) = \Phi^{t, t_i} \eta_i$ ergibt als Ordnung für den lokalen Diskretisierungsfehler

$$\tau_{i+1} = \tau(t_i, \eta_i, h) = \frac{1}{h} (z(t_{i+1}) - \eta_{i+1}) = \mathcal{O}(h^2). \quad (5.5)$$

Der lokale Diskretisierungsfehler ist damit um eine Ordnung besser als im Fall des PZV. Allerdings sind pro Schritt zwei Funktionsauswertungen von $f(\cdot, \cdot)$ erforderlich, also zwei Zuwächse hk_1 und hk_2 zu berechnen.

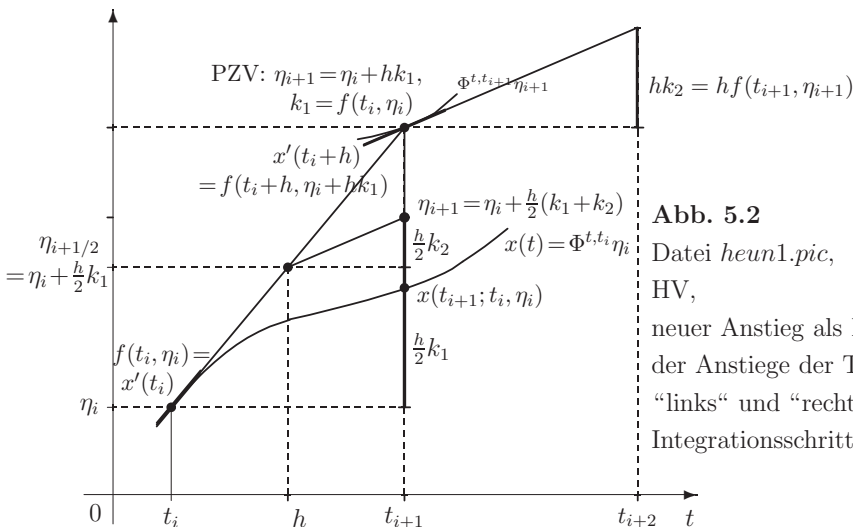
Die MATLAB-Funktionen dazu mit $t \in [t_0, t_1]$, $h = (t_1 - t_0)/N$ und $x(t_0) = x_0$ bzw. $x(t_0) = \mathbf{x}_0$ sind für gDGL bzw. SysgDGL wie folgt.

```
% Datei collatz1.m
function [t,x] = collatz1(f,t0,t1,x0,N)
    h = (t1-t0)/N;
    t(1) = t0; x(1) = x0;
    for i = 1:N
        t(i+1) = t0+i*h;
        k1 = feval(f,t(i),x(i));
        k2 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k1);
        x(i+1) = x(i)+h*k2;
    end;

% Datei collatz.m
function [t,x] = collatz(f,t0,t1,x0,N)
    h = (t1-t0)/N;
    n = size(x0,1);
    x = zeros(n,N+1);
    t(1) = t0;
    x(:,1) = x0;
    for i = 1:N
        t(i+1) = t0+i*h;
        k1 = feval(f,t(i),x(:,i));
        k2 = feval(f,t(i)+h,x(:,i)+h*k1);
        x(:,i+1) = x(:,i)+0.5*h*(k1+k2);
    end;
```

(b) **Heun-Verfahren (HV)** von K. Heun¹ (1900)

Es stellt ebenfalls ein verbessertes PZV (VPZV) dar und heißt auch Sehnentrapezregel. Hier wird als Steigung der Mittelwert der Funktion $f(\cdot, \cdot)$ in den Punkten (t_i, η_i) und $(t_i + h, \eta_i + hf(t_i, \eta_i))$ gewählt.



¹Karl Ludwig Wilhelm Max Heun (1859-1929), deutscher Mathematiker, arbeitete an mechanischen Problemen wie d'Alembertsches Prinzip, Eulersche Gleichungen, Lagrange-Formalismus

Die Berechnungsvorschrift lautet also

$$\eta_{i+1} = \eta_i + \frac{h}{2}[f(t_i, \eta_i) + f(t_i + h, \eta_i + hf(t_i, \eta_i))] \quad (5.6)$$

bzw. in der Form mit den Anstiegen k_1, k_2

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f(t_i + h, \eta_i + hk_1), \\ \eta_{i+1} &= \eta_i + hk, \quad k = \frac{1}{2}(k_1 + k_2). \end{aligned} \quad (5.7)$$

Notiert man das PZV als $\bar{\eta}_{i+1} = \eta_i + hk_1$, so ergibt sich für das HV die Formel

$$\eta_{i+1} = \eta_i + hk_1 + \frac{h}{2}(k_2 - k_1) = \bar{\eta}_{i+1} + \frac{h}{2}(k_2 - k_1),$$

sodass die Verbesserung (Korrekturterm) $\frac{h}{2}(k_2 - k_1)$ zum PZV sichtbar wird.

Die Taylor-Entwicklung (5.1) liefert für (5.7)

$$\begin{aligned} \eta_{i+1} &= \eta_i + \frac{h}{2}f(t_i, \eta_i) + \frac{h}{2}[f(t_i, \eta_i) + hf_t(t_i, \eta_i) + hf(t_i, \eta_i)f_x(t_i, \eta_i) + \mathcal{O}(h^2)] \\ &= \eta_i + hf(t_i, \eta_i) + \frac{h^2}{2}[f_t(t_i, \eta_i) + f(t_i, \eta_i)f_x(t_i, \eta_i)] + \mathcal{O}(h^3). \end{aligned}$$

Wie im Teil (a) ergibt sich damit ein lokaler Diskretisierungsfehler der Ordnung $\mathcal{O}(h^2)$.

Dazu folgt die MATLAB-Funktion mit $t \in [t_0, t_1]$, $x(t_0) = x_0$ und $h = (t_1 - t_0)/N$.

```
% Datei heun1.m
function [t,x] = heun1(f,t0,t1,x0,N)
    h = (t1-t0)/N;
    t(1) = t0;
    x(1) = x0;
    for i = 1:N
        t(i+1) = t0+h*i;
        k1 = feval(f,t(i),x(i));
        k2 = feval(f,t(i)+h,x(i)+h*k1);
        x(i+1) = x(i)+0.5*h*(k1+k2);
    end;
```

Wie beim MPZV (5.3) ist analog die vektorielle Version zu notieren.

Eine allgemeine Klasse von Verfahren für AWP lässt sich wie folgt formulieren.

Definition 5.1

(a) Ein explizites ESV für die gDGL $x' = f(t, x)$ bzw. für das SysgDGL $\mathbf{x}' = \mathbf{f}(t, \mathbf{x})$ ist eine Formel der Gestalt

$$\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h), \quad i = 0, 1, \dots, \quad (5.8)$$

mit einer vorgegebenen, i. Allg. von $f(\cdot, \cdot)$ bzw. von $\mathbf{f}(\cdot, \cdot)$ abhängigen Funktion

$$\Psi = \Psi(t_i, \eta_i, h) = \Psi(t_i, \eta_i; h, \mathbf{f}), \quad (5.9)$$

die auch Verfahrens- oder Inkrementfunktion genannt wird.

(b) Der lokale Diskretisierungsfehler (Approximationsfehler) für das ESV ist definiert durch

$$\tau(t_i, \eta_i, h) = \frac{1}{h}(\Phi^{t_i+h, t_i} \eta_i - \eta_{i+1}) = \frac{1}{h}(\Phi^{t_i+h, t_i} \eta_i - \eta_i) - \Psi(t_i, \eta_i, h) \quad (5.10)$$

(man vergleiche dies mit der Definition 4.4 und Formel (4.62)).

(c) Das ESV heißt konsistent (verträglich), wenn

$$\lim_{h \rightarrow 0} \tau(t_i, \eta_i, h) = 0. \quad (5.11)$$

gilt. Das ist nach Gleichung (5.8) und der Definition von $\Phi^{t_i, t_i} \eta_i$ genau für

$$\lim_{h \rightarrow 0} \Psi(t_i, \eta_i, h) = f(t_i, \eta_i). \quad (5.12)$$

der Fall.

(d) Das ESV ist ein Verfahren der Ordnung $p > 0$, falls

$$\tau(t_i, \eta_i, h) = \mathcal{O}(h^p) \text{ bzw. } |\tau(t_i, \eta_i, h)| \leq C h^p, \quad C \geq 0. \quad (5.13)$$

Die Größe p bezeichnet man auch als Konsistenzordnung.

Nach dieser Definition ist das PZV ein ESV der Ordnung 1. Das MPZV und das HV sind ESV der Ordnung 2.

Weitere ESV können z.B. mit Hilfe des Ansatzes bzw. der Verfahrensfunktion

$$\Psi(t_i, \eta_i, h) = b_1 f(t_i, \eta_i) + b_2 f(t_i + ch, \eta_i + ah f(t_i, \eta_i)) \quad (5.14)$$

konstruiert werden, wobei die Koeffizienten b_1 , b_2 , a und c frei gewählt werden können. Alle oben aufgeführten Beispiele genügen diesem Ansatz. Für das PZV ist $b_1 = 1$ und $b_2 = 0$, für das MPZV ist $b_1 = 0$, $b_2 = 1$ sowie $a = c = \frac{1}{2}$, für das HV gilt $b_1 = b_2 = \frac{1}{2}$ und $a = c = 1$.

Es ist nicht zufällig, dass $\sum_i b_i = 1$ und hier auch $a = c$ gelten.

Damit hat die Verfahrensfunktion bei Einbeziehung der Zuwächse $hk_1 = hf(t_i, \eta_i)$ und $hk_2 = hf(t_i + ch, \eta_i + ahk_1)$ die Gestalt

$$\Psi(t_i, \eta_i, h) = k = b_1 k_1 + b_2 k_2.$$

Die Koeffizienten bzw. Parameter kann man in einem Schema darstellen.

Seien $\mathbf{c} = (c_1, c_2)^T = (0, c)^T$ und $\mathbf{b} = (b_1, b_2)^T$ Spaltenvektoren sowie $A = (a_{ij})$ eine (2×2) -Matrix mit $a_{21} = a$ und sonst Nullen, dann erhält man die Tableaus

$$\frac{\mathbf{c}}{1} \mid \frac{A}{\mathbf{b}^T} \quad \text{bzw.} \quad \frac{c_1 = 0}{c_2} \mid \frac{a_{21}}{b_1 \quad b_2}$$

Geeignete Koeffizienten ergeben sich aus dem folgenden Satz.

Satz 5.1 *Ein ESV mit dem Ansatz (5.14) ist genau dann von (mindestens) der Konsistenzordnung 2, wenn*

$$b_1 + b_2 = 1, \quad b_2 a = \frac{1}{2} \quad \text{und} \quad b_2 c = \frac{1}{2}. \quad (5.15)$$

Eine höhere als die Ordnung 2 ist mit diesem Ansatz i. Allg. nicht zu erreichen.

Beweis. Mit dem Ansatz (5.14) hat Ψ um (t_i, η_i) die Taylor-Entwicklung

$$\Psi(t_i, \eta_i, h) = (b_1 + b_2)f(t_i, \eta_i) + b_2 h [c f_t(t_i, \eta_i) + a f(t_i, \eta_i) f_x(t_i, \eta_i)] + \mathcal{O}(h^2). \quad (5.16)$$

Die Bedingungen (5.15) folgen durch Einsetzen dieser Entwicklung in das ESV (5.8) und durch Vergleich mit der Taylor-Entwicklung (4.65) für $z(t_{i+1}) = \Phi^{t_{i+1}, t_i} \eta_i$, also aus

$$\begin{aligned} \tau(t_i, \eta_i, h) &= \frac{1}{h}(\Phi^{t_i+h, t_i} \eta_i - \eta_{i+1}) = \frac{1}{h}(z(t_{i+1}) - \eta_{i+1}) \\ &= \frac{1}{h}\{\eta_i + h f(t_i, \eta_i) + \frac{h^2}{2}[f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i)] + \mathcal{O}(h^3) \\ &\quad - \eta_i - h[(b_1 + b_2)f(t_i, \eta_i) + b_2 h(c f_t(t_i, \eta_i) + a f(t_i, \eta_i) f_x(t_i, \eta_i)) + \mathcal{O}(h^2)]\} \\ &= (1 - b_1 - b_2)f(t_i, \eta_i) \\ &\quad + h(\frac{1}{2} - b_2 c) f_t(t_i, \eta_i) + h(\frac{1}{2} - b_2 a) f(t_i, \eta_i) f_x(t_i, \eta_i) + \mathcal{O}(h^2). \end{aligned}$$

□

Mit $b = b_2 \in (0, 1]$ als freien Parameter folgen die anderen Größen $b_1 = 1 - b$, $a = c = \frac{1}{2b}$ und die erwähnten Sonderfälle $b = 1$ (MPZV) und $b = \frac{1}{2}$ (HV). $b = 0$ führt zum PZV.

5.2 Runge-Kutta-Verfahren

Ein Integrationsschritt für ein explizites ESV mit dem Ansatz (5.14) kann in die folgenden Rechenschritte zerlegt werden.

Algorithmus 5.1 Integrationsschritt für ein explizites ESV mit dem Ansatz (5.14)

Gegeben sei (t_i, η_i) .

S1 Man berechne $k_1 = f(t_i, \eta_i)$.

S2 Man berechne $k_2 = f(t_i + ch, \eta_i + ahk_1)$.

S3 Man setze $\Psi(t_i, \eta_i, h) = b_1 k_1 + b_2 k_2$ bzw. $k = b_1 k_1 + b_2 k_2$.

S4 Man berechne (t_{i+1}, η_{i+1}) mit $t_{i+1} = t_i + h$ und

$$\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h) \quad \text{bzw.} \quad \eta_{i+1} = \eta_i + hk.$$

Verfahren dieser Art werden wegen der Berechnung von zwei Anstiegen k_i als 2-stufige Runge²-Kutta³-Verfahren (RKV) bezeichnet.

Dies stellt den Spezialfall eines s -stufigen **RKV** dar. Der Ansatz (5.14) soll nun verallgemeinert und analysiert werden, um eine höhere Ordnung zu erzielen.

Definition 5.2 Ein 3-stufiges RKV ist ein explizites ESV mit dem Ansatz

$$\begin{aligned} k_1 &= f(t_i + c_1 h, \eta_i), \\ k_2 &= f(t_i + c_2 h, \eta_i + h a_{21} k_1), \\ k_3 &= f(t_i + c_3 h, \eta_i + h (a_{31} k_1 + a_{32} k_2)), \\ \Psi(t_i, \eta_i, h) &= k = b_1 k_1 + b_2 k_2 + b_3 k_3, \end{aligned} \tag{5.17}$$

mit geeigneten Koeffizienten a_{ij} , b_i , c_i .

Beispiel 5.2

(1) Für die Koeffizienten

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}, \quad A = (a_{ij}) = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & \frac{2}{3} & 0 \end{pmatrix}$$

erhält man das Parameterschema

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^T \end{array} = \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & \frac{1}{3} & \frac{2}{3} & \\ \hline 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{array}. \tag{5.18}$$

Damit ergeben sich die Anstiege

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hf(t_i, \eta_i)), \\ k_3 &= f(t_i + h, \eta_i + h(\frac{1}{3}f(t_i, \eta_i) + \frac{2}{3}f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hf(t_i, \eta_i)))), \\ \Psi(t_i, \eta_i, h) &= \frac{1}{3}(k_1 + k_2 + k_3), \end{aligned}$$

²Carle David Tolmé Runge (1856-1927), deutscher Mathematiker, Arbeitsgebiete waren Differentialgeometrie, Lösung von algebraischen und gDGL

³Martin Wilhelm Kutta (1867-1944), deutscher Mathematiker, arbeitete auf dem Gebiet der Lösung von gDGL

bzw. in Kurzform

$$\begin{aligned}
 k_1 &= f(t_i, \eta_i), \\
 k_2 &= f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hk_1), \\
 k_3 &= f(t_i + h, \eta_i + h(\frac{1}{3}k_1 + \frac{2}{3}k_2)), \\
 \Psi(t_i, \eta_i, h) &= k = \frac{1}{3}(k_1 + k_2 + k_3).
 \end{aligned} \tag{5.19}$$

Wir verwenden dieses Verfahren nun zur Lösung der autonomen gDGL

$$x' = f(x).$$

In diesem Fall lauten die Gleichungen zur Berechnung der Anstiege

$$\begin{aligned}
 k_1 &= f(\eta_i), \\
 k_2 &= f(\eta_i + \frac{1}{2}hf(\eta_i)), \\
 k_3 &= f(\eta_i + h(\frac{1}{3}f(\eta_i) + \frac{2}{3}f(\eta_i + \frac{1}{2}hf(\eta_i)))).
 \end{aligned}$$

Zur Abschätzung des lokalen Diskretisierungsfehlers betrachten wir die Taylor-Entwicklungen der k_i . Wir schreiben abkürzend f_i anstelle von $f(\eta_i)$, sowie f'_i für die Ableitung von $f(x)$ an der Stelle η_i .

$$\begin{aligned}
 k_1 &= f_i, \\
 k_2 &= f_i + \frac{1}{2}hf_i f'_i + \mathcal{O}(h^2), \\
 k_3 &= f_i + hf_i f'_i + \mathcal{O}(h^2).
 \end{aligned}$$

Hieraus ergibt sich Ψ durch

$$\Psi(t_i, \eta_i, h) = \frac{1}{3}(k_1 + k_2 + k_3) = f_i + \frac{1}{2}hf_i f'_i + \mathcal{O}(h^2).$$

Vergleichen wir dies mit der Taylor-Entwicklung von $z(t) = \Phi^{t, t_i} \eta_i$ (vergl. (4.65))

$$z(t_{i+1}) = \eta_i + hf_i + \frac{1}{2}h^2 f_i f'_i + \mathcal{O}(h^3),$$

so folgt für den lokalen Diskretisierungsfehler

$$\tau(t_i, \eta_i, h) = \mathcal{O}(h^2).$$

Damit hat das Verfahren für autonome gDGL mindestens die Ordnung 2.

Die Genauigkeitsordnung 3 wird aber nicht erreicht.

(2) Mit den Feldern

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{6} \\ \frac{3}{10} \\ \frac{8}{15} \end{pmatrix}, \quad A = (a_{ij}) = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 \\ -\frac{3}{16} & \frac{15}{16} & 0 \end{pmatrix}$$

erhält man das Parameterschema

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^T \end{array} = \begin{array}{c|cc} 0 & & \\ c_2 & \frac{1}{3} & \\ c_3 & -\frac{3}{16} & \frac{15}{16} \\ \hline 1 & \frac{1}{6} & \frac{3}{10} & \frac{8}{15} \end{array}. \quad (5.20)$$

Damit ergeben sich die Anstiege

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f(t_i + c_2 h, \eta_i + \frac{1}{3} h f(t_i, \eta_i)), \\ k_3 &= f(t_i + c_3 h, \eta_i + h(-\frac{3}{16} f(t_i, \eta_i) + \frac{15}{16} f(t_i + c_2 h, \eta_i + \frac{1}{3} h f(t_i, \eta_i))), \\ \Psi(t_i, \eta_i, h) &= \frac{1}{30}(5k_1 + 9k_2 + 16k_3), \end{aligned}$$

bzw. in Kurzform

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f(t_i + c_2 h, \eta_i + \frac{1}{3} h k_1), \\ k_3 &= f(t_i + c_3 h, \eta_i + h[-\frac{3}{16} k_1 + \frac{15}{16} k_2]), \\ \Psi(t_i, \eta_i, h) &= k = \frac{1}{30}(5k_1 + 9k_2 + 16k_3). \end{aligned} \quad (5.21)$$

Da die Koeffizienten c_i bisher nicht definiert sind, benutzen wir dieses Verfahren zur Lösung der autonomen gDGL $x' = f(x)$, wo formal die Koeffizienten c_i ja nicht benötigt werden. In diesem Fall lauten die Gleichungen zur Berechnung der Anstiege

$$\begin{aligned} k_1 &= f(\eta_i), \\ k_2 &= f(\eta_i + \frac{1}{3} h f(\eta_i)), \\ k_3 &= f(\eta_i + h[-\frac{3}{16} f(\eta_i) + \frac{15}{16} f(\eta_i + \frac{1}{3} h f(\eta_i))]). \end{aligned}$$

Die Abschätzung des lokalen Diskretisierungsfehlers ist wie in Teil (1) und liefert mindestens die Ordnung 2.

Wenn aber die Koeffizienten c_i definiert werden gemäß

$$c_1 = 0, \quad c_2 = a_{21} = \frac{1}{3}, \quad c_3 = a_{31} + a_{32} = \frac{3}{4},$$

steigt die Genauigkeitsordnung auf 3.

Nun kann man hier den Algorithmus geschickt umschreiben, sodass man ohne die Parametertableaus mit wenigen Koeffizienten auskommt. Die Anstiege k_i werden allgemeiner notiert und mit einer kurzen Folge von Zwischengrößen erhält man

$$\eta_{i+1} = \eta_i + h(b_1 k_1 + b_2 k_2 + b_3 k_3).$$

Algorithmus 5.2 (3,3)-RKV für autonome gDGL

Man wählt die Schrittweite h und definiert die zugehörigen Stellen $t_i = t_0 + i h$.

Die Werte η_i an den Knoten t_i sind rekursiv definiert durch

$$\begin{aligned}
 \eta_0 &= x(t_0), \\
 i &= 0, 1, 2, \dots \\
 q_1 &= h f(\eta_i), & k_1 &= f(\eta_i), \\
 z_1 &= \eta_i + \alpha_1 q_1, \\
 q_2 &= h f(z_1) + \beta_1 q_1, & k_2 &= f(z_1) = f(\eta_i + \alpha_1 h k_1), \\
 z_2 &= z_1 + \alpha_2 q_2, \\
 q_3 &= h f(z_2) + \beta_2 q_2, & k_3 &= f(z_2) = f(\eta_i + \alpha_1 h k_1 + \alpha_2 (\beta_1 h k_1 + h k_2)) \\
 & & &= f(\eta_i + (\alpha_1 + \alpha_2 \beta_1) h k_1 + \alpha_2 h k_2), \\
 \eta_{i+1} &= z_2 + \alpha_3 q_3 = \eta_i + \alpha_1 q_1 + \alpha_2 q_2 + \alpha_3 q_3 \\
 &= \eta_i + h [(\alpha_1 + \alpha_2 \beta_1 + \alpha_3 \beta_1 \beta_2) k_1 + (\alpha_2 + \alpha_3 \beta_2) k_2 + \alpha_3 k_3].
 \end{aligned} \tag{5.22}$$

Damit ist eine kompakte Implementierung in einer Programmiersprache möglich, insbesondere auch für großdimensionale SysgDGL.

Das zugehörige Parametertableau ist

$$\begin{array}{c|ccc}
 0 & & & \\
 c_2 & \alpha_1 & & \\
 c_3 & \alpha_1 + \alpha_2 \beta_1 & \alpha_2 & \\
 \hline
 1 & \alpha_1 + \alpha_2 \beta_1 + \alpha_3 \beta_1 \beta_2 & \alpha_2 + \alpha_3 \beta_2 & \alpha_3
 \end{array} . \tag{5.23}$$

Mit den Werten

$$\alpha_1 = \frac{1}{3}, \quad \alpha_2 = \frac{15}{16}, \quad \alpha_3 = \frac{8}{15}, \quad \beta_1 = -\frac{5}{9}, \quad \beta_2 = -\frac{153}{128}$$

stimmt das Tableau (5.23) mit dem Schema (5.20) überein und das RKV ist von der Konsistenzordnung 3.

Definition 5.3 Ein explizites s -stufiges RKV für die gDGL $x' = f(t, x)$ ist ein ESV der Form (5.8) $\eta_{i+1} = \eta_i + h \Psi(t_i, \eta_i, h)$ mit

$$\Psi(t_i, \eta_i, h) = \sum_{i=1}^s b_i k_i \tag{5.24}$$

(Achtung: Umbezeichnung der Indizes) und

$$k_i = f\left(t_i + c_i h, \eta_i + h \sum_{j=1}^{i-1} a_{ij} k_j\right), \quad i = 1, 2, \dots, s. \tag{5.25}$$

Übung 5.1 Zeigen Sie, dass für die autonome gDGL ein 3-stufiges RKV mindestens die Ordnung 2 hat, wenn gilt

$$b_1 + b_2 + b_3 = 1, \quad (5.26)$$

$$b_2 a_{21} + b_3 (a_{31} + a_{32}) = \frac{1}{2}. \quad (5.27)$$

Ähnliche Abschätzungen wie in Beispiel 5.2 können auch für nicht autonome gDGL durchgeführt werden, auch für höherstufige RKV, welche wie folgt definiert sind. Dazu folgt die MATLAB-Funktion mit $t \in [t_0, t_1]$, $x(t_0) = x_0$ und $h = (t_1 - t_0)/N$.

Die Koeffiziententableaus A , \mathbf{b} , \mathbf{c} werden als Funktionsparameter einbezogen.

Da MATLAB konsequent zwischen Zeilen- und Spaltenvektoren unterscheidet, kann man für die Programmierung geschickt Skalarprodukte notieren wie $\mathbf{k}(1:i-1)*\mathbf{A}(i,1:i-1)'$ für $a_{i,1}k_1 + \dots + a_{i,i-1}k_{i-1}$ bzw. $\mathbf{k}*\mathbf{b}$ für $b_1k_1 + \dots + b_s k_s$.

```
% Datei rk1.m
% Explizite s-stufige Runge-Kutta-Verfahren fuer gDGL
% mit Koeffiziententableaus A,b,c

function [t,x] = rk1(A,b,c,f,t0,t1,x0,N)
    if N>0, h = (t1-t0)/N; else h = 0; end
    t(1) = t0; x(1) = x0;
    s = size(b,1);
    k = zeros(1,s);

    for l = 1:N
        t(l+1) = t0+l*h;
        k(1) = feval(f,t(1),x(1));
        for i = 2:s
            k(i) = feval(f,t(1)+h*c(i),x(1)+h*k(1:i-1)*A(i,1:i-1)');
        end;
        x(l+1) = x(1)+h*k*b;
    end;
```

Die Anweisungen der l -Schleife in `rk1` kann man auch verkürzt notieren, wobei man dann pro Schritt wegen der Nulleinträge in der Matrix A mehr als $s^2/2$ Multiplikationen mit Null in Kauf nimmt (Version `rk1a`).

```
% Datei rk1a.m
function [t,x] = rk1a(A,b,c,f,t0,t1,x0,N)
    if N>0, h = (t1-t0)/N; else h = 0; end
    t(1) = t0; x(1) = x0;
    s = size(b,1);
    k = zeros(1,s);
    for l = 1:N
        t(l+1) = t0+l*h;
        for i = 1:s
            k(i) = feval(f,t(1)+h*c(i),x(1)+h*k*A(i,:))';
        end;
        x(l+1) = x(1)+h*k*b;
    end;
```

Es soll noch auf eine Modifikation `rk1b` von `rk1` hingewiesen werden, die jedoch in manchen älteren MATLAB-Versionen zu Fehlern führt. Das Problem, welches hier in der inneren i -Schleife auftritt, ist entsprechend kommentiert.

```
% Datei rkv1b.m
...
for l = 1:N
    t(l+1) = t0+l*h;
    for i = 1:s
        k(i) = feval(f,t(l)+h*c(i),x(l)+h*k(1:i-1)*A(i,1:i-1)');

        % -> Fehler bei i=1 wegen 1:i-1=[], deshalb auch k(1)=[]
        % damit wird die Komponente k(1) geloescht und
        % k(:) hat nur noch s-1 Komponenten

    end;
    x(l+1) = x(l)+h*k*b;
end;
```

Analog programmiert man die vektorielle Version.

```
% Datei rkv.m
% Explizite Runge-Kutta-Verfahren fuer SysgDgl
% mit Koeffiziententableaus A,b,c

function [t,x] = rkv(A,b,c,f,t0,t1,x0,N)
    if N>0, h = (t1-t0)/N; else h = 0; end
    n = size(x0,1);
    x = zeros(n,N+1);
    t(1) = t0;
    x(:,1) = x0;
    s = size(b,1);
    k = zeros(n,s);

    for l=1:N
        t(l+1) = t0+l*h;
        k(:,1) = feval(f,t(l),x(:,l));
        for i=2:s
            k(:,i) = feval(f,t(l)+h*c(i),x(:,l)+h*k(:,1:i-1)*A(i,1:i-1)');
        end;
        x(:,l+1) = x(:,l)+h*k*b;
    end;
```

Es soll das 3-stufige RKV 2. Ordnung aus Beispiel 5.2 für das AWP $x' = tx$, $x(0) = 1$, $t \in [0, 5]$, mit $N = 5000$ gerechnet werden.

Die MATLAB-Anweisungen sind

```
format long
t0 = 0
t1 = 5
x0 = 1
N = 5000

% Parameterschema
A = [0 0 0; 1/2 0 0; 1/3 2/3 0];
b = [1/3 1/3 1/3]';
c = [0 1/2 1]';

[t,x] = rkv1(A,b,c,'f8',t0,t1,x0,N);
% Auswertung der Naehungsloesung
```

Analog kann man alle bisherigen ESV auf diese Weise notieren und rechnen.

Durch die Taylor-Entwicklung lassen sich die Bedingungen für die Konsistenzordnung s -stufiger RKV angeben.

Satz 5.2 Zwischen den Koeffizienten c_i und a_{ij} gelte zusätzlich die Beziehung⁴

$$c_i = \sum_{j=1}^s a_{ij}. \quad (5.28)$$

Dann besitzt das s -stufige RKV genau dann (mindestens) die Konsistenzordnung **1**, wenn

$$\mathbf{b}^T \mathbf{1} = \sum_{i=1}^s b_i = 1, \quad (5.29)$$

2, wenn zusätzlich gilt

$$\mathbf{b}^T \mathbf{c} = \sum_{i=1}^s b_i c_i = \frac{1}{2}, \quad (5.30)$$

3, wenn zusätzlich gilt

$$\sum_{i=1}^s b_i c_i^2 = \frac{1}{3}, \quad \mathbf{b}^T \mathbf{A} \mathbf{c} = \sum_{i,j=1}^s b_i a_{ij} c_j = \frac{1}{6}, \quad (5.31)$$

4, wenn zusätzlich gilt

$$\sum_{i=1}^s b_i c_i^3 = \frac{1}{4}, \quad \sum_{i,j=1}^s b_i a_{ij} c_j^2 = \frac{1}{12}, \quad \sum_{i,j=1}^s b_i c_i a_{ij} c_j = \frac{1}{8}, \quad \mathbf{b}^T \mathbf{A}^2 \mathbf{c} = \sum_{i,j,k=1}^s b_i a_{ij} a_{jk} c_k = \frac{1}{24}. \quad (5.32)$$

Die in diesem Satz angegebenen Gleichungen sind nichtlinear. Deshalb stellt die Bestimmung geeigneter Koeffizienten, die sogenannte Verfahrenskonstruktion, eine schwierige Aufgabe dar, was später gezeigt wird.

Wir wollen im Folgenden die Situation bei $s = 4$ Stufen näher betrachten und Verfahren der Ordnung $p = 4$ konstruieren.

Insgesamt hat man mit der Konvention $c_i = \sum_j a_{ij}$ die 10 Koeffizienten $b_1, b_2, b_3, b_4, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}$, so zu bestimmen, dass sämtliche Bedingungen (5.29) bis (5.32) erfüllt sind.

Die Bedingung $\sum_i b_i = 1$ erinnert sofort an die numerische Integration und legt nahe, b_1, \dots, b_4 als Gewichte einer Quadraturformel zu interpretieren. Fassen wir die c_i zusätzlich als Stützstellen im Intervall $[0, 1]$ auf, so folgt aus den Bedingungen

$$\sum_i b_i c_i = \frac{1}{2}, \quad \sum_i b_i c_i^2 = \frac{1}{3}, \quad \sum_i b_i c_i^3 = \frac{1}{4},$$

dass die Quadraturformel exakt für alle Polynome höchstens 3. Grades sein muss.

⁴Eine Begründung für diese Bedingung findet man auch in Abschnitt 4.2.1 von P. Deuffhard/ F. Bornemann, Numerische Mathematik II, de Gruyter, 1994.

Zwei Quadraturformeln können wir nun heranziehen.

(a) Newton-3/8-Regel

Hier ist

$$\mathbf{c} = (0, \frac{1}{3}, \frac{2}{3}, 1)^T, \quad \mathbf{b} = (\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8})^T.$$

Mit dieser Festlegung liefern $\sum b_i a_{ij} c_j = \frac{1}{6}$, $\sum b_i c_i a_{ij} c_j = \frac{1}{8}$, $\sum b_i a_{ij} a_{jk} c_k = \frac{1}{24}$ die Beziehungen $a_{32} = 1$, $a_{42} c_2 + a_{43} c_3 = \frac{1}{3}$, $b_4 a_{43} a_{32} c_2 = \frac{1}{24}$. Damit ist $a_{43} = 1$ und $a_{42} = -1$. Aus der Definition der c_i ergeben sich schließlich noch $a_{21} = \frac{1}{3}$, $a_{31} = -\frac{1}{3}$, $a_{41} = 1$. Zu überprüfen ist noch die übrig gebliebene Bedingung $\sum b_i a_{ij} c_j^2 = \frac{1}{12}$, die von den Koeffizienten tatsächlich erfüllt wird. Das Resultat ist eine (4,4)-RKV, auch Kutta-3/8-Formel genannt.

(b) Simpson-Regel

Sie besitzt eigentlich nur 3 Knoten. Deswegen wird der mittlere verdoppelt, sodass

$$\mathbf{c} = (0, \frac{1}{2}, \frac{1}{2}, 1)^T, \quad \mathbf{b} = (\frac{1}{6}, \frac{2}{6}, \frac{2}{6}, \frac{1}{6})^T.$$

Ein analoges Vorgehen wie unter (a) führt auf das KRKV.

Die Bedingungsgleichungen für Verfahren hoher Ordnung ($p \geq 5$) kann man zwar elegant aufstellen, ihre Zahl nimmt jedoch stark zu.

Ordnung p	1	2	3	4	5	6	7	8	9	10
Mindeststufen s	1	2	3	4	6	7	9	11	12	13
#Bedingungsgleichungen	1	2	4	8	17	37	85	200		
#Koeffizienten	1	3	6	10	21	28	45	66		

Die Koeffizientenbestimmung ist dann nur noch mit weiteren vereinfachenden Annahmen und mit Computer-Algebra-Unterstützung möglich.

Im Folgenden ist aufgeführt, wie viele Stufen notwendig sind, um eine bestimmte Ordnung zu erreichen. Offensichtlich gilt die Ungleichung $s \geq p$.

#Stufen s	1	2	3	4	5	6	7	8	9	11
maximale Ordnung p	1	2	3	4	4	5	6	6	7	8

Koeffizientenbestimmung

Für die $(4, p)$ -RKV mit der höchstmöglichen Genauigkeitsordnung $p = 4$ sollen die aufwändigen Berechnungen der Koeffizienten des Parameterschemas nachvollzogen werden [85]. Damit werden auch die wesentlichen Schritte zum Beweis von Satz 5.2 erbracht.

Ziel ist es, aus dem Vergleich des Taylor-Polynoms zur Verfahrensfunktion $\Psi(t_n, \eta_n, h)$ (5.9) mit ihrer Darstellung mittels der Zuwächse (5.24), (5.25) die Bedingungen für die Koeffizienten abzuleiten.

Dazu notieren wir zunächst die Funktion $x' = f(t, x(t))$ und ihre Ableitungen x'' , x''' , ... mit den symbolischen Differentialoperatoren $D = (\frac{\partial}{\partial t} + f \frac{\partial}{\partial x})$ und D^k aus Abschnitt 3.4.

$$\begin{aligned}
x' &= f(t, x(t)), \\
x'' &= \left(\frac{\partial}{\partial t} + f \frac{\partial}{\partial x} \right) f = Df, \\
x''' &= \left(\frac{\partial^2}{\partial t^2} + 2f \frac{\partial^2}{\partial t \partial x} + f^2 \frac{\partial^2}{\partial x^2} \right) f + f_x \left(\frac{\partial}{\partial t} + f \frac{\partial}{\partial x} \right) f = D^2 f + f_x Df, \\
x'''' &= D^3 f + f_x D^2 f + f_x^2 Df + 3Df Df_x, \\
x''''' &= D^4 f + 6Df D^2 f_x + 4D^2 f Df_x + f_x^2 D^2 f + \\
&\quad + f_x^3 D^2 f + 3f_{xx} (Df)^2 + f_x D^3 f + 7f_x Df Df_x.
\end{aligned} \tag{5.33}$$

Dann brauchen wir die Taylor-Reihen

$$\begin{aligned}
x(t+h) &= x(t) + \sum_{i=1}^p \frac{x^{(i)}(t)}{i!} h^i + R_p = x(t) + h \sum_{i=1}^p \frac{x^{(i)}(t)}{i!} h^{i-1} + R_p \\
&= x(t) + h \sum_{i=1}^p \frac{f^{(i-1)}(t, x(t))}{i!} h^{i-1} + R_p, \\
x(t_{n+1}) &= x(t_n) + \sum_{i=1}^p \frac{x^{(i)}(t_n)}{i!} h^i + R_p.
\end{aligned} \tag{5.34}$$

Man erhält den lokalen Diskretisierungsfehler auch dadurch, dass man in das Diskretisierungsverfahren anstelle von η_n den exakten Lösungswert $x(t_n)$ einsetzt. Das bedeutet

$$\begin{aligned}
\eta_{n+1} &= \eta_n + h \Psi(t_n, \eta_n, h), \\
0 &= \frac{1}{h} (\eta_{n+1} - \eta_n) - \Psi(t_n, \eta_n, h), \\
\tau(t_n, x_n, h) &= \tau_{n+1} = \frac{1}{h} (x_{n+1} - x_n) - \Psi(t_n, x_n, h), \quad x_k = x(t_k).
\end{aligned} \tag{5.35}$$

Somit stehen sich gegenüber die beiden Formeln

$$\Psi(t_n, x_n, h) = \sum_{i=1}^p \frac{f^{(i-1)}(t_n, x_n)}{i!} h^{i-1} \tag{5.36}$$

und

$$\begin{aligned}
\Psi(t_n, x_n, h) &= \sum_{i=1}^s b_i f_{i,n} = \sum_{i=1}^s b_i k_i, \quad \text{wobei} \\
k_1 &= f_{1,n} = f(t_n, x_n), \\
k_i &= f_{i,n} = f\left(t_n + c_i h, x_n + h \sum_{j=1}^{i-1} a_{ij} f_{j,n}\right), \quad i = 2, 3, \dots, s.
\end{aligned} \tag{5.37}$$

Aus (5.33), (5.34) folgt

$$x_{n+1} - x_n = \sum_{i=1}^5 \frac{h^i}{i!} x_n^{(i)} + R_5,$$

$$\begin{aligned}
x_{n+1} - x_n &= \left[h f + \frac{h^2}{2!} Df + \frac{h^3}{3!} (D^2 f + f_x Df) + \right. \\
&\quad + \frac{h^4}{4!} (D^3 f + f_x D^2 f + f_x^2 Df + 3Df Df_x) + \\
&\quad \left. + \frac{h^5}{5!} (D^4 f + \dots) \right]_n + \mathcal{O}(h^6).
\end{aligned} \tag{5.38}$$

Jetzt notieren wir die allgemeine Taylor-Reihenentwicklung in der symbolischen Schreibweise für

$$f\left(t_n + c_i h, x_n + h\left(\sum_{j=1}^{i-1} a_{ij}\right) f_n\right) = \sum_{k=0}^{\infty} \frac{h^k}{k!} D_i^k f(t_n, x_n), \quad f_n = k_1, \tag{5.39}$$

mit dem Differentialoperator

$$D_i = c_i \frac{\partial}{\partial t} + \left(\sum_{j=1}^{i-1} a_{ij}\right) f_n \frac{\partial}{\partial x}.$$

Damit kann man durch symbolisches Potenzieren die Glieder höherer Ordnung D_i^k angeben.

Unter Verwendung von (5.39) wird nun die Formel

$$x_{n+1} - x_n = h \sum_{i=1}^s b_i f_{i,n} = h \sum_{i=1}^s b_i k_i \tag{5.40}$$

mit den Anstiegen k_i ausgewertet.

(1) Es ist $c_1 = 0$ und $k_1 = f_{1,n} = f(t_n, x_n) = f_n$.

$$\begin{aligned}
(2) \quad f_{2,n} &= f(t_n + c_2 h, x_n + h a_{21} k_1) = f(t_n + c_2 h, x_n + h a_{21} f_n), \quad i = 2 \\
&= \sum_{k=0}^{\infty} \frac{h^k}{k!} D_2^k f(t_n, x_n),
\end{aligned}$$

$$k_2 = f(t_n, x_n) + h \left[c_2 \frac{\partial f_n}{\partial t} + a_{21} \frac{\partial f_n}{\partial x} \right] + \dots$$

$$\begin{aligned}
(3) \quad f_{3,n} &= f(t_n + c_3 h, x_n + h(a_{31} k_1 + a_{32} k_2)), \quad i = 3 \\
&= f(t_n + c_3 h, x_n + h(a_{31} + a_{32}) k_1 + h a_{32} (k_2 - k_1)) \\
&= \sum_{k=0}^{\infty} \frac{h^k}{k!} (D_3 + a_{32} (k_2 - k_1))^k f(t_n, x_n),
\end{aligned}$$

$$k_3 = f(t_n, x_n) + h \left[c_3 \frac{\partial f_n}{\partial t} + (a_{31} + a_{32}) \frac{\partial f_n}{\partial x} + a_{32} (k_2 - k_1) \frac{\partial f_n}{\partial x} \right] + \dots$$

$$\begin{aligned}
(4) \quad f_{4,n} &= f(t_n + c_4 h, x_n + h(a_{41} k_1 + a_{42} k_2 + a_{43} k_3)), \quad i = 4 \\
&= f(t_n + c_4 h, x_n + h(a_{41} k_1 + a_{41} (k_1 - k_1) \\
&\quad + a_{42} k_1 + a_{42} (k_2 - k_1) \\
&\quad + a_{43} k_1 + a_{43} (k_3 - k_1))),
\end{aligned}$$

$$\begin{aligned}
f_{4,n} &= f\left(t_n + c_4 h, x_n + h \sum_{m=1}^3 [a_{4,m} k_1 + a_{4,m} (k_m - k_1)]\right) \\
&= f\left(t_n + c_4 h, x_n + h\left(\sum_{m=1}^3 a_{4,m}\right) k_1 + h \sum_{m=2}^3 a_{4,m} (k_m - k_1)\right) \\
&= \sum_{k=0}^{\infty} \frac{h^k}{k!} \left(D_4 + \sum_{m=2}^3 a_{4,m} (k_m - k_1)\right)^k f(t_n, x_n), \\
k_4 &= f(t_n, x_n) + h \left[c_4 \frac{\partial f_n}{\partial t} + (a_{41} + a_{42} + a_{43}) \frac{\partial f_n}{\partial x} + \left(\sum_{m=2}^3 a_{4,m} (k_m - k_1)\right) \frac{\partial f_n}{\partial x} \right] + \dots .
\end{aligned}$$

(5) Allgemein

$$\begin{aligned}
f_{i,n} &= f\left(t_n + c_i h, x_n + h \sum_{m=1}^{i-1} a_{im} k_i\right) \\
&= f\left(t_n + c_i h, x_n + h\left(\sum_{m=1}^3 a_{im}\right) k_1 + h \sum_{m=1}^3 a_{im} (k_m - k_1)\right) \\
&= \sum_{k=0}^{\infty} \frac{h^k}{k!} \left(D_i + \left(\sum_{m=2}^{i-1} a_{im} (k_m - k_1)\right) \frac{\partial}{\partial x}\right)^k f(t_n, x_n), \\
k_i &= f(t_n, x_n) + h \left[c_i \frac{\partial f_n}{\partial t} + \left(\sum_{m=1}^{i-1} a_{im}\right) \frac{\partial f_n}{\partial x} + \left(\sum_{m=2}^{i-1} a_{im} (k_m - k_1)\right) \frac{\partial f_n}{\partial x} \right] + \dots .
\end{aligned}$$

Das sind alles Entwicklungen bzgl. der Potenz h im Punkt (t_n, x_n) .

(1) $k_1 = f = f(t_n, x_n)$.

(2) $k_2 = f + h D_2 f + \frac{h^2}{2!} D_2^2 f + \frac{h^3}{3!} D_2^3 f + \frac{h^4}{4!} D_2^4 f + \dots$

$$\begin{aligned}
(3) \quad k_3 &= f + h D_3 f + h a_{32} (k_2 - k_1) f_x + \frac{h^2}{2} \left(D_3 + h a_{32} (k_2 - k_1) \frac{\partial}{\partial x}\right)^2 f + \\
&\quad + \frac{h^3}{6} \left(D_3 + h a_{32} (k_2 - k_1) \frac{\partial}{\partial x}\right)^3 f + \frac{h^4}{24} \left(D_3 + h a_{32} (k_2 - k_1) \frac{\partial}{\partial x}\right)^4 f + \dots, \\
&\quad \text{wobei } k_2 - k_1 = h D_2 f + \frac{h^2}{2!} D_2^2 f + \frac{h^3}{3!} D_2^3 f + \frac{h^4}{4!} D_2^4 f + \dots, \\
&= f + h D_3 f + h^2 \left(\frac{1}{2} D_3^2 f + a_{32} f_x D_2 f\right) + h^3 \left(\frac{1}{6} D_3^3 f + \frac{a_{32}}{2} f_x D_2^2 f + a_{32} D_2 f D_3 f_x\right) + \\
&\quad + h^4 \left(\frac{1}{24} D_3^4 f + \frac{a_{32}}{6} f_x D_2^3 f + \frac{a_{32}}{2} D_2^2 f D_3 f_x + \frac{a_{32}^2}{2} f_{xx} (D_2 f)^2 + \frac{a_{32}}{2} D_2 f D_3^2 f_x\right) + \mathcal{O}(h^5).
\end{aligned}$$

$$\begin{aligned}
(4) \quad k_4 &= f + h D_4 f + h^2 \left(\frac{1}{2} D_4^2 f + a_{42} f_x D_2 f + a_{43} f_x D_3 f\right) \\
&\quad + h^3 \left(\frac{1}{6} D_4^3 f + \frac{a_{42}}{2} f_x D_2^2 f + a_{32} a_{43} f_x^2 D_2 f + \right. \\
&\quad \left. + \frac{a_{43}}{2} f_x D_3^2 f + a_{42} D_2 f D_4 f_x + a_{43} D_3 f D_4 f_x\right) + \\
&\quad + h^4 \left(\frac{1}{24} D_4^4 f + \frac{a_{42}}{6} f_x D_2^3 f + \frac{a_{32} a_{43}}{2} f_x^2 D_2^2 f + a_{32} a_{43} f_x D_2 f D_3 f_x + \frac{a_{43}^2}{6} f_x D_3^3 f + \right. \\
&\quad \left. + \frac{a_{42}}{2} D_4 f_x D_2^2 f + \frac{a_{43}}{2} D_4 f_x D_3^2 f + \frac{a_{42}^2}{2} f_{xx} D_2^2 f + a_{42} a_{43} f_{xx} D_2 f D_3 f + \right. \\
&\quad \left. + \frac{a_{43}^2}{2} f_{xx} D_3^2 f + \frac{a_{42}}{2} D_2 f D_4^2 f_x + \frac{a_{43}}{2} D_3 f D_4^2 f_x + a_{32} a_{43} f_x D_2 f D_4 f_x\right) + \mathcal{O}(h^5).
\end{aligned}$$

Der Vergleich der Formeln (5.38) und (5.40) mit $f = f(t_n, x_n)$ heißt

$$h f + \frac{h^2}{2!} D f + \frac{h^3}{3!} (D^2 f + f_x D f) + \frac{h^4}{4!} (D^3 f + \dots) + \frac{h^5}{5!} (D^4 f + \dots) + \mathcal{O}(h^6) = h \sum_{i=1}^s b_i k_i.$$

Man macht den Abgleich der ersten vier Glieder, was auf die folgenden Beziehungen führt.

$$(a) \quad b_1 + b_2 + b_3 + b_4 = 1,$$

$$(b) \quad b_2 D_2 f + b_3 D_3 f + b_4 D_4 f = \frac{1}{2!} Df,$$

$$(c) \quad \frac{1}{2} (b_2 D_2^2 f + b_3 D_3^2 f + b_4 D_4^2 f) + f_x (b_3 a_{32} D_2 f + b_4 (a_{42} D_2 f + a_{43} D_3 f)) = \frac{1}{3!} (D^2 f + f_x Df),$$

$$(d) \quad \frac{1}{6} (b_2 D_2^3 f + b_3 D_3^3 f + b_4 D_4^3 f) + \frac{1}{2} f_x (b_3 a_{32} D_2^2 f + b_4 (a_{42} D_2^2 f + a_{43} D_3^2 f)) + b_3 a_{32} D_2 f D_3 f_x + b_4 (a_{42} D_2 f D_4 f_x + a_{43} D_3 f D_4 f_x) + b_4 a_{32} a_{43} f_x^2 D_2 f = \frac{1}{4!} (D^3 f + f_x D^2 f + 3 Df Df_x + f_x^2 Df).$$

Die vier Bedingungen (a) – (d) für die dreizehn Unbekannten im Parametertableau

$$\begin{array}{c|cccc} c_1 = 0 & & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ c_4 & a_{41} & a_{42} & a_{43} & \\ \hline 1 & b_1 & b_2 & b_3 & b_4 \end{array}$$

lassen Freiheitsgrade zu.

Faktisch haben wir hier acht Gleichungen, weil in den Formeln (c) und (d) auch die inneren Komponenten paarweise übereinstimmen müssen. In (c) sind es der absolute Term sowie der Term bei f_x , in (d) sind es der absolute Term sowie die Terme bei f_x , $D_k f_x$ und f_x^2 . Die Formeln (a) und (b) liefern jeweils eine Gleichung.

Darüber hinaus sollen die acht Gleichungen unabhängig von $f(t, x)$ sein. Das bedeutet, dass die Quotienten

$$\frac{D_j f}{Df}, \quad j = 2, 3, 4, \quad \frac{D_j f_x}{Df_x}, \quad j = 3, 4, \quad (5.41)$$

konstant sind. Dieser Umstand ist verknüpft mit den drei Grundbedingungen

$$c_i = \sum_{j=1}^{i-1} a_{ij}, \quad i = 2, 3, 4. \quad (5.42)$$

Letztendlich bleiben also elf Gleichungen, drei gemäß (5.42) und noch weitere acht aus (a) – (d), für die dreizehn unbekannt Parameter und damit zwei Freiheitsgrade.

$$\begin{aligned} 1 &= b_1 + b_2 + b_3 + b_4 \quad \text{aus (a)}, \\ \frac{1}{2} &= c_2 b_2 + c_3 b_3 + c_4 b_4 \quad \text{aus (b)}, \\ \frac{1}{3} &= c_2^2 b_2 + c_3^2 b_3 + c_4^2 b_4 \quad \text{aus (c)}, \\ \frac{1}{6} &= c_2 a_{32} b_3 + (c_2 a_{42} + c_3 a_{43}) b_4, \\ \frac{1}{4} &= c_2^3 b_2 + c_3^3 b_3 + c_4^3 b_4, \quad \text{aus (d)}, \\ \frac{1}{12} &= c_2^2 a_{32} b_3 + (c_2^2 a_{42} + c_3^2 a_{43}) b_4, \\ \frac{1}{8} &= c_2 c_3 a_{32} b_3 + c_4 (c_2 a_{42} + c_3 a_{43}) b_4, \\ \frac{1}{24} &= c_2 a_{32} a_{43} b_4. \end{aligned} \quad (5.43)$$

Daraus ergibt sich die Klasse von 4-stufigen RKV mit dem KRKV als Spezialfall.

Aus der letzten Gleichung folgt, dass die Parameter $c_2 = a_{21}, a_{32}, a_{43}, b_4$ nicht verschwinden, d.h. die äußere Diagonale des Parameterschemas enthält nur Nichtnullelemente. Im Allg. verwendet man Schemata mit wachsenden Koeffizienten c_i .

2-stufige RKV brauchen die Bedingungen (a) und (b) und führen auf das System

$$\begin{aligned} c_2 &= a_{21}, \\ 1 &= b_1 + b_2 \quad \text{aus (a)}, \\ \frac{1}{2} &= c_2 b_2 \quad \text{aus (b)}, \quad c_2, b_2 \neq 0. \end{aligned} \tag{5.44}$$

Spezialfälle sind die mit $c_2 = \frac{1}{2}, \frac{2}{3}, 1$, was auf die folgenden RKV führt:

MPZV (5.2), (2,2)-RKV mit kleinstem Fehler sowie HV (5.6).

Man vergleiche die Ergebnisse auch mit der Aussage im Satz 5.1.

3-stufige RKV benötigen die Bedingungen (a), (b) und (c) und führen auf das Gleichungssystem

$$\begin{aligned} c_2 &= a_{21}, \\ c_3 &= a_{31} + a_{32}, \\ 1 &= b_1 + b_2 + b_3 \quad \text{aus (a)}, \\ \frac{1}{2} &= c_2 b_2 + c_3 b_3 \quad \text{aus (b)}, \\ \frac{1}{6} &= c_2^2 b_2 + c_3^2 b_3, \quad \frac{1}{6} = c_2 a_{32} b_3 \quad \text{aus (c)}, \quad c_2, b_3 \neq 0. \end{aligned} \tag{5.45}$$

Man notiert die zweiparametrische Menge der Lösungen in Abhängigkeit von c_2 und c_3 . Dabei seien $c_2 \neq c_3$ und $c_2, c_3 \neq 0$.

Die Formeln für die anderen Parameter lauten

$$\begin{aligned} b_1 &= 1 + \frac{2 - 3(c_2 + c_3)}{6c_2c_3}, \quad b_2 = \frac{3c_3 - 2}{6c_2(c_3 - c_2)}, \quad b_3 = \frac{2 - 3c_2}{6c_3(c_3 - c_2)}, \\ a_{21} &= c_2, \quad a_{31} = \frac{3c_2c_3(1 - c_2) - c_3^2}{c_2(2 - 3c_2)}, \quad a_{32} = \frac{c_3^2 - c_3c_2}{c_2(2 - 3c_2)}. \end{aligned} \tag{5.46}$$

Interessante Fälle sind die folgenden:

$c_2 = \frac{1}{2}, c_3 = 1$: (3,3)-Kutta-Verfahren (Simpson-Regel),

$c_2 = \frac{1}{2}, c_3 = \frac{3}{4}$: (3,3)-RKV mit kleinstem Fehler,

$c_2 = 1, c_3 = \frac{1}{2}$: (3,3)-RK-Fehlberg-Formel.

Sonderfälle

(1) $c_2 = \frac{2}{3}, c_3 = \frac{2}{3}$: (3,3)-RKV mit

$$b_1 = \frac{1}{4}, \quad b_2 = \frac{3}{4} - b_3, \quad a_{21} = \frac{2}{3}, \quad a_{31} = \frac{2}{3} - \frac{b_3}{4}, \quad a_{32} = \frac{b_3}{4}.$$

Parameterschema

$$\begin{array}{c|ccc} \frac{2}{3} & & \frac{2}{3} & \\ \frac{2}{3} & \frac{2}{3} - \frac{b_3}{4} & \frac{b_3}{4} & \\ \hline & \frac{1}{4} & \frac{3}{4} - b_3 & b_3 \end{array}$$

$$b_3 = \frac{3}{4}: (3,3)\text{-Heun-Verfahren}, \quad b_3 = \frac{3}{8}: (3,3)\text{-Nyström-Verfahren}.$$

(2) $c_2 = \frac{2}{3}$, $c_3 = 0$: (3,3)-RKV mit

$$b_2 = \frac{3}{4}, \quad b_1 = \frac{1}{4} - b_3, \quad a_{21} = \frac{2}{3}, \quad a_{31} = -\frac{1}{4b_3}, \quad a_{32} = \frac{1}{4b_3}.$$

Parameterschema

$$\begin{array}{c|cc} \frac{2}{3} & \frac{2}{3} & \\ 0 & -\frac{1}{4b_3} & \frac{1}{4b_3} \\ \hline & \frac{1}{4} - b_3 & \frac{3}{4} \quad b_3 \end{array}$$

(3) $c_2 = \frac{1}{2}$, $c_3 = 1$: (3,2)-RKV mit Parameterschema

$$\begin{array}{c|cc} \frac{1}{2} & \frac{1}{2} & \\ 1 & \frac{1}{3} & \frac{2}{3} \\ \hline & \frac{1}{3} & \frac{1}{3} \quad \frac{1}{3} \end{array}$$

4-stufige RKV mit dem Gleichungssystem (5.43), was den Bedingungen (5.29) – (5.32) entspricht, notiert man ebenfalls als zweiparametrische Menge der Lösungen in Abhängigkeit von c_2 und c_3 .

Dabei sind natürlich nur die Fälle $c_2 = a_{21}$, a_{32} , a_{43} , $b_4 \neq 0$ zugelassen. Aber für die nachfolgenden Formeln braucht man zusätzlich $c_2 \neq 1$, $c_3 \neq 0, 1$, $c_2 \neq c_3$ sowie das Nichtverschwinden der Nenner in den Formeln für a_{32} , a_{42} , a_{43} , die danach als Sonderfälle betrachtet werden.

$$\begin{aligned} b_1 &= \frac{1}{2} + \frac{1 - 2(c_2 + c_3)}{12c_2c_3}, & b_2 &= \frac{2c_3 - 1}{12c_2(c_3 - c_2)(1 - c_2)}, \\ b_3 &= \frac{1 - 2c_2}{12c_3(c_3 - c_2)(1 - c_3)}, & b_4 &= \frac{1}{2} + \frac{-3 + 2(c_2 + c_3)}{12(1 - c_2)(1 - c_3)}, \\ c_4 &= 1, \\ a_{21} &= c_2, \\ a_{32} &= \frac{c_3(c_3 - c_2)}{2c_2(1 - 2c_2)}, & a_{31} &= c_3 - a_{32}, \\ a_{42} &= \frac{(1 - c_2)(c_2 + c_3 - 1 - (2c_3 - 1)^2)}{2c_2(c_3 - c_2)(6c_2c_3 - 4(c_2 + c_3) + 3)}, \\ a_{43} &= \frac{(1 - 2c_2)(1 - c_2)(1 - c_3)}{c_3(c_3 - c_2)(6c_2c_3 - 4(c_2 + c_3) + 3)}, \\ a_{41} &= c_4 - a_{42} - a_{43}. \end{aligned} \tag{5.47}$$

Wichtig ist der Nachweis von $c_4 = 1$.

Wir nummerieren im Gleichungssystem (5.43) der Reihe nach die Formeln von (1) bis (8) und bilden geeignete Linearkombination von Gleichungen, um systematisch Parameter zu eliminieren.

$$(9) : c_4 \cdot (4) - (7), \quad a_{32} = \frac{4c_4 - 3}{24(c_4 - c_3)c_2 b_3},$$

$$(10) : (6) - c_2 \cdot (4), \quad a_{43} = \frac{1 - 2c_2}{12c_2(c_3 - c_2)b_4},$$

(11): Elimination von b_4 aus (8) und (10)

$$c_3(c_3 - c_2) = \frac{1}{2}(1 - 2c_2)c_2 a_{32},$$

(12): Elimination von a_{32} und b_3 aus (9) und (11)

$$a_{32} = \frac{c_3(c_3 - c_2)}{2c_2(1 - 2c_2)},$$

$$b_3 = \frac{4c_4 - 3}{24(c_4 - c_3)c_2 a_{32}} = \frac{(4c_4 - 3)(1 - 2c_2)}{12(c_4 - c_3)c_3(c_3 - c_2)},$$

$$(13) : c_2 c_4 \cdot (2) - (c_2 + c_4) \cdot (3) + (4),$$

(14): Berechnung von c_4 aus (12) und (13): $c_4 = 1$.

Damit folgen auch die anderen Parameter.

Mit b_3 und $c_4 = 1$ folgt aus (2)–(3) die Formel $b_2 = \frac{1/6 - c_3(1 - c_3)b_3}{c_2(1 - c_2)} = \dots$, dann die Größen $b_4 = \frac{1}{2} - c_2 b_2 - c_3 b_3 = \dots$ und $b_1 = 1 - b_2 - b_3 - b_4 = \dots$.

Aus (8) oder (10) ergibt sich unmittelbar a_{43} und aus (4), (6) oder (7) die Größe a_{42} . Es ist $c_3 \neq 1$, weil die Annahme $c_3 = 1$ mit (4) und (7) zu $\frac{1}{6} = c_2 a_{32} b_3 = \frac{1}{8}$ führt, was einen Widerspruch darstellt.

Interessante Fälle sind die folgenden:

$$c_2 = \frac{1}{3}, \quad c_3 = \frac{2}{3}: \quad (4,4)\text{-RKV, } 3/8\text{-Formel,}$$

$$c_2 = \frac{1}{4}, \quad c_3 = \frac{1}{2}: \quad (4,4)\text{-RKV, Ceschino-Formel,}$$

$$c_2 = \frac{2}{5}, \quad c_3 = \frac{3}{5}: \quad (4,4)\text{-RKV, Kuntzmann-Formel,}$$

$$c_2 = \frac{2}{5}, \quad c_3 = \frac{14 - 3\sqrt{5}}{16}: \quad (4,4)\text{-RKV mit kleinstem Fehler.}$$

Sonderfälle sind einparametrische Lösungsformeln.

(1) $c_2 = c_3$, damit $c_2 = c_3 = \frac{1}{2}$ und freier Parameter ist b_3 .

$$b_1 = b_4 = \frac{1}{6}, \quad b_2 = \frac{2}{3} - b_3, \quad a_{32} = \frac{b_3}{6}, \quad a_{42} = 1 - 3b_3, \quad a_{43} = 3b_3,$$

KRKV (Simpson-Regel) mit $b_3 = \frac{1}{3}$,

England-Formel mit $b_3 = \frac{2}{3}$.

Wenn $c_2 = \frac{1}{2}$ ist, dann ist $a_{32} = \infty$, es sei denn auch $c_3 = \frac{1}{2}$.

Nur mit $c_3 = \frac{1}{2}$ werden die Formeln a_{32} , a_{42} , a_{43} als "0/0"-Ausdrücke trotzdem endliche Grenzwerte liefern.

(2) $c_2 = 1$ und freier Parameter ist $b_4 \neq 0$.

$$c_3 = \frac{1}{2}, \quad b_1 = \frac{1}{6}, \quad b_2 = \frac{1}{6} - b_4, \quad b_3 = \frac{2}{3}, \quad a_{32} = \frac{1}{8}, \quad a_{42} = -\frac{b_4}{12}, \quad a_{43} = \frac{b_4}{3}.$$

Parameterschema allgemein und für $b_4 = \frac{1}{6}$

$$\begin{array}{c|ccc} 1 & 1 & & \\ \frac{1}{2} & \frac{3}{8} & \frac{1}{8} & \\ 1 & \frac{4-b_4}{4} & -\frac{b_4}{12} & \frac{b_4}{3} \\ \hline & \frac{1}{6} & \frac{1-6b_4}{6} & \frac{2}{3} \quad b_4 \end{array} \quad \begin{array}{c|ccc} 1 & 1 & & \\ \frac{1}{2} & \frac{3}{8} & \frac{1}{8} & \\ 1 & \frac{69}{72} & -\frac{1}{72} & \frac{1}{18} \\ \hline & \frac{1}{6} & 0 & \frac{2}{3} \quad \frac{1}{6} \end{array}$$

(3) $c_3 = 0$ und freier Parameter ist $b_3 \neq 0$.

$$c_2 = \frac{1}{2}, \quad b_1 = \frac{1}{6} - b_3, \quad b_2 = \frac{2}{3}, \quad b_4 = \frac{1}{6}, \quad a_{32} = \frac{b_3}{12}, \quad a_{42} = \frac{3}{2}, \quad a_{43} = 6b_3.$$

Parameterschema allgemein und für $b_3 = \frac{1}{6}$

$$\begin{array}{c|ccc} \frac{1}{2} & \frac{1}{2} & & \\ 0 & -\frac{b_3}{12} & \frac{b_3}{12} & \\ 1 & -\frac{1+12b_3}{2} & \frac{3}{2} & 6b_3 \\ \hline & \frac{1-6b_3}{6} & \frac{2}{3} & b_3 \quad \frac{1}{6} \end{array} \quad \begin{array}{c|ccc} \frac{1}{2} & \frac{1}{2} & & \\ 0 & -\frac{1}{72} & \frac{1}{72} & \\ 1 & -\frac{3}{2} & \frac{3}{2} & 1 \\ \hline & 0 & \frac{2}{3} & \frac{1}{6} \quad \frac{1}{6} \end{array}$$

Fehlerbestimmung

Um den lokalen Diskretisierungsfehler besser zu beschreiben, ermittelt man oft den führenden Fehlerterm von τ_{n+1} in den Beziehungen (5.35).

Hat dieser die h -Potenz p , so notiert man

$$\tau_{n+1} = \gamma_p h^p + \mathcal{O}(h^{p+1}), \quad (5.48)$$

wobei natürlich γ_p von $f(t, x)$ abhängt.

Man kann diesen Umstand auch so formulieren.

Das ESV $\eta_{n+1} = \eta_n + h\Psi(t_n, \eta_n, h)$ ist genau für die Potenzen von h bis zu h^p oder $x_{n+1} - x_n - h\Psi(t_n, x_n, h) = \mathcal{O}(h^{p+1})$ bei Einsetzen der exakten Lösung in das ESV. Es wird dabei angenommen, dass die weiteren Fehlerglieder nicht ins Gewicht fallen.

Für die (s, p) -RKV, speziell für die ersten (p, p) -RKV ($p \leq 4$), sollen die Berechnungen zum Fehlerkoeffizient γ_p gemacht werden [85].

Wir verwenden die Beziehungen (5.38) und (5.40) mit den zugehörigen Anstiegen k_1, k_2, k_3, k_4 .

Das RKV der Ordnung 2 mit seinem Parameterschema

$$\begin{array}{c|cc} c_1 = 0 & & \\ c_2 & a_{21} & \\ \hline 1 & b_1 & b_2 \end{array}$$

wird sich beim Fehlerabgleich bezüglich der h^2 -Potenz zunächst von seiner Langform

$$(c) \quad \frac{1}{2} (b_2 D_2^2 f + b_3 D_3^2 f + b_4 D_4^2 f) + f_x (b_3 a_{32} D_2 f + b_4 (a_{42} D_2 f + a_{43} D_3 f)) = \frac{1}{3!} (D^2 f + f_x D f)$$

wegen $c_i = b_i = a_{ij} = 0, i \geq 3$, zur Gleichung $\frac{1}{2} b_2 D_2^2 f = \frac{1}{3!} (D^2 f + f_x D f)$ vereinfachen.

Damit ergibt sich der Fehlerterm

$$\begin{aligned}
 \gamma_2 &= \frac{1}{3!} (D^2 f + f_x Df) - \frac{1}{2} b_2 D_2^2 f \\
 &= \frac{1}{6} D^2 f - \frac{1}{2} b_2 (c_2 \frac{\partial}{\partial t} + a_{21} f \frac{\partial}{\partial x})^2 f + \frac{1}{6} f_x Df, \quad c_2 = a_{21} \\
 &= \frac{1}{6} D^2 f - \frac{1}{2} c_2^2 b_2 (\frac{\partial}{\partial t} + f \frac{\partial}{\partial x})^2 f + \frac{1}{6} f_x Df \\
 &= \frac{1}{6} D^2 f - \frac{1}{2} c_2^2 b_2 D^2 f + \frac{1}{6} f_x Df \\
 &= \frac{1}{6} (1 - 3 c_2^2 b_2) D^2 f + \frac{1}{6} f_x Df.
 \end{aligned} \tag{5.49}$$

Analog folgen aus (d) und dem nächsten Ableich die Koeffizienten γ_3 und γ_4 .

$$\begin{aligned}
 \gamma_3 &= \frac{1}{24} (1 - 4(c_2^3 b_2 + c_3^3 b_3)) D^3 f + \frac{1}{24} (1 - 12 c_2^2 a_{32} b_3) f_x Df + \\
 &\quad + (\frac{1}{8} - c_2 c_3 a_{32} b_3) Df Df_x + \frac{1}{24} f_x^2 Df,
 \end{aligned} \tag{5.50}$$

$$\begin{aligned}
 \gamma_4 &= \frac{1}{120} (1 - 5(c_2^4 b_2 + c_3^4 b_3 + c_4^4 b_4)) D^4 f + \\
 &\quad + \frac{1}{20} (1 - 10(c_2 c_3^2 a_{32} b_3 + c_4^2 (c_2 a_{42} + c_3 a_{43}) b_4)) D^2 f_x Df + \\
 &\quad + \frac{1}{30} (1 - 15(c_2^2 c_3 a_{32} b_3 + c_4 (c_2^2 a_{42} + c_3^2 a_{43}) b_4)) Df_x D^2 f + \\
 &\quad + \frac{1}{120} (1 - 60 c_2^2 a_{32} a_{43} b_4) f_x^2 D^2 f + \\
 &\quad + \frac{1}{40} (1 - 20(c_2^2 a_{32}^2 b_3 + (c_2 a_{42} + c_3 a_{43})^2 b_4)) f_{xx} D^2 f + \\
 &\quad + \frac{1}{120} (1 - 20(c_2^3 a_{32} b_3 + (c_2^3 a_{42} + c_3^3 a_{43}) b_4)) f_x D^3 f + \\
 &\quad + (\frac{7}{120} - c_2 (c_3 + c_4) a_{32} a_{43} b_4) f_x Df_x Df + \frac{1}{120} f_x^3 Df.
 \end{aligned} \tag{5.51}$$

Um γ_p in (5.48) abzuschätzen, setzen wir folgende Ungleichungen für die Funktion $f(t, x)$ und ihre Ableitungen in einem Gebiet, das den Punkt (t_n, x_n) und alle anderen auftretenden Punkte enthält, voraus.

$$|f(t, x)| < M, \quad \left| \frac{\partial^{i+j} f}{\partial t^i \partial x^j} \right| < \frac{L^{i+j}}{M^{j-1}}, \quad \text{wobei } i + j \leq p. \tag{5.52}$$

Die Darstellung ist so gewählt, dass weitere Abschätzungen zwar grob sind, aber sich günstig notieren lassen.

Z.B. gilt damit für die Ableitung $D^2 f$ die Ungleichung

$$|D^2 f| = \left| \frac{\partial^2 f}{\partial t^2} + 2f \frac{\partial^2 f}{\partial t \partial x} + f^2 \frac{\partial^2 f}{\partial x^2} \right| < 4ML^2.$$

Damit erhalten wir der Reihe nach

$$|\gamma_2| < (\frac{2}{3} |1 - 3 c_2^2 b_2| + \frac{1}{3}) ML^2, \tag{5.53}$$

$$\begin{aligned}
 |\gamma_3| &< (\frac{1}{3} |1 - 4(c_2^3 b_2 + c_3^3 b_3)| + \frac{1}{6} |1 - 12 c_2^2 a_{32} b_3| + \\
 &\quad + \frac{1}{2} |1 - 8 c_2 c_3 a_{32} b_3| + \frac{1}{24}) ML^3,
 \end{aligned} \tag{5.54}$$

$$\begin{aligned}
|\gamma_4| < & (16|d_1| + 4|d_2| + |d_2 + 3d_3| + |2d_2 + 3d_3| + |d_2 + d_3| + \\
& + |d_3| + 8|d_4| + |d_5| + |2d_5 + d_7| + |d_5 + d_6 + d_7| + |d_6| + \\
& + |2d_6 + d_7| + |d_7| + 2|d_8|) ML^4,
\end{aligned} \tag{5.55}$$

wobei $c_4 = 1$ und

$$\begin{aligned}
d_1 &= \frac{1}{120} (1 - 5(c_2^4 b_2 + c_3^4 b_3 + b_4)), \\
d_2 &= \frac{1}{20} (1 - 10(c_2 c_3^2 a_{32} b_3 + (c_2 a_{42} + c_3 a_{43}) b_4), \\
d_3 &= \frac{1}{120} (1 - 20(c_2^3 a_{32} b_3 + (c_2^3 a_{42} + c_3^3 a_{43}) b_4), \\
d_4 &= \frac{1}{30} (1 - 15(c_2^2 c_3 a_{32} b_3 + (c_2^2 a_{42} + c_3^2 a_{43}) b_4), \\
d_5 &= \frac{1}{120} (1 - 60c_2^2 a_{32} a_{43} b_4), \\
d_6 &= \frac{1}{40} (1 - 20(c_2^2 a_{32}^2 b_3 + (c_2 a_{42} + c_3 a_{43})^2 b_4), \\
d_7 &= \frac{7}{120} - c_2(c_3 + 1) a_{32} a_{43} b_4, \\
d_8 &= \frac{1}{120}.
\end{aligned}$$

Beispiel 5.3 Das KRKV ist 4-stufig, d.h. $s = 4$, von 4. Ordnung und gegeben durch

$$\mathbf{b} = \frac{1}{6} \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{5.56}$$

Die darin vorkommenden Konstanten bzw. charakteristischen Koeffizienten, also die RK-Knoten c_i und RK-Gewichte b_i sowie die Größen a_{ij} , werden in dem von J.C. Butcher⁵ eingeführten Parameterschema übersichtlich zusammengefasst.

Parameterschema

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^T \end{array} = \begin{array}{c|cc} 0 & & \\ \hline \frac{1}{2} & \frac{1}{2} & \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \hline 1 & 0 & 0 & 1 \\ \hline 1 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Die explizite Formulierung des KRKV (5.8) lautet

$$\begin{aligned}
k_1 &= f(t_i, \eta_i), \\
k_2 &= f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hk_1), \\
k_3 &= f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hk_2), \\
k_4 &= f(t_i + h, \eta_i + hk_3), \\
\Psi(t_i, \eta_i, h) &= k = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).
\end{aligned} \tag{5.57}$$

⁵John C. Butcher (1933-...), neuseeländischer Mathematiker, Arbeitsgebiete Numerik von DGL, Wissenschaftliches Rechnen

Das KRKV genügt der Abschätzung (5.55) mit $|\gamma_4| < \frac{73}{720}ML^4$.

```
% Datei krkv1.m
% Explizites Klassisches Runge-Kutta-Verfahren fuer gDGL

function [t,x] = krkv1(f,t0,t1,x0,N)
    if N>0, h = (t1-t0)/N; else h = 0; end
    t(1) = t0;
    x(1) = x0;

    for i = 1:N
        t(i+1) = t0+i*h;
        k1 = feval(f,t(i),x(i));
        k2 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k1);
        k3 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k2);
        k4 = feval(f,t(i)+h,x(i)+h*k3);
        x(i+1) = x(i)+h/6*(k1+2*k2+2*k3+k4);
    end;
end;
```

Genauso kann man die vektorielle Version definieren, wobei sie normalerweise für ein SysgDGL verwendet wird. Sie funktioniert natürlich auch im skalaren Fall.

```
% f9s.m    x(1)(0)=1, x(2)(0)=1
function y = f9s(t,x)
    y = zeros(max(size(x)),1);
    y(1) = t-x(2);
    y(2) = -t+x(1)*x(2);
end

% Datei krkv.m
% Explizites Klassisches Runge-Kutta-Verfahren fuer SysgDGL

function [t,x] = krkv(f,t0,t1,x0,N)
    if N>0, h = (t1-t0)/N; else h = 0; end
    n = size(x0,1);
    x = zeros(n,N+1);
    t(1) = t0;
    x(:,1) = x0;

    for i = 1:N
        t(i+1) = t0+i*h;
        k1 = feval(f,t(i),x(:,i));    % f(t(i),x(:,i))
        k2 = feval(f,t(i)+0.5*h,x(:,i)+0.5*h*k1);
        k3 = feval(f,t(i)+0.5*h,x(:,i)+0.5*h*k2);
        k4 = feval(f,t(i)+h,x(:,i)+h*k3);
        x(:,i+1) = x(:,i)+h/6*(k1+2*k2+2*k3+k4);
    end;
end;
```

Die Einbeziehung des gesamten Parameterschemas ist nicht angebracht, weil die Matrix A nur drei Nichtnulleinträge enthält.

Die geometrische Illustration des KRKV mit den vier Anstiegen ist wie folgt.

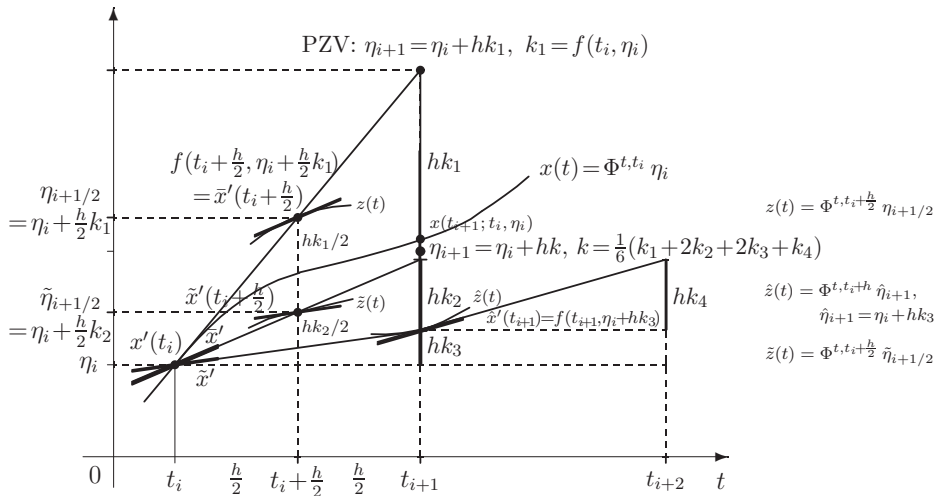


Abb. 5.3 Datei *krkv.pic*, Wahl des Anstiegs als Mittelwert der Anstiege von vier Trajektorien “links“, “Mitte“ und “rechts“ sowie Schritt beim KRKV

Übung 5.2 Zeigen Sie Folgendes.

- (a) Das 3-stufige Verfahren aus Beispiel 5.2 hat die Konsistenzordnung 2. Es ist somit ein (3,2)-RKV.
- (b) Im ESV ist die Ordnung $p = 4$ mit $s = 3$ Stufen nicht erreichbar.

5.2.1 Parameterschemata

In einer Übersicht stellen wir ausgewählte Parameterschemata bzw. Butcher-Tableaus zu expliziten RKV zusammen. Die allgemeine Form des Parameterschemas eines s -stufigen Verfahrens der Ordnung p ($s \geq p$), also des (s, p) -RKV ist

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^T \end{array} = \begin{array}{c|cccccc} c_1 = 0 & & & & & & \\ c_2 & a_{21} & & & & & \\ c_3 & a_{31} & a_{32} & & & & \\ c_4 & a_{41} & a_{42} & a_{43} & & & \\ \dots & \dots & \dots & \dots & \dots & & \\ c_s & a_{s,1} & a_{s,2} & a_{s,3} & \dots & a_{s,s-1} & \\ \hline 1 & b_1 & b_2 & b_3 & \dots & b_{s-1} & b_s \end{array} \quad (5.58)$$

Dabei lässt man den Koeffizienten $c_1 = 0$ oftmals weg.

Empfohlen werden meist solche Parameterschemata, wo die Koeffizienten a_{ij} und die RK-Gewichte b_i nicht negativ sind.

- (1) (1,1)-RKV, PZV (2) (2,2)-RKV, MPZV (3) (2,2)-RKV, HV (4) (2,2)-RKV

$$\begin{array}{c|c} c_1 = 0 & \\ \hline & 1 \end{array} \quad \begin{array}{c|cc} \frac{1}{2} & \frac{1}{2} \\ \hline & 0 & 1 \end{array} \quad \begin{array}{c|c} 1 & 1 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad \begin{array}{c|cc} \frac{2}{3} & \frac{2}{3} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

Wenn $f(t, x)$ nur von der Variablen t abhängt, so korrespondieren die Verfahren mit den Quadraturformeln von Newton-Côtes, (1) mit der Rechteckregel "links", (2) mit der Mittelpunktsregel ($|\gamma_2| < \frac{1}{2}ML^2$) und (3) mit der Trapezregel ($|\gamma_2| < \frac{2}{3}ML^2$). Das Verfahren (4) liefert von allen 2-stufigen Verfahren 2. Ordnung den kleinsten Fehler ($|\gamma_2| < \frac{1}{3}ML^2$).

- (5) (3,2)-RKV (6) (3,2)-RKV (7) (3,3)-RKV, opt. (8) (3,3)-Kutta-V.

$$\begin{array}{c|ccc} \frac{1}{2} & \frac{1}{2} \\ \hline 1 & \frac{1}{3} & \frac{2}{3} \\ \hline & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{array} \quad \begin{array}{c|ccc} \frac{1}{4} & \frac{1}{4} \\ \hline \frac{27}{40} & -\frac{189}{800} & \frac{729}{800} \\ \hline & \frac{214}{891} & \frac{27}{891} & \frac{650}{891} \end{array} \quad \begin{array}{c|ccc} \frac{1}{2} & \frac{1}{2} \\ \hline \frac{3}{4} & 0 & \frac{3}{4} \\ \hline & \frac{2}{9} & \frac{3}{9} & \frac{4}{9} \end{array} \quad \begin{array}{c|ccc} \frac{1}{2} & \frac{1}{2} \\ \hline 1 & -1 & 2 \\ \hline & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{array}$$

Das Verfahren (6) hat fast die Ordnung drei. Das Verfahren (7) liefert von allen 3-stufigen Verfahren 3. Ordnung den kleinsten Fehler ($|\gamma_3| < \frac{1}{9}ML^3$). Wenn $f(t, x)$ nur von der Variablen t abhängt, so korrespondiert das Verfahren (8) mit der Quadraturformel von Simpson ($|\gamma_3| < \frac{13}{36}ML^2$).

- (9) (3,3)-Heun (10) (3,3)-Nyström (11) (3,3)-RKV (12) (3,3)-RK-Fehlberg

$$\begin{array}{c|ccc} \frac{1}{3} & \frac{1}{3} \\ \hline \frac{2}{3} & 0 & \frac{2}{3} \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \quad \begin{array}{c|ccc} \frac{2}{3} & \frac{2}{3} \\ \hline \frac{2}{3} & 0 & \frac{2}{3} \\ \hline & \frac{2}{8} & \frac{3}{8} & \frac{3}{8} \end{array} \quad \begin{array}{c|ccc} \frac{1}{3} & \frac{1}{3} \\ \hline \frac{3}{4} & -\frac{3}{16} & \frac{15}{16} \\ \hline & \frac{1}{6} & \frac{3}{10} & \frac{8}{15} \end{array} \quad \begin{array}{c|ccc} 1 & 1 \\ \hline \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \hline & \frac{1}{6} & \frac{1}{6} & \frac{4}{6} \end{array}$$

(13) (3,3)-Kuntzmann-Formel, optimal

$$c_2 = \frac{10-2\sqrt{13}}{6} = 0.464816242, c_3 = \frac{1+\sqrt{13}}{6} = 0.767591879,$$

$$b_3 = \frac{3c_2-2}{6c_3(c_2-c_3)}, b_2 = \frac{1-2b_3c_3}{2c_2}, b_1 = 1 - b_2 - b_3, a_{21} = c_2, a_{32} = \frac{1}{6b_3c_2}, a_{31} = c_3 - a_{32}$$

c_2	$\frac{10-2\sqrt{13}}{6} = 0.464816242$		
c_3	$-\frac{5-\sqrt{13}}{24} = -0.058102030$	$\frac{3+\sqrt{13}}{8} = 0.825693909$	
	$\frac{23-5\sqrt{13}}{24} = 0.207176818$	$\frac{5+\sqrt{13}}{24} = 0.358564636$	$\frac{\sqrt{13}-1}{6} = 0.434258546$

(14) (4,4)-RKV, KRKV (15) (4,4)-RKV, 3/8-Formel (16) (4,4)-RKV, England-F.

$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$

$\frac{1}{3}$	$\frac{1}{3}$			
$\frac{2}{3}$	$-\frac{1}{3}$	1		
1	1	-1	1	
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$		
1	0	-1	2	
	$\frac{1}{6}$	0	$\frac{4}{6}$	$\frac{1}{6}$

(17) (4,4)-RKV, Kuntzmann-Formel (18) (4,4)-RKV, Ceschino-Formel

$\frac{2}{5}$	$\frac{2}{5}$			
$\frac{3}{5}$	$-\frac{3}{20}$	$\frac{3}{4}$		
1	$\frac{19}{44}$	$-\frac{15}{44}$	$\frac{40}{44}$	
	$\frac{55}{360}$	$\frac{125}{360}$	$\frac{125}{360}$	$\frac{55}{360}$

$\frac{1}{4}$	$\frac{1}{4}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	1	-2	2	
	$\frac{1}{6}$	0	$\frac{4}{6}$	$\frac{1}{6}$

(19) (4,4)-RKV, Gill-Modifikation, $c=1/\sqrt{2}$ (20) (4,2)-RKV, Gill-Modif. I, $c=1/\sqrt{2}$
 Koeff. a_{32} fehlerhaft, Ordnung = 2

$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	$c - \frac{1}{2}$	$1 - c$		
1	0	$-c$	$1 + c$	
	$\frac{1}{6}$	$\frac{1-c}{3}$	$\frac{1+c}{3}$	$\frac{1}{6}$

$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	$c - \frac{1}{2}$	$\frac{1}{2} - c$		
1	0	$-c$	$1 + c$	
	$\frac{1}{6}$	$\frac{1-c}{3}$	$\frac{1+c}{3}$	$\frac{1}{6}$

(21) (4,4)-RKV mit minimalem Diskretisierungsfehler

$$\alpha_2 = \frac{2}{5} = 0.4, \quad \alpha_3 = \frac{7}{8} - \frac{3}{16}\sqrt{5} = 0.455737254218789,$$

$$\beta = \frac{1-\alpha_2}{(\alpha_3-\alpha_2)(6\alpha_2\alpha_3-4(\alpha_2+\alpha_3)+3)} = 16.047205832726471$$

$\frac{2}{5}$	$\frac{2}{5}$			
$\frac{14-3\sqrt{5}}{16}$	$\alpha_3 - \frac{\alpha_3(\alpha_3-\alpha_2)}{2\alpha_2(1-2\alpha_2)}$	$\frac{\alpha_3(\alpha_3-\alpha_2)}{2\alpha_2(1-2\alpha_2)}$		
1	$1 - a_{42} - a_{43}$	$\frac{\beta(-1+\alpha_2+\alpha_3-(2\alpha_3-1)^2)}{2\alpha_2}$	$\frac{\beta(1-2\alpha_2)(1-\alpha_3)}{\alpha_3}$	
	$1 - C_2 - C_3 - C_4$	$\frac{2\alpha_3-1}{12\alpha_2(\alpha_3-\alpha_2)(1-\alpha_2)}$	$\frac{1-2\alpha_2}{12\alpha_3(\alpha_3-\alpha_2)(1-\alpha_3)}$	$\frac{1}{2} + \frac{2(\alpha_2+\alpha_3)-3}{12(1-\alpha_2)(1-\alpha_3)}$

$\frac{2}{5}$	0.4				
$\frac{14-3\sqrt{5}}{16}$	0.296977609247754	0.158759644971036			
1	0.218100388225920	-3.050965148692931	3.832864760467010		
	0.174760282262690	-0.551480662878733	1.205535599396524	0.171184781219519	

Dieses (4,4)-RKV genügt der Abschätzung $|\gamma_4| < 0.054627 ML^4$.

(22) (5,4)-RKV, Scraton-Formel (1964)

Ordnung $p = 5$ wird erreicht, falls für alle t entlang der Lösung $f_i(t, x) \neq 0$ gilt.

$\frac{2}{9}$	$\frac{2}{9}$				
$\frac{1}{3}$	$\frac{1}{12}$	$\frac{3}{12}$			
$\frac{3}{4}$	$\frac{69}{128}$	$-\frac{243}{128}$	$\frac{270}{128}$		
$\frac{9}{10}$	$-\frac{3105}{10000}$	$\frac{18225}{10000}$	$-\frac{11016}{10000}$	$\frac{4896}{10000}$	
	$\frac{1445}{13770}$	0	$\frac{6561}{13770}$	$\frac{3264}{13770}$	$\frac{2500}{13770}$

Wenn man in der Scraton-Formel von η_{i+1} den Wert

$$E_i = h \frac{(81k_1 - 243k_3 + 408k_4 - 250k_5)(95k_1 - 405k_2 + 342k_3 - 32k_4)}{183600(k_4 - k_1)}$$

abzieht, dann gilt

$$\eta_{i+1} - E_i = x(t_{i+1}) + \mathcal{O}(h^6).$$

Somit ist E_i der führende Fehleranteil von η_{i+1} und von der Ordnung h^5 .

Dies könnte man z.B. für eine Schrittweitensteuerung des Verfahrens verwenden.

(23) (6,5)-RKV, Butcher-Verfahren (1964)

Für $p \geq 5$ muss stets $s > p$ sein (Butcher 1964).

$\frac{1}{4}$	$\frac{1}{4}$					$\frac{1}{8}$	$\frac{1}{8}$					
$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$				$\frac{1}{4}$	0	$\frac{1}{4}$				
$\frac{1}{2}$	0	$-\frac{1}{2}$	1			$\frac{1}{2}$	$\frac{1}{2}$	-1	1			
$\frac{3}{4}$	$\frac{3}{16}$	0	0	$\frac{9}{16}$		$\frac{3}{4}$	$\frac{3}{16}$	0	0	$\frac{9}{16}$		
1	$-\frac{3}{7}$	$\frac{2}{7}$	$\frac{12}{7}$	$-\frac{12}{7}$	$\frac{8}{7}$	1	$-\frac{5}{7}$	$\frac{4}{7}$	$\frac{12}{7}$	$-\frac{12}{7}$	$\frac{8}{7}$	
	$\frac{7}{90}$	0	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$		$\frac{7}{90}$	0	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$	$\frac{7}{90}$

(24) (6,5)-RKV, Nyström-Formel (1925)

$$\begin{array}{r|rrrrrr}
 \frac{1}{3} & \frac{1}{3} & & & & & \\
 \frac{2}{5} & \frac{4}{25} & \frac{6}{25} & & & & \\
 1 & \frac{1}{4} & -\frac{12}{4} & \frac{15}{4} & & & \\
 \frac{2}{3} & \frac{6}{81} & \frac{90}{81} & -\frac{50}{81} & \frac{8}{81} & & \\
 \frac{4}{5} & \frac{6}{75} & \frac{36}{75} & \frac{10}{75} & \frac{8}{75} & 0 & \\
 \hline
 & \frac{23}{192} & 0 & \frac{125}{192} & 0 & -\frac{81}{192} & \frac{125}{192}
 \end{array}$$

(25) (6,5)-RKV, Sarafyan-Formel

$$\begin{array}{r|rrrrrr}
 \frac{1}{2} & \frac{1}{2} & & & & & \\
 \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & & & & \\
 1 & 0 & -1 & 2 & & & \\
 \frac{2}{3} & \frac{7}{27} & \frac{10}{27} & 0 & \frac{1}{27} & & \\
 \frac{1}{5} & \frac{28}{625} & -\frac{125}{625} & \frac{546}{625} & \frac{54}{625} & -\frac{378}{625} & \\
 \hline
 & \frac{14}{336} & 0 & 0 & \frac{35}{336} & \frac{162}{336} & \frac{125}{336}
 \end{array}$$

(26) (5,5)-RKV, Shanks-Formel

$$\begin{array}{r|rrrrrr}
 \frac{1}{9000} & \frac{1}{9000} & & & & & \\
 \frac{3}{10} & -\frac{4047}{10} & \frac{4050}{10} & & & & \\
 \frac{3}{4} & \frac{20241}{8} & -\frac{20250}{8} & \frac{15}{8} & & & \\
 1 & -\frac{931041}{81} & \frac{931500}{81} & -\frac{490}{81} & \frac{112}{81} & & \\
 \hline
 & \frac{105}{1134} & 0 & \frac{500}{1134} & \frac{448}{1134} & \frac{81}{1134} &
 \end{array}$$

(27) (6,6)-RKV, Shanks-Formel

$$\begin{array}{r|rrrrrrr}
 \frac{1}{300} & \frac{1}{300} & & & & & & \\
 \frac{1}{5} & -\frac{29}{5} & \frac{30}{5} & & & & & \\
 \frac{3}{5} & \frac{323}{5} & -66 & 2 & & & & \\
 \frac{14}{15} & -\frac{510104}{810} & \frac{521640}{810} & -\frac{12705}{810} & \frac{1925}{810} & & & \\
 1 & -\frac{417923}{77} & \frac{427350}{77} & -\frac{10695}{77} & \frac{1309}{77} & -\frac{54}{77} & & \\
 \hline
 & \frac{198}{3698} & 0 & \frac{1225}{3698} & \frac{1540}{3698} & \frac{810}{3698} & -\frac{77}{3698} &
 \end{array}$$

(28) (7,5)-RKV, Dormand-Prince-Verfahren

$\frac{1}{5}$	$\frac{1}{5}$							
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$						
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$					
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$				
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$			
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0	
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0	

(29) (7,6)-RKV, Butcher-Verfahren (1964)

$\frac{1}{3}$	$\frac{1}{3}$							
$\frac{2}{3}$	0	$\frac{2}{3}$						
$\frac{1}{3}$	$\frac{1}{12}$	$\frac{4}{12}$	$-\frac{1}{12}$					
$\frac{1}{2}$	$-\frac{1}{16}$	$\frac{18}{16}$	$-\frac{3}{16}$	$-\frac{6}{16}$				
$\frac{1}{2}$	0	$\frac{9}{8}$	$-\frac{3}{8}$	$-\frac{6}{8}$	$\frac{4}{8}$			
1	$\frac{9}{44}$	$-\frac{36}{44}$	$\frac{63}{44}$	$\frac{72}{44}$	0	$-\frac{64}{44}$		
	$\frac{11}{120}$	0	$\frac{81}{120}$	$\frac{81}{120}$	$-\frac{32}{120}$	$-\frac{32}{120}$	$\frac{11}{120}$	

(30) (8,6)-RKV, Huta-Formel (1956)

$\frac{1}{9}$	$\frac{1}{9}$							
$\frac{1}{6}$	$\frac{1}{24}$	$\frac{3}{24}$						
$\frac{1}{3}$	$\frac{1}{6}$	$-\frac{3}{6}$	$\frac{4}{6}$					
$\frac{1}{2}$	$-\frac{5}{8}$	$\frac{27}{8}$	$-\frac{24}{8}$	$\frac{48}{8}$				
$\frac{2}{3}$	$\frac{221}{9}$	$-\frac{981}{9}$	$\frac{867}{9}$	$-\frac{102}{9}$	$\frac{1}{9}$			
$\frac{5}{6}$	$-\frac{783}{48}$	$\frac{678}{48}$	$-\frac{472}{48}$	$-\frac{66}{48}$	$\frac{80}{48}$	$\frac{3}{48}$		
1	$\frac{761}{82}$	$-\frac{2079}{82}$	$\frac{1002}{82}$	$\frac{834}{82}$	$-\frac{454}{82}$	$-\frac{9}{82}$	$\frac{72}{82}$	
	$\frac{41}{840}$	$\frac{216}{840}$	0	$\frac{27}{840}$	$\frac{272}{840}$	$\frac{27}{840}$	$\frac{216}{840}$	$\frac{41}{840}$

(31) (6,5)-RKV, Luther-Verfahren, $f(t, x) = f(t)$

1	1				
1	$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{4}$	$\frac{14}{64}$	$\frac{5}{64}$	$-\frac{3}{64}$		
$\frac{1}{2}$	$-\frac{12}{96}$	$-\frac{12}{96}$	$\frac{8}{96}$	$\frac{64}{96}$	
$\frac{3}{4}$	0	$-\frac{9}{64}$	$\frac{5}{64}$	$\frac{16}{64}$	$\frac{36}{64}$
<hr/>					
	$\frac{7}{90}$	0	$\frac{7}{90}$	$\frac{32}{90}$	$\frac{12}{90}$
				$\frac{32}{90}$	$\frac{32}{90}$

1	1				
$\frac{1}{2}$	$\frac{3}{8}$	$\frac{1}{8}$			
1	$-\frac{1}{2}$	$-\frac{1}{2}$	2		
$\frac{1}{4}$	$\frac{4}{64}$	$-\frac{5}{64}$	$\frac{20}{64}$	$-\frac{3}{64}$	
$\frac{3}{4}$	$\frac{12}{64}$	$\frac{9}{64}$	$-\frac{12}{64}$	$\frac{7}{64}$	$\frac{32}{64}$
<hr/>					
	$\frac{7}{90}$	0	$\frac{12}{90}$	$\frac{7}{90}$	$\frac{32}{90}$
				$\frac{32}{90}$	$\frac{32}{90}$

(32) (6,5)-RKV, Fehlberg-Verfahren

$\frac{1}{6}$	$\frac{1}{6}$				
$\frac{4}{15}$	$\frac{4}{75}$	$\frac{16}{75}$			
$\frac{2}{3}$	$\frac{5}{6}$	$-\frac{8}{3}$	$\frac{5}{2}$		
$\frac{4}{5}$	$-\frac{8}{5}$	$\frac{144}{25}$	-4	$\frac{16}{25}$	
1	$\frac{361}{320}$	$-\frac{18}{5}$	$\frac{407}{128}$	$-\frac{11}{80}$	$\frac{55}{128}$
<hr/>					
	$\frac{31}{384}$	0	$\frac{1125}{2816}$	$\frac{9}{32}$	$\frac{125}{768}$
				$\frac{5}{66}$	

(33) (6,5)-RKV, Lawson-Verfahren

$\frac{1}{2}$	$\frac{1}{2}$				
$\frac{1}{4}$	$\frac{3}{16}$	$\frac{1}{16}$			
$\frac{1}{2}$	0	0	$\frac{1}{2}$		
$\frac{3}{4}$	0	$-\frac{3}{16}$	$\frac{6}{16}$	$\frac{9}{16}$	
1	$\frac{5}{7}$	$\frac{4}{7}$	$\frac{6}{7}$	$-\frac{12}{7}$	$\frac{8}{7}$
<hr/>					
	$\frac{7}{90}$	0	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$
				$\frac{7}{90}$	$\frac{7}{90}$

Weitere explizite ESV sind die Shanks-Formeln (7,7), (9,7), (10,8), (12,8) und Fehlberg-Formeln (8,7), (9,8).

Sie alle bieten sich für die näherungsweise Lösung von nicht steifen SysGDGL an.

Maple-Rechnungen mit Prozedur

```

Ordnungsbestimmung von s-stufigen RKV
Matrix      A = (a_ij), 1<=i,j<=s
Knoten      c = (c_1,c_2,...,c_s)^T
Gewichte    b = (b_1,b_2,...,b_s)^T
Butcher-Schema c | A
              -----
              1 | b^T
    
```

Grundbedingung: $c_i = \sum(a_{ij}, j=1..s) = a_i \cdot 1^T, i=1,2,\dots,s$

Ordnung 1: $s_1 = b_1 + b_2 + \dots + b_s = b^T \cdot 1 = 1$
 Ordnung 2: $s_2 = b_1 c_1 + b_2 c_2 + \dots + b_s c_s = b^T \cdot c = 1/2$
 Ordnung 3: $s_{31} = b_1 c_1^2 + b_2 c_2^2 + \dots + b_s c_s^2 = 1/3$
 $s_{32} = \sum(b_i a_{ij} c_j, i,j=1..s) = b^T \cdot A \cdot c = 1/6$
 Ordnung 4: $s_{41} = b_1 c_1^3 + b_2 c_2^3 + \dots + b_s c_s^3 = 1/4$
 $s_{42} = \sum(b_i c_i a_{ij} c_j, i,j=1..s) = 1/8$
 $s_{43} = \sum(b_i a_{ij} c_j^2, i,j=1..s) = 1/12$
 $s_{44} = \sum(b_i a_{ij} a_{jk} c_k, i,j,k=1..s) = b^T \cdot A^2 \cdot c = 1/24$

```
> ordnung := proc(s::posint,a,b,c)
  local s1,s2,s31,s32,s41,s42,s43,s44;
  s1 := sum(b[i], i=1..s);
  s2 := sum(b[i]*c[i], i=1..s);
  s31 := sum(b[i]*c[i]^2, i=1..s);
  s32 := sum(sum(b[i]*a[i,j]*c[j], j=1..s), i=1..s);
  s41 := sum(b[i]*c[i]^3, i=1..s);
  s42 := sum(sum(b[i]*c[i]*a[i,j]*c[j], j=1..s), i=1..s);
  s43 := sum(sum(b[i]*a[i,j]*c[j]^2, j=1..s), i=1..s);
  s44 := sum(sum(sum(b[i]*a[i,j]*a[j,k]*c[k], k=1..s), j=1..s), i=1..s);
  [s1, s2, [s31, s32], [s41, s42, s43, s44]]
end;
```

2-stufiges RKV 2. Ordnung, HV

```
> RKknoten := vector([0,1]);
  RKgewichte := vector([1/2,1/2]);
  RKmatrix := matrix([[0,0],[1,0]]);
> Heun := ordnung(2,RKmatrix,RKgewichte,RKknoten);
```

$$Heun := [1, \frac{1}{2}, [\frac{1}{2}, 0], [\frac{1}{2}, 0, 0, 0]]$$

Weitere ausgewählte Ergebnisse zu obigen (s, p) -RKV mit der Maple-Prozedur

Ordnung	1	2	3	4
Koeffizienten	[1, 1/2, [1/3, 1/6],			[1/4, 1/8, 1/12, 1/24]]

(3) Heun	[1, 1/2, [1/2, 0],			[1/2, 0, 0, 0]]
(5) (3,2)-RKV	[1, 1/2, [5/12, 1/9],			[3/8, 1/9, 1/18, 0]]
(6) (3,2)-RKV	[1, 1/2, [353/1056, 117/704],			[9497/42240, 3159/28160,
				117/2816, 0]]
Einige Parameter sind bzgl. der Ordnung 4 genau bzw. nahe der exakten.				
(7) (3,3)-RKV	[1, 1/2, [1/3, 1/6],			[11/48, 1/8, 1/12, 0]]
(8) Kutta	[1, 1/2, [1/3, 1/6],			[1/4, 1/6, 1/12, 0]]
(12) Fehlberg	[1, 1/2, [1/3, 1/6],			[1/4, 1/12, 1/6, 0]]
(13) Kuntzmann	[1, 1/2, [1/3, 1/6],			[0.232, 0.128, 0.077, 0]]
(14) KRKV	[1, 1/2, [1/3, 1/6],			[1/4, 1/8, 1/12, 1/24]]
(15)-(18) RKV	[1, 1/2, [1/3, 1/6],			[1/4, 1/8, 1/12, 1/24]]
(20) (4,2)-Gill	[1, 1/2, [1/3,			[1/4, 1/12-sqrt(2)/48,
	1/12-sqrt(2)/24]			1/24-sqrt(2)/48, -sqrt(2)/48]]

Beispiel 5.4 Es sollen mehrere s -stufige RKV p -ter Ordnung für das AWP $x' = tx$, $x(0) = 1$, $t \in [0, 5]$, gerechnet und verglichen werden.

Die MATLAB-Anweisungen und Ergebnisse sind wie folgt.

```
% rkv1.m
% Explizite s-stufige Runge-Kutta-Verfahren fuer gDGl
% mit Koeffiziententableaus A,b,c

function [t,x] = rkv1(A,b,c,f,t0,t1,x0,N)
    if N>0, h = (t1-t0)/N; else h = 0; end
    t(1) = t0;
    x(1) = x0;
    s = size(b,1);
    k = zeros(1,s);

    for i = 1:N
        t(i+1) = t0+i*h;
        k(1) = feval(f,t(i),x(i));
        for j = 2:s
            k(j) = feval(f,t(i)+h*c(j),x(i)+h*k(1:j-1)*A(j,1:j-1)');
        end;
        x(i+1) = x(i)+h*k*b;
    end;

% rkv.m
% Explizite s-stufige Runge-Kutta-Verfahren fuer SysgDGl
% mit Koeffiziententableaus A,b,c

function [t,x] = rkv(A,b,c,f,t0,T,x0,N)
    if N>0, h = (T-t0)/N; else h = 0; end
    n = size(x0,1);
    x = zeros(n,N+1);
    t(1) = t0;
    x(:,1) = x0;
    s = size(b,1);
    k = zeros(n,s);

    for i=1:N
        t(i+1) = t0+i*h;
        k(:,1) = feval(f,t(i),x(:,i));
        for j=2:s
            k(:,j) = feval(f,t(i)+h*c(j),x(:,i)+h*k(:,1:j-1)*A(j,1:j-1)');
        end;
        x(:,i+1) = x(:,i)+h*k*b;
    end;

% Rahmenprogramm
% r_rkv1.m
% Explizite s-stufige Runge-Kutta-Verfahren fuer gDGl
% mit Koeffiziententableaus A,b,c
% --> f8.m x'=tx, x(0)=1, x(t)=e^(t^2/2)

clear all
clc
format long
```



```

% Daten zum AWP
t0 = 0
t1 = 5
x0 = 1
N = 50 % 500, 5000

disp('Test')
[te,xe] = euler1('f8',t0,t1,x0,N);
eu = xe(N+1)
[tc,xc] = collatz1('f8',t0,t1,x0,N);
co = xc(N+1)
[th,xh] = heun1('f8',t0,t1,x0,N);
he = xh(N+1)

A = [0 0 0; 1/2 0 0; 1/3 2/3 0];
b = [1/3 1/3 1/3]';
c = [0 1/2 1]';

disp('(3,3)-RKV')
disp('rkv1')
[t,x] = rkv1(A,b,c,'f8',t0,t1,x0,N);
k3 = x(N+1)

A = [0 0 0 0; 1/2 0 0 0; 0 1/2 0 0; 0 0 1 0];
b = [1/6 1/3 1/3 1/6]';
c = [0 1/2 1/2 1]';

disp('KRKV')
disp('rkv1')
[t,x] = rkv1(A,b,c,'f8',t0,t1,x0,N);

disp('rkv1a')
[t,x] = rkv1a(A,b,c,'f8',t0,t1,x0,N);

disp('rkv1b -> Fehler')
[t,x] = rkv1b(A,b,c,'f8',t0,t1,x0,N);

disp('krkv1')
[t,x] = krkv1('f8',t0,t1,x0,N);
kl = x(N+1)

tt = 0:0.05:5;
xx = exp(0.5*tt.^2);
xxe = xx(max(size(xx)));

diary r_rkv1.pro
disp('Endwerte: eu,co,he,k3,kl, exakt')
disp(eu),disp(co),disp(he),disp(k3),disp(kl),disp(xxe)
disp('absolute Fehler')
disp(abs(eu-xxe)),disp(abs(co-xxe)),disp(abs(he-xxe)),...
disp(abs(k3-xxe)),disp(abs(kl-xxe))
disp('relative Fehler')
disp(abs((eu-xxe)/xxe)),disp(abs((co-xxe)/xxe)),disp(abs((he-xxe)/xxe)),...
disp(abs((k3-xxe)/xxe)),disp(abs((kl-xxe)/xxe))
diary off
pause

plot(tt,xx,'b-',t,x,'w:',te,xe,'r--',tc,xc,'y--',th,xh,'m--');
axis([0 5 0 300000]);

```

Rechnungen mit $N = 50, 500, 5000$

Wegen der stark wachsenden Lösungsfunktion werden die Fehler am Intervallende $t_1 = 5$ am größten sein.

Verfahren: PZV (Euler), MPZV (Collatz), VPZV (HV, Heun), (3,2)-RKV, KRKV.

Ein Vergleich der Fehler zwischen den Spalten mit Verkleinerung der Schrittweite auf jeweils ein Zehntel bestätigt die Fehlerordnung der ausgewählten ESV:

$\mathcal{O}(h)$ für PZV (Euler),

$\mathcal{O}(h^2)$ für MPZV (Collatz), VPZV (HV, Heun), (3,2)-RKV,

$\mathcal{O}(h^4)$ für KRKV.

N=50	N=500	N=5000

Endwerte bei t1=5: num=eu,co,he,k3,k1		
4.081299792895148e+004	2.136999698295539e+005	2.621638117623221e+005
2.159128551528072e+005	2.675790681261132e+005	2.683294838491181e+005
2.207581197691850e+005	2.676598006087716e+005	2.683303194229084e+005
2.489819295168908e+005	2.681157883042428e+005	2.683350523720851e+005
2.679281899526874e+005	2.683372303182735e+005	2.683372865150718e+005
exakt		
2.683372865208745e+005		
absolute Fehler num-xxe		
2.275242885919230e+005	5.463731669132054e+004	6.173474758552387e+003
5.242443136806722e+004	7.582183947613230e+002	7.80267175636254
4.757916675168942e+004	6.774859121029149e+002	6.96709796611685
1.935535700398360e+004	2.214982166317059e+002	2.23414878943004
4.090965681871166e+002	0.05620260100113	0.00000580266350
relative Fehler (num-xxe)/xxe		
0.84790411180603	0.20361432956158	0.02300639929171
0.19536767345223	0.00282561698597	0.00002907785145
0.17731105270004	0.00252475502338	0.00002596395773
0.07213070257561	0.00082544703162	0.00000832589767
0.00152456102352	0.00000020944760	0.00000000002162

5.3 Konvergenz von Einschrittverfahren

Bisher haben wir uns nur mit dem lokalen Diskretisierungsfehler befasst, also dem Fehler, der in jedem Schritt mit h neu dadurch entsteht, dass die zu approximierende Lösung eines AWP z.B. durch ein Geradenstück approximiert wird. Wir werden nun feststellen, dass dieser Fehler auch für die globale Abweichung der Näherung von der exakten Trajektorie maßgeblich ist.

Im Folgenden untersuchen wir im Intervall $[t_0, T]$ das AWP

$$x' = f(t, x), \quad x(t_0) = x_0. \quad (5.59)$$

Die exakte Lösung bezeichnen wir mit $x(t)$, also $x(t) = x(t; t_0, x_0) = \Phi^{t, t_0} x_0$.

Um einen Wert $x(t)$, $t \in [t_0, T]$, zu approximieren, kann mit Hilfe der Verfahrensfunktion Ψ ein ESV der Form (5.8) gewählt und auf eine Schrittweite $h = (t - t_0)/N$, $N \in \mathbb{N}$, angewandt werden. Untersucht werden soll, wie der globale Diskretisierungsfehler oder Fehler der Näherungslösung

$$e(t, h) = \eta_N - x(t), \quad h = (t - t_0)/N, \quad (5.60)$$

von der Schrittweite h bzw. von der Anzahl N der Teilschritte abhängt. Natürlich kann man damit auch an den Zwischenstellen $t_i = t_0 + ih$ den Fehler

$$e_i = e_i(h) = e(t_i, h) = \eta_i - x(t_i)$$

betrachten.

Wir bezeichnen das ESV als konvergent im Intervall $[t_0, T]$, falls für alle $t \in [t_0, T]$ gilt

$$\lim_{h \rightarrow 0} e(t, h) = 0 \quad \text{bzw.} \quad \lim_{N \rightarrow \infty} e(t, h) = 0. \quad (5.61)$$

Zunächst formulieren wir noch einmal Sätze, welche die Konvergenzordnung des PZV beinhalten. Dabei verwenden wir die Betrachtungen zum Diskretisierungsfehler im Abschnitt 4.3.

Satz 5.3 *Hat $f(t, x)$ im AWP (5.59) stetige partielle Ableitungen f_t und f_x für alle $t \in [t_0, T]$ und $x \in \mathbb{R}$, so ist das PZV konsistent mit der Ordnung 1, d.h. für den lokalen Diskretisierungsfehler (4.61) gilt*

$$\exists C > 0 \forall h, 0 < h < h_0 < T - t_0 \quad |\tau_{i+1}| \leq Ch, \quad i = 0, 1, 2, \dots \quad (5.62)$$

Beweis.

Mit $\eta_0 = x(t_0)$ ist $\tau_0 = 0$.

Sei $x(t)$ die Lösung des AWP (5.59). Weiter gilt

$$x''(t) = \frac{d}{dt} f(t, x(t)) = f_t(t, x(t)) + f_x(t, x(t)) f'(t, x(t)).$$

Wegen $f \in C^1([t_0, T] \times \mathbb{R})$ ist $x \in C^2[t_0, T]$ und damit

$$\exists M_2 > 0 \quad M_2 = \max_{t \in [t_0, T]} |x''(t)|.$$

Für den lokalen Diskretisierungsfehler erhält man durch Taylor-Reihenentwicklung der Lösung bei t_i

$$\begin{aligned} x(t_{i+1}) &= x(t_i) + hx'(t_i) + \frac{h^2}{2} x''(\xi), \quad \xi \in [t_i, t_{i+1}], \\ \tau_{i+1} &= \frac{1}{h} [x(t_{i+1}) - x(t_i)] - f(t_i, x(t_i)) \\ &= \frac{1}{h} [x(t_i) + hx'(t_i) + \frac{h^2}{2} x''(\xi) - x(t_i)] - f(t_i, x(t_i)) \\ &= x'(t_i) - f(t_i, x(t_i)) + \frac{h}{2} x''(\xi) \\ &= \frac{h}{2} x''(\xi), \\ |\tau_{i+1}| &\leq \frac{M_2}{2} h. \end{aligned}$$

Folglich ist $C = \frac{M_2}{2}$ und $\tau_{i+1} = \mathcal{O}(h)$. □

Damit ist auch der globale Diskretisierungsfehler e_i (4.63) von der Ordnung Eins.

Satz 5.4 *Ist $x \in C^2[t_0, T]$, d.h. $f(t, x)$ besitzt stetige partielle Ableitungen f_t und f_x in $[t_0, T] \times \mathbb{R}$, sowie*

$$\exists C > 0 \quad |e_0| \leq Ch, \quad h = (T - t_0)/N, \quad N > 0, \quad (5.63)$$

dann ist das PZV konvergent mit der Ordnung 1 und es gilt

$$|e_i| \leq \left(C + \frac{M_2}{2K} \right) h e^{K(t_i - t_0)}, \quad i = 0, 1, 2, \dots, N, \quad (5.64)$$

wobei $M_2 = \max_{t \in [t_0, T]} |x''(t)|$ und $K = \max_{t \in [t_0, T]} |f_x(t, x(t))|$.

Beweis. Der erste Konvergenzbeweis stammt von Cauchy (1840).

Sei $\tau = \max_{i=1, \dots, N} |\tau_i|$. Nach Satz 5.3 gilt

$$\tau \leq \frac{M_2}{2} h.$$

Aus den Beziehungen

$$\begin{aligned} \eta_{i+1} &= \eta_i + h f(t_i, \eta_i), \\ x(t_{i+1}) &= x(t_i) + h f(t_i, x(t_i)) + h \tau_{i+1}, \quad i = 0, 1, \dots, N, \end{aligned}$$

folgt durch Subtraktion

$$e_{i+1} = \eta_{i+1} - x(t_{i+1}) = e_i + h [f(t_i, \eta_i) - f(t_i, x(t_i))] - h \tau_{i+1}.$$

Mit dem Mittelwertsatz der Differentialrechnung ergibt sich

$$\begin{aligned} e_{i+1} &= \eta_{i+1} - x(t_{i+1}) = e_i + h f_x(t_i, \theta) [\eta_i - x(t_i)] - h\tau_{i+1}, \\ |e_{i+1}| &\leq (1 + Kh) |e_i| + h\tau, \quad i = 0, 1, \dots, N-1. \end{aligned}$$

Aus der letzten Ungleichung folgt rekursiv

$$\begin{aligned} |e_{i+1}| &\leq (1 + Kh)^2 |e_{i-1}| + [1 + (1 + Kh)] h\tau \\ &\leq (1 + Kh)^3 |e_{i-2}| + [1 + (1 + Kh) + (1 + Kh)^2] h\tau \\ &\leq \dots \\ &\leq (1 + Kh)^{i+1} |e_0| + h\tau \sum_{j=0}^i (1 + Kh)^j \\ &\leq (1 + Kh)^{i+1} |e_0| + h\tau \frac{(1 + Kh)^{i+1} - 1}{(1 + Kh) - 1} \\ &\leq (1 + Kh)^{i+1} |e_0| + \tau \frac{(1 + Kh)^{i+1} - 1}{K} \\ &\leq (1 + Kh)^{i+1} |e_0| + \tau \frac{(1 + Kh)^{i+1}}{K}, \\ |e_{i+1}| &\leq (1 + Kh)^{i+1} \left(|e_0| + \frac{\tau}{K} \right), \quad i = 0, 1, \dots, N-1, \\ |e_i| &\leq (1 + Kh)^i \left(|e_0| + \frac{M_2}{2K} h \right), \quad i = 1, 2, \dots, N. \end{aligned}$$

Wegen $1 + z \leq e^z$ für alle reellen Zahlen folgt $(1 + Kh)^i \leq e^{K(t_i - t_0)}$ und mit (5.63)

$$|e_i| \leq \left(|e_0| + \frac{M_2}{2K} h \right) e^{K(t_i - t_0)} \leq \left(C + \frac{M_2}{2K} \right) h e^{K(t_i - t_0)}, \quad i = 1, 2, \dots, N.$$

Für $|e_0|$ trifft die letzte Abschätzung ebenfalls zu. □

Bemerkung 5.1

(a) Falls $e_0 = 0$ ist, was oft vorausgesetzt bzw. garantiert wird, hat man auch die Konvergenzordnung 1 und

$$|e_i| \leq \frac{M_2}{2K} h e^{K(t_i - t_0)}, \quad i = 1, 2, \dots, N.$$

(b) Die Größe K kann auch die Lipschitz-Konstante der Funktion $f(t, x)$ bzgl. x sein, was eine etwas schwächere Voraussetzung bedeuten würde.

(c) Die Ungleichungskette für die Abschätzung von $|e_{i+1}|$ kann im Fall $e_0 = 0$ auch notiert werden als

$$|e_i| \leq \frac{M_2}{2K} h (e^{K(t_i - t_0)} - 1), \quad i = 0, 1, 2, \dots, N.$$

(d) Wegen $e_i = \mathcal{O}(h)$ ist für hinreichend kleine Schrittweiten h eine Halbierung des globalen Diskretisierungsfehlers zu erwarten, wenn h halbiert wird.

Dass die Schranke (5.64) für $|e_i|$ sehr grob ist und deshalb nur theoretische Bedeutung hat, möge das folgende Beispiel verdeutlichen.

Beispiel 5.5 Das AWP

$$x'(t) = -x(t), \quad x(0) = 1, \quad t \in [t_0, T] = [0, 5], \quad (5.65)$$

besitzt die Lösung $x(t) = e^{-t}$.

Die Näherungslösung η_i , $i = 0(1)N$, berechnen wir mittels PZV mit den Parametern $N = 320$, $h = \frac{T-t_0}{N} = \frac{1}{64}$.

Wegen

$$M_2 = \max_{t \in [t_0, T]} |x''(t)| = \max_{t \in [0, 5]} e^{-t} = 1$$

und

$$K = \max_{t \in [t_0, T]} |f_x(t, x(t))| = \max_{t \in [0, 5]} 1 = 1$$

lautet die Fehlerschranke (5.64) mit $C = 0$

$$|e_i| \leq \frac{M_2}{2K} h e^{K(t_i - t_0)} = \frac{1}{2} h e^{ih}, \quad i = 0, 1, 2, \dots, N.$$

Sie ist relativ grob.

In der Tabelle 5.1 werden die Näherungswerte η_i des PZV an den Stellen $t = 0(1)5$ gelistet und zum Vergleich herangezogen.

i	t_i	η_i	$e_i = \eta_i - x(t_i)$	Schranke
0	0	1	0	0.007813
64	1	0.364987	-0.002893	0.021237
128	2	0.133215	-0.002120	0.057727
192	3	0.048622	-0.001165	0.156918
256	4	0.017746	-0.000570	0.426548
320	5	0.006477	-0.000261	1.159478
		$x(5) = 0.006738$		

Tab. 5.1 Näherungswerte η_i des PZV mit $h = \frac{1}{64}$ im Vergleich zu $x(t_i)$, Fehler und Schranke (Dezimalzahlen gerundet)

Wäre $x(5) = e^{-5} = 0.006738$ mit einer Genauigkeit von 10^{-3} zu bestimmen, so erhielte man nach der Fehlerschranke für die Schrittweite

$$\frac{1}{2} h e^5 \approx 10^{-3}, \quad \text{d.h.} \quad h \approx \frac{1}{74000}.$$

Man erreicht die geforderte Genauigkeit jedoch bereits mit $h \approx \frac{1}{64}$, wobei auch bei allen anderen Stellen die Fehler bei der Größenordnung $\mathcal{O}(10^{-3})$ liegen.

Die Rechnung mit der Schrittweite $h = \frac{1}{74000}$ ergibt in der vierten Spalte der Tabelle 5.1 den Fehlerverlauf $[0, -2.485e-6, -1.828e-6, -1.009e-6, -4.950e-7, -2.276e-7]$.

Das folgende allgemeinere Ergebnis zeigt ebenfalls den globalen Fehler in Relation zum lokalen Diskretisierungsfehler.

Satz 5.5 Die Funktion Ψ sei stetig und erfülle die Lipschitz-Bedingung

$$|\Psi(t, x_1, h) - \Psi(t, x_2, h)| \leq M |x_1 - x_2| \quad (5.66)$$

mit einer Konstanten $M \geq 0$.

Dann gilt für das durch Ψ definierte ESV (5.8):

Ist das ESV ein Verfahren der Konsistenzordnung p (vergl. Def. 5.1 (d)), so erfüllt der globale Diskretisierungsfehler mit $h = (t - t_0)/N$ und mit einem geeigneten $C > 0$ eine Abschätzung der Form

$$|e(t, h)| \leq C h^p \frac{e^{M(t-t_0)} - 1}{M}. \quad (5.67)$$

Bevor dieser Satz bewiesen wird, soll seine Aussage kurz kommentiert werden.

Bemerkung 5.2

(a) Da ein Konvergenzsatz immer mit hinreichend kleinen Schrittweiten h bzw. $h \rightarrow 0$ operiert, werden diese so klein sein, dass indirekt auch die Schrittweitenbeschränkung $h < h_{max}$ für ein stabiles Lösungsverhalten eingehalten wird. Dabei muss hier die Schranke h_{max} nicht explizit angegeben werden.

(b) Die Bedingung (5.66) lässt sich in den bisher vorgestellten Verfahren leicht aus der Lipschitz-Stetigkeit (4.18) der rechten Seite $f(t, x)$ ableiten, welche wir ja immer vorausgesetzt haben. Beispielsweise folgt im skalaren Fall

$$|f(t, x_1) - f(t, x_2)| \leq L |x_1 - x_2|$$

und für das HV (5.6) mit

$$\Psi(t, x, h) = \frac{1}{2}[f(t, x) + f(t + h, x + hf(t, x))]$$

die Abschätzung

$$\begin{aligned} |\Psi(t, x_1, h) - \Psi(t, x_2, h)| &\leq \frac{1}{2}|f(t, x_1) - f(t, x_2)| \\ &\quad + \frac{1}{2}|f(t + h, x_1 + hf(t, x_1)) - f(t + h, x_2 + hf(t, x_2))| \\ &\leq \frac{1}{2}L|x_1 - x_2| + \frac{1}{2}L(|x_1 - x_2| + h|f(t, x_1) - f(t, x_2)|) \\ &\leq L\left(1 + \frac{1}{2}hL\right) |x_1 - x_2|, \quad M = L\left(1 + \frac{1}{2}hL\right). \end{aligned}$$

(c) Der globale Fehler ist wie der lokale Fehler von der Ordnung $\mathcal{O}(h^p)$.

Zu beachten ist allerdings, dass die rechte Seite der Abschätzung (5.67) mit zunehmendem Abstand vom Anfangspunkt t_0 exponentiell wächst.

Zum Beweis des Satzes 5.5 benötigen wir das folgende Hilfsergebnis.

Lemma 5.6 *Genügt eine Zahlenfolge ξ_i , $i = 0, 1, 2, \dots$, einer Abschätzung der Form*

$$|\xi_{i+1}| \leq (1 + \delta)|\xi_i| + B \quad (5.68)$$

mit den Konstanten $\delta > 0$ und $B \geq 0$, so gilt für alle $k \in \mathbb{N}$

$$|\xi_k| \leq e^{k\delta} |\xi_0| + B \frac{e^{k\delta} - 1}{\delta}. \quad (5.69)$$

Beweis. Lemma 5.6

Die iterative Anwendung der Abschätzung (5.68) ergibt mittels Induktion

$$\begin{aligned} |\xi_1| &\leq (1 + \delta)|\xi_0| + B, \\ |\xi_2| &\leq (1 + \delta)|\xi_1| + B \leq (1 + \delta)^2 |\xi_0| + B(1 + (1 + \delta)), \\ |\xi_3| &\leq (1 + \delta)|\xi_2| + B \leq (1 + \delta)^3 |\xi_0| + B(1 + (1 + \delta) + (1 + \delta)^2), \dots, \\ |\xi_k| &\leq (1 + \delta)^k |\xi_0| + B \cdot (1 + (1 + \delta) + \dots + (1 + \delta)^{k-1}) \\ &= (1 + \delta)^k |\xi_0| + B \frac{(1 + \delta)^k - 1}{\delta} \\ &\leq e^{k\delta} |\xi_0| + B \frac{e^{k\delta} - 1}{\delta}. \end{aligned}$$

Letztere Ungleichung folgt wegen $1 + \delta \leq e^\delta$, $\delta > 0$. □

Beweis. Satz 5.5

Mit $h = (t - t_0)/N$ definieren wir die Knoten $t_i = t_0 + ih$, die exakten Werte an den Knoten $x_i = x(t_i)$, sowie die globalen Diskretisierungsfehler an den Knoten $e_i = \eta_i - x_i$. Aus der rekursiven Definition (5.8) für η_{i+1} und der Beziehung $x_{i+1} = \Phi^{t_{i+1}, t_i} x_i$ folgt durch geschicktes Umordnen

$$\begin{aligned} e_{i+1} &= \eta_{i+1} - x_{i+1} \\ &= \eta_i + h\Psi(t_i, \eta_i, h) - \Phi^{t_{i+1}, t_i} x_i \\ &= (\eta_i - x_i) + h(\Psi(t_i, \eta_i, h) - \Psi(t_i, x_i, h)) + h\left(\Psi(t_i, x_i, h) - \frac{\Phi^{t_{i+1}, t_i} x_i - x_i}{h}\right). \end{aligned}$$

Nach der Lipschitz-Bedingung (5.66) ist

$$|\Psi(t_i, \eta_i, h) - \Psi(t_i, x_i, h)| \leq M |\eta_i - x_i| = M |e_i|.$$

Wegen der Konsistenzordnung p des ESV (vergl. (5.10), (5.13)) gilt

$$\left| \Psi(t_i, x_i, h) - \frac{\Phi^{t_{i+1}, t_i} x_i - x_i}{h} \right| \leq C h^p$$

mit einem geeigneten C . Hieraus ergibt sich für die Folge e_i die Abschätzung

$$|e_{i+1}| \leq (1 + Mh) |e_i| + C h^{p+1}. \quad (5.70)$$

Wegen Lemma 5.6 folgt hieraus zunächst

$$\begin{aligned} |e_N| &\leq e^{M N h} |e_0| + C h^{p+1} \frac{e^{M N h} - 1}{M h} \\ &\leq e^{M(t-t_0)} |e_0| + C h^p \frac{e^{M(t-t_0)} - 1}{M}, \quad t = t_N. \end{aligned} \quad (5.71)$$

Bei genauer Beachtung der AB ist $e_0 = 0$ und man erhält die Abschätzung (5.67). \square

Bemerkung 5.3 (a) Um in jedem Fall die Konvergenz zu sichern, muss bzgl. der AB

$$\lim_{h \rightarrow 0} e_0 = 0 \text{ bzw. } |e_0| \leq C_1 h^q, \quad q > 0, \quad C_1 \geq 0, \quad (5.72)$$

gelten. Meistens macht man jedoch bei der Approximation der AB keine Fehler.

(b) Falls $|e_0| \leq C_1 h^q$ ist, und insbesondere bei $e_0 = 0$, überträgt sich die Konsistenzordnung p auf die Konvergenz und damit auf die Genauigkeit der Lösung.

Die Konvergenzordnung ist dann ebenfalls p .

(c) Für s -stufige ESV der Ordnung p ist die Notation (s, p) -ESV gebräuchlich.

Damit sind PZV=(1,1)-RKV, MPZV=(2,2)-RKV, HV=(2,2)-RKV, KRKV=(4,4)-RKV.

Beispiel 5.6 Wir testen mehrere Verfahren unterschiedlicher Konvergenzordnungen, das PZV, MPZV, HV, 3-stufige RKV 2. Ordnung (5.18) und KRKV, für das AWP

$$x'(t) = t x(t), \quad x(0) = 1. \quad (5.73)$$

Die exakte Lösung ist gegeben durch $x(t) = e^{t^2/2}$. Diese stark wachsende Funktion ist auf dem Intervall $[0, +\infty)$ definiert, mehr noch, sie ist global.

ESV für Näherungslösungen werden auch Folgen von wachsenden Werten η_i erzeugen und sich für immer größer werdendes t_i von der exakten Lösung langsam entfernen. Nur in einem endlichen Intervall kann bei einer Schrittweite $h \rightarrow 0$ das ESV mit der entsprechenden Ordnung konvergieren.

Zur Berechnung des Funktionswertes $x(5) = 2.683\,372\,865\,208\dots e+05$ wird das Intervall $[0, T] = [0, 5]$ in $N = 50, 500$ und 5000 gleiche Teilintervalle unterteilt.

In der Tabelle 5.2 sind die Näherungswerte η_N der ESV gelistet.

RKV	$h = 0.1$	$h = 0.01$	$h = 0.001$
	η_N	η_N	η_N
PZV	4.081 299e+04	2.136 999e+05	2.621 638e+05
MPZV	2.159 128e+05	2.675 790e+05	2.683 294e+05
HV	2.207 581e+05	2.676 598e+05	2.683 303e+05
(3,2)-RKV	2.489 819e+05	2.681 157e+05	2.683 350e+05
KRKV	2.679 281e+05	2.683 372 303e+05	2.683 372 865 150e+05

Tab. 5.2 Näherungswerte η_N der ESV im Vergleich zu $x(5) = 2.683\,372\,865\,208e+05$ (Dezimalzahlen abgeschnitten)

Die globalen Fehler $|\eta_N - x(5)|$ sind in der Tabelle 5.3 dargestellt. Daneben sind die relativen Fehler

$$\left| \frac{\eta_N - x(5)}{x(5)} \right|$$

zu finden, die angeben, auf wie viele führende Stellen das numerische Ergebnis exakt ist.

RKV	$h = 0.1$		$h = 0.01$		$h = 0.001$	
	Fehler	rel. Fehler	Fehler	rel. Fehler	Fehler	rel. Fehler
PZV	2.275e+05	8.479e-01	5.463e+04	2.036e-01	6.173e+03	2.300e-02
MPZV	5.242e+04	1.953e-01	7.582e+02	2.825e-03	7.802e+00	2.907e-05
HV	4.757e+04	1.177e-01	6.774e+02	2.524e-03	6.967e+00	2.596e-05
(3,2)-RKV	1.935e+04	7.213e-02	2.214e+02	8.254e-04	2.234e+00	8.325e-06
KRKV	4.090e+02	1.524e-03	5.620e-02	2.094e-07	5.802e-06	2.162e-11

Tab. 5.3 Fehler bei numerischer Berechnung zu $x(5) = 2.683\,372\,865\,208e+05$ (Dezimalzahlen abgeschnitten)

Es zeigt sich, dass das KRKV bzgl. der Fehlergüte die anderen deutlich übertrifft und ohne großen Aufwand mit vier Funktionswertberechnungen pro Schritt eine Genauigkeit liefert, welche nahe an rechnerinterne Darstellungsgenauigkeiten reeller Zahlen (GKA *double*) kommt.

Wir rechnen noch das KRKV mit $N = 5\,000 \cdot 2^k$, $k = 0, 1, \dots, 6$, und vergleichen dann η_N mit $x(5) = e^{25/2}$.

x(5)	=	2.68337 28652 08745e+05		
eta(5000)	=	2.68337 28651 50718e+05	eta()-x(5)	= 0.0000 0580 27
eta(10000)	=	2.68337 28652 05117e+05	eta()-x(5)	= 0.0000 0036 28
eta(20000)	=	2.68337 28652 08528e+05	eta()-x(5)	= 0.0000 0002 17
eta(40000)	=	2.68337 28652 08747e+05	eta()-x(5)	= 0.0000 0000 02
eta(80000)	=	2.68337 28652 08786e+05	eta()-x(5)	= 0.0000 0000 41
eta(160000)	=	2.68337 28652 08765e+05	eta()-x(5)	= 0.0000 0000 20
eta(320000)	=	2.68337 28652 08852e+05	eta()-x(5)	= 0.0000 0001 07

Ein hohe Anzahl von Teilintervallen führt zur Akkumulation von Rundungsfehlern.

Die Einbeziehung von Rundungsfehlern in die Betrachtung führt auf die Verallgemeinerung von Satz 5.5.

Satz 5.7 *Unter den gleichen Voraussetzungen wie im Satz 5.5 sowie bei Einbeziehung von Rundungsfehlern δ_i in jedem Schritt des ESV gilt die Abschätzung*

$$|\tilde{e}(t, h)| = |\tilde{\eta}_N - x(t)| \leq \frac{e^{M(t-t_0)} - 1}{M} \left(C h^p + \frac{\delta}{h} \right), \quad t = t_N, \tag{5.74}$$

wobei $\delta = \max_{i=0, \dots, N} |\delta_i|$.

Verschiedene Aspekte sind also bei der Schrittweitenwahl zu berücksichtigen.

Die Abb. 5.4 gibt eine qualitative Darstellung des Gesamtfehlers im Fall einer äquidistanten Schrittweite. Bei Vergrößerung der Schrittweite verringert sich der Aufwand, möglicherweise steigt aber der globale Fehler an. Zu kleine Schrittweiten dagegen bedeuten mehr Aufwand und größeren Einfluss von Rundungsfehlern. Insbesondere gibt es auch eine untere Schranke, welche die Schrittweite nicht unterschreiten sollte.

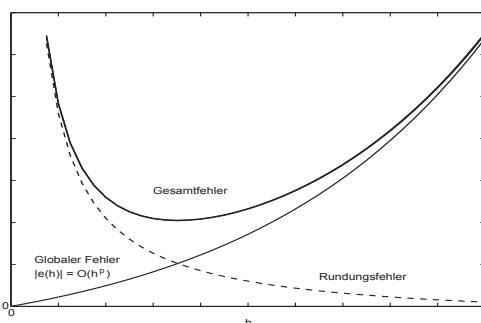


Abb. 5.4

Datei *rund1.ps*

Verhalten

des globalen Fehlers

und Rundungsfehlers

in Abhängigkeit von h

So einfach und illustrativ das PZV ist, so schnell sind auch die Grenzen dieses ESV zu erkennen. Konvergenzbetrachtungen beinhalten den Fall, was bei $h \rightarrow 0$ passiert. Kleine Verfahrensfehler können also nur mit der Bedingung $h \ll 1$ erwartet werden.

In Beispiel 4.2 (a) wurde der andere Aspekt der Beschränkung der Schrittweite h schon diskutiert. Die Erkenntnisse sollen durch ein einfaches AWP vertieft werden.

Beispiel 5.7 Das AWP

$$x'(t) = f(t, x(t)) = -10x(t), \quad x(0) = 1, \quad t \in [t_0, T] = [0, 1], \quad (5.75)$$

hat die Lösung $x(t) = e^{-10t}$. Damit bezeichnet man die gDGl auch als inhärent stabil.

Mit dem PZV und der Schrittweite $h = 0.2$ sollen die Näherungslösung η_i , $i = 0, 1, \dots$, sowie die Ableitungswerte η'_i berechnet werden.

i	t_i	η_i	η'_i	$x(t_i)$
0	0	1	-10	1
1	0.2	-1	10	0.135335
2	0.4	1	-10	0.018316
3	0.6	-1	10	0.002479

Tab. 5.4

Näherungswerte η_i , η'_i

des PZV im Vergleich zu $x(t_i)$

Die gDGl hat die charakteristische Gleichung $\lambda + 10 = 0$, sodass $\lambda = \Re \lambda = -10 < 0$ ist.

Das PZV

$$\eta_{i+1} = \eta_i + h f(t_i, \eta_i) = (1 - 10h) \eta_i, \quad i = 0, 1, \dots, \quad \eta_0 = 1,$$

hat mit $h = 0.2$ die einfache Gestalt $\eta_{i+1} = -\eta_i$, d.h. $\eta_{i+1} = (-1)^{i+1}$.

Mit einer kleineren Schrittweite h , z.B. $h < 0.1$, können akzeptable Ergebnisse erzielt werden. Dies ist Gegenstand von Stabilitätsuntersuchungen.

5.4 Schrittweitensteuerung

Bei der Wahl einer geeigneten Schrittweite h muss ein Kompromiss zwischen hoher Rechengenauigkeit und moderatem Rechenaufwand gefunden werden. Hinzu kommt, dass sich die "ideale" oder "optimale" Schrittweite im Verlauf der Integration des AWP ändern kann. Das Ziel ist eine Schrittweitenkontrolle bzw. eine adaptive Schrittweitensteuerung (adaptive Gittersteuerung, SWS), welche im Verlauf der Rechnungen die Schrittweite korrigiert und optimiert. Einige Möglichkeiten sollen aufgezeigt werden.

5.4.1 Schrittweitenkontrolle mittels lokalen Diskretisierungsfehler

Die Messung des lokalen Diskretisierungsfehlers geschieht durch die Auswertung eines anderen geeigneten ESV, des sogenannten Kontrollverfahrens, welches eine höhere Ordnung als das ursprüngliche Verfahren hat.

Gegeben seien ein ESV und ein Kontrollverfahren zu einem AWP für eine gDGL durch

$$\eta_{i+1} = \eta_i + h \Psi(t_i, \eta_i, h), \quad (5.76)$$

$$\hat{\eta}_{i+1} = \eta_i + h \hat{\Psi}(t_i, \eta_i, h). \quad (5.77)$$

Sei $z(t) = \phi^{t, t_i} \eta_i$. Haben das ESV den lokalen Diskretisierungsfehler (Konsistenzordnung) p und das Kontrollverfahren die Ordnung $p+1$, so ist

$$\begin{aligned} \frac{1}{h} |\eta_{i+1} - \hat{\eta}_{i+1}| &= \frac{1}{h} |\eta_{i+1} - z(t_{i+1}) + z(t_{i+1}) - \hat{\eta}_{i+1}| \\ &\leq \frac{1}{h} |\eta_{i+1} - z(t_{i+1})| + \frac{1}{h} |\hat{\eta}_{i+1} - z(t_{i+1})| \\ &\leq C h^p + \hat{C} h^{p+1} \approx C h^p \end{aligned} \quad (5.78)$$

und daher

$$\frac{1}{h} |\eta_{i+1} - \hat{\eta}_{i+1}| \approx \frac{1}{h} |\eta_{i+1} - z(t_{i+1})| = |\tau(t_i, \eta_i, h)|. \quad (5.79)$$

Dies ermöglicht die Schätzung des lokalen Diskretisierungsfehlers mit Hilfe des Kontrollverfahrens.

Der Einsatz von Kontrollverfahren sollte möglichst ohne zusätzlichen Rechenaufwand erfolgen. Ist beispielsweise η_{i+1} gegeben durch ein s -stufiges RKV nach Berechnung der Steigungen k_1, k_2, \dots, k_s , so sollte das Kontrollverfahren die Werte k_1, k_2, \dots, k_s einbeziehen und ansonsten ohne weitere Werte auskommen. Da aber RKV in der Regel so konstruiert sind, dass ohne weitere Funktionsauswertungen eine Ordnungssteigerung nicht möglich ist, bedient man sich z.B. des Fehlberg-Tricks. Dabei bezieht man den Wert

$$\eta_{i+1} = \eta_i + h \sum_{i=1}^s b_i k_i \quad (5.80)$$

mit ein, welcher ohnehin bei akzeptierter Schrittweite berechnet werden muss.

Nun soll das KRKV der Ordnung $p = 4$ als Kontrollverfahren zur Steuerung eines Verfahrens 3. Ordnung benutzt werden. Das KRKV ist gegeben durch

$$\mathbf{b} = \frac{1}{6} \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (5.81)$$

mit der Berechnungsvorschrift

$$\eta_{i+1} = \eta_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4). \quad (5.82)$$

Versuch 1

Versuchen wir, ein Verfahren 3. Ordnung mit \mathbf{c} und A , aber neuen Gewichten $\hat{b}_1, \dots, \hat{b}_4$ zu erzeugen. Aus den Ordnungsbedingungen ergibt sich nach Satz 5.2 das Gleichungssystem

$$\begin{aligned} \hat{\mathbf{b}}^T \mathbf{1} &= \sum_{i=1}^s \hat{b}_i = 1, \\ \hat{\mathbf{b}}^T \mathbf{c} &= \sum_{i=1}^s \hat{b}_i c_i = \frac{1}{2}, \\ \sum_{i=1}^s \hat{b}_i c_i^2 &= \frac{1}{3}, \quad \hat{\mathbf{b}}^T A \mathbf{c} = \sum_{i,j=1}^s \hat{b}_i a_{ij} c_j = \frac{1}{6}, \end{aligned}$$

also

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ 0 & \frac{1}{4} & \frac{1}{4} & 1 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \\ \hat{b}_4 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \frac{1}{6} \end{pmatrix}. \quad (5.83)$$

Dieses Gleichungssystem ist regulär und hat die eindeutige Lösung $\hat{\mathbf{b}} = \mathbf{b} = \frac{1}{6}(1, 2, 2, 1)^T$, sodass kein neues Verfahren 3. Ordnung entsteht.

Versuch 2 Fehlberg-Trick

Wir definieren im Parameterschema des KRKV eine zusätzliche Zeile $c_5, a_{51}, a_{52}, a_{53}, a_{54}, a_{55} = 0$, den Steigungswert

$$k_5 = f(t_i + h, \eta_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)), \quad (5.84)$$

sodass $c_5 = 1$ und $(a_{51}, a_{52}, a_{53}, a_{54}) = \frac{1}{6}(1, 2, 2, 1)$ sind, und führen die Gewichte $\hat{\mathbf{b}} = (\hat{b}_1, \hat{b}_2, \dots, \hat{b}_5)^T$ ein. Das unterbestimmte Gleichungssystem für $\hat{\mathbf{b}}$, welches auf die Ordnung 3 führt, ist nun

$$\tilde{A} \hat{\mathbf{b}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 & 1 \\ 0 & \frac{1}{4} & \frac{1}{4} & 1 & 1 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \\ \hat{b}_4 \\ \hat{b}_5 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \frac{1}{6} \end{pmatrix}. \quad (5.85)$$

Seine allgemeine parameterabhängige Lösung lautet

$$\hat{\mathbf{b}} = \frac{1}{6}(1, 2, 2, 1 - 6u, 6u), \quad u \in \mathbb{R}, \quad (5.86)$$

und es ist

$$\text{kern}(\tilde{A}) = \{\mathbf{z} : \tilde{A}\mathbf{z} = \mathbf{0}\} = \text{span}(0, 0, 0, 1, -1)^T. \quad (5.87)$$

Eine spezielle Lösung bei $u = 0$ ist

$$\hat{\mathbf{b}} = \frac{1}{6}(1, 2, 2, 1, 0)^T, \quad (5.88)$$

und sie führt wieder auf das KRKV.

Verfahren 3. Ordnung sind gegeben z.B. mit $u = \frac{1}{6}$ und $u = \frac{1}{12}$ durch

$$\hat{\mathbf{b}} = \frac{1}{6}(1, 2, 2, 0, 1)^T, \quad (5.89)$$

$$\hat{\mathbf{b}} = \frac{1}{6}(1, 2, 2, \frac{1}{2}, \frac{1}{2})^T. \quad (5.90)$$

Wenn man bei der praktischen Realisierung die SWS mit

$$|\eta_{i+1} - \hat{\eta}_{i+1}| \approx h|\tau(t_i, \eta_i, h)| = \mathcal{O}(h^{p+1})$$

macht, besteht natürlich auch die Möglichkeit, die genaueren Werte η_{i+1} und damit das KRKV für den Fortgang der Rechnung zu verwenden. Das würde bedeuten, dass man zum KRKV der Ordnung 4 seine SWS durch den Vergleich mit einem RKV niedrigerer Ordnung erhält. Insgesamt geht die Idee auf E. Fehlberg zurück und ist in den eingebetteten Runge-Kutta-Fehlberg-Verfahren (RKFFV) realisiert.

Beispiel 5.8 (a) Gegeben sei das AWP

$$x' = f(t, x) = t \cos(x), \quad x(-2) = 1, \quad t \in [t_0, T] = [-2, 5]. \quad (5.91)$$

Die exakte Lösung ist

$$x(t) = \arctan \left(\frac{(1 + \sin(1)) e^{t^2-4} - 1 + \sin(1)}{(1 + \sin(1)) e^{t^2-4} + 1 - \sin(1)}, \frac{2 \cos(1) e^{t^2/2-2}}{(1 + \sin(1)) e^{t^2-4} + 1 - \sin(1)} \right), \quad (5.92)$$

wobei

$$\arctan(y, z) = -i \ln \left(\frac{z + iy}{\sqrt{z^2 + y^2}} \right), \quad i = \sqrt{-1}.$$

Zunächst interessieren wir uns für das Richtungsfeld zur gDGl und AWP mit ausgewählten AB. Wir nutzen das Maple-Tool `DEplot` mit dem darin implementierten KRKV. Zusätzlich ist in der Abbildung 5.5 die exakte Lösung des AWP (5.91) als gestrichelte Kurve eingetragen. Seine Näherungslösung mit dem KRKV liegt nur knapp daneben.

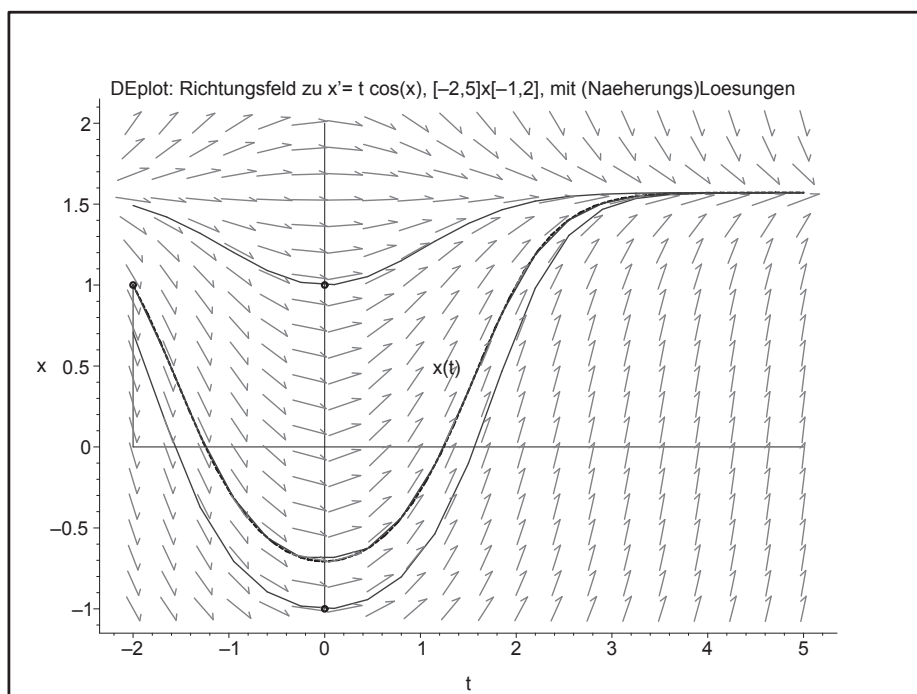


Abb. 5.5 Datei *richt8.ps*,

Richtungsfeld zur gDGl $x' = f(t, x) = t \cos(x)$ sowie Näherungslösungen zu den AB $x(-2) = 1$, $x(0) = -1$, $x(0) = 1$, sowie $x(t)$ mit AB $x(-2) = 1$

Wir zeichnen das Richtungsfeld mit MATLAB und tragen die Trajektorie (5.92) ein.

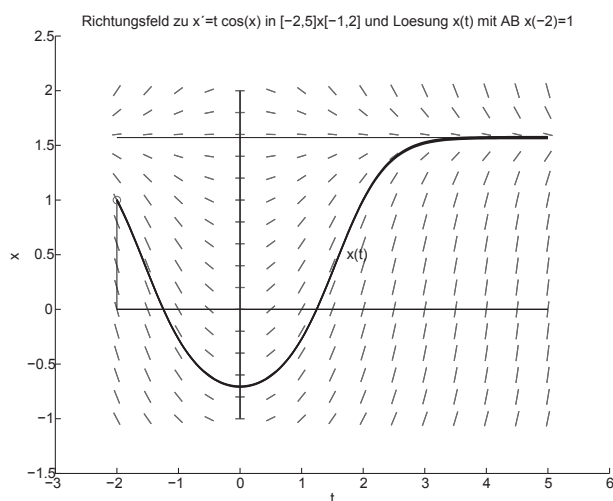


Abb. 5.6

Datei *r_f12b.ps*,
Richtungsfeld zur gDGl
 $x' = f(t, x) = t \cos(x)$
sowie Lösung $x(t)$
zur AB $x(-2) = 1$

Die Lösungstrajektorie $x(t) \equiv \frac{\pi}{2}$ stellt eine stabile Gleichgewichtslage dar.

Nun vergleichen wir die Werte verschiedener ESV $\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h)$, $i = 0, 1, \dots, N-1$, $h = (T - t_0)/N = 7/N$, an der Stelle $t = 5$ bei $N = 28$.

ESV	η_N	$ \eta_N - x(5) $	$\frac{ \eta_N - x(5) }{ x(5) }$
PZV	1.570 796 326	0.000 016 158	0.000 010 287
MPZV	1.570 590 480	0.000 189 687	0.000 120 759
HV	1.570 511 459	0.000 268 708	0.000 171 067
(3,2)-RKV (5.18)	1.570 768 044	0.000 012 123	0.000 007 717
(5,3)-RKV, b mit (5.89)	1.570 785 264	0.000 005 096	0.000 003 244
(5,3)-RKV, b mit (5.90)	1.570 781 407	0.000 001 239	0.000 000 788
KRKV	1.570 776 502	0.000 003 665	0.000 002 333
$x(5)$	1.570 780 167		

Tab. 5.5 Numer. Berechnung von η_N mit ESV, $h = 7/N = 0.25$, zu $x(5)$ und Fehler

ESV	$\max_{i=0(1)N} \eta_i - x(t_i) $	$\max_{i=0(1)N} \frac{ \eta_i - x(t_i) }{ x(t_i) }$
PZV	0.353 038 584	42.225 454 988
MPZV	0.010 177 358	0.176 545 563
HV	0.022 439 851	2.853 572 504
(3,2)-RKV (5.18)	0.010 693 805	1.362 920 749
(5,3)-RKV, b mit (5.89)	0.000 892 715	0.063 207 246
(5,3)-RKV, b mit (5.90)	0.000 316 554	0.041 707 089
KRKV	0.000 302 954	0.020 213 802

Tab. 5.6 Maximale Fehler im Intervall $[-2, 5]$

Wir bemerken, dass das (5,3)-RKV, **b** mit (5.90), einem Vergleich mit dem KRKV einigermassen standhält.

```
% rkv1_3a.m
% Explizites 5-stufiges Runge-Kutta-Verfahren der Ordnung 3 fuer gDG1
% + Zusatzzeile c[5]=1, A[5,:]=1/6(1,2,2,1,0) und b=1/6(1,2,2,0,1)^T
function [t,x,x4] = rkv1_3a(f,t0,t1,x0,N)
if N>0, h = (t1-t0)/N; else h = 0; end
t(1) = t0;
x(1) = x0;
x4(1) = x0;
for i = 1:N
    t(i+1) = t0+i*h;
    k1 = feval(f,t(i),x(i));
    k2 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k1);
    k3 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k2);
    k4 = feval(f,t(i)+h,x(i)+h*k3);
    x4(i+1) = x(i)+h/6*(k1+2*k2+2*k3+k4);
    k5 = feval(f,t(i)+h,x4(i+1));
    x(i+1) = x(i)+h/6*(k1+2*k2+2*k3+k5);
end;
```



```

% Datei rkv1_3b.m
% Explizites 5-stufiges Runge-Kutta-Verfahren der Ordnung 3 fuer gDGL
% + Zusatzzeile c[5]=1, A[5,:]=1/6(1,2,2,1,0) und b=1/6(1,2,2,1/2,1/2)^T

function [t,x,x4] = rkv1_3b(f,t0,t1,x0,N)
if N>0, h = (t1-t0)/N; else h = 0; end
t(1) = t0;
x(1) = x0; x4(1) = x0;
for i = 1:N
    t(i+1) = t0+i*h;
    k1 = feval(f,t(i),x(i));
    k2 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k1);
    k3 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k2);
    k4 = feval(f,t(i)+h,x(i)+h*k3);
    x4(i+1) = x(i)+h/6*(k1+2*k2+2*k3+k4);
    k5 = feval(f,t(i)+h,x4(i+1));
    x(i+1) = x(i)+h/6*(k1+2*k2+2*k3+0.5*k4+0.5*k5);
end;

```

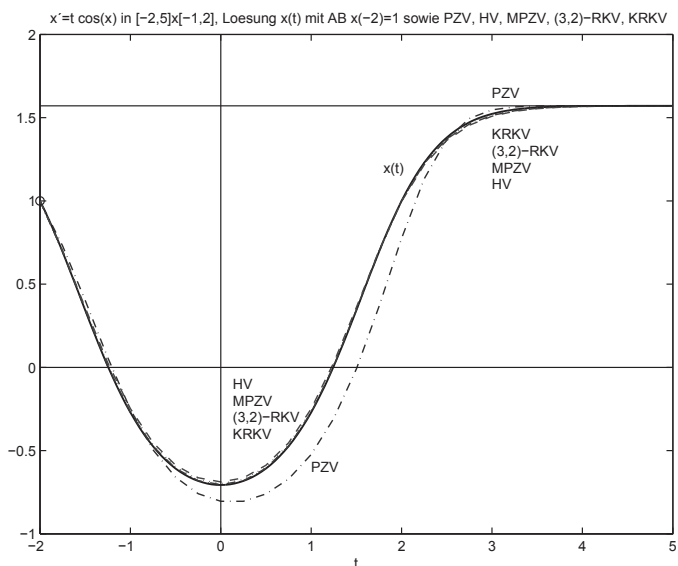


Abb. 5.7

Datei *r_f12c.ps*,
 $x' = f(t, x) = t \cos(x)$
 und Lösung $x(t)$
 mit der AB $x(-2) = 1$,
 dazu Näherungen
 mit PZV, HV, MPZV,
 (3,2)-RKV, KRKV

Die Schrittweite h soll nun kontrolliert, aber noch nicht gesteuert werden.

Wir rechnen dazu an den äquidistanten Knoten $t_i = t_0 + ih = -2 + \frac{i}{4}$, $i = 0, 1, \dots$, $N = 28$, und beobachten, wie sich der Fehler verhält. Damit ist im Weiteren auch eine Empfehlung für Schrittweitenveränderungen machbar.

Für die Schrittweitenkontrolle benutzen wir, wie das in den Dateien *rkv1_3a.m* und *rkv1_3b.m* erkennbar ist, die RKV der Ordnung $p = 3$ mit (5.89), (5.90) und das KRKV als Kontrollverfahren. Damit ist der lokale Diskretisierungsfehler $\tau(t_i, \eta_i, h)$ von der h -Ordnung 3 und $|\eta_{i+1} - \hat{\eta}_{i+1}| \approx \mathcal{O}(h^4)$.

In den nachfolgenden Tabellen sind in der zweiten Spalte die Ergebnisse von (5,3)-RKV und daneben die "lokale" Berechnung (der Schritt) mit dem KRKV, woraus sich die Kontrollgröße $|\eta_{i+1} - \hat{\eta}_{i+1}|$ in der Spalte 4 ergibt.

Andererseits rechnen wir das KRKV (Spalte 7) und jeweils "lokal" dazu das ungenauere (5,3)-RKV (Spalte 6) mit der Kontrollgröße in der Spalte 5 zwecks Vergleich. Diese Vorgehensweise ist in den Funktionen *rkv1_3am.m* und *rkv1_3bm.m* implementiert.

```
% rkv1_3am.m
% Explizites 5-stufiges Runge-Kutta-Verfahren der Ordnung 3 fuer gDG1
% + Zusatzzeile c[5]=1, A[5,:]=1/6(1,2,2,1,0) und b=1/6(1,2,2,0,1)^T

function [t,x,x3] = rkv1_3am(f,t0,t1,x0,N)
    if N>0, h = (t1-t0)/N; else h = 0; end
    t(1) = t0;
    x(1) = x0;
    x3(1) = x0;
    for i = 1:N
        t(i+1) = t0+i*h;
        k1 = feval(f,t(i),x(i));
        k2 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k1);
        k3 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k2);
        k4 = feval(f,t(i)+h,x(i)+h*k3);
        x(i+1) = x(i)+h/6*(k1+2*k2+2*k3+k4);
        k5 = feval(f,t(i)+h,x(i+1));
        x3(i+1) = x(i)+h/6*(k1+2*k2+2*k3+k5);
    end;

% rkv1_3bm.m
% Explizites 5-stufiges Runge-Kutta-Verfahren der Ordnung 3 fuer gDG1
% + Zusatzzeile c[5]=1, A[5,:]=1/6(1,2,2,1,0) und b=1/6(1,2,2,1/2,1/2)^T

function [t,x,x3] = rkv1_3bm(f,t0,t1,x0,N)
    if N>0, h = (t1-t0)/N; else h = 0; end
    t(1) = t0;
    x(1) = x0;
    x3(1) = x0;
    for i = 1:N
        t(i+1) = t0+i*h;
        k1 = feval(f,t(i),x(i));
        k2 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k1);
        k3 = feval(f,t(i)+0.5*h,x(i)+0.5*h*k2);
        k4 = feval(f,t(i)+h,x(i)+h*k3);
        x(i+1) = x(i)+h/6*(k1+2*k2+2*k3+k4);
        k5 = feval(f,t(i)+h,x(i+1));
        x3(i+1) = x(i)+h/6*(k1+2*k2+2*k3+0.5*k4+0.5*k5);
    end;
```

In der folgenden Tabelle verwenden wir das (5,3)-RKV, **b** mit (5.89), und das KRKV.

t(i)	eta3(i)	eta34(i)	. - .	. - .	eta43(i)	eta4(i)
-2.00	1.000000	1.000000	0.000000000	0.000000000	1.000000	1.000000
-1.75	0.694204	0.694014	0.000190662	0.000190662	0.694204	0.694014
-1.50	0.344616	0.344549	0.000067027	0.000066983	0.344382	0.344315
-1.25	0.007911	0.007911	0.000000652	0.000000623	0.007592	0.007592
-1.00	-0.269993	-0.269982	0.000010467	0.000010479	-0.270301	-0.270291
-0.75	-0.473345	-0.473337	0.000007298	0.000007303	-0.473620	-0.473612
-0.50	-0.607166	-0.607164	0.000002671	0.000002672	-0.607413	-0.607411
-0.25	-0.682064	-0.682064	0.000000398	0.000000399	-0.682295	-0.682295
0.00	-0.706083	-0.706083	0.000000000	0.000000000	-0.706309	-0.706309
0.25	-0.682064	-0.682064	0.000000003	0.000000003	-0.682295	-0.682295

0.50	-0.607167	-0.607167	0.000000231	0.000000231	-0.607411	-0.607410
0.75	-0.473350	-0.473348	0.000001900	0.000001900	-0.473614	-0.473612
1.00	-0.270010	-0.270005	0.000004768	0.000004772	-0.270294	-0.270289
1.25	0.007880	0.007879	0.000000415	0.000000400	0.007590	0.007589
1.50	0.344615	0.344580	0.000034956	0.000034931	0.344342	0.344307
1.75	0.694387	0.694259	0.000128518	0.000128472	0.694136	0.694007
2.00	1.000567	1.000243	0.000324180	0.000324128	1.000300	0.999976
2.25	1.229943	1.229398	0.000545854	0.000546094	1.229577	1.229031
2.50	1.381751	1.381112	0.000638625	0.000639994	1.381238	1.380598
2.75	1.472928	1.472341	0.000586867	0.000590235	1.472328	1.471738
3.00	1.523395	1.522933	0.000461431	0.000466955	1.522817	1.522351
3.25	1.549314	1.548991	0.000323380	0.000330478	1.548841	1.548510
3.50	1.561696	1.561490	0.000205390	0.000213070	1.561355	1.561142
3.75	1.567198	1.567079	0.000118982	0.000126220	1.566980	1.566853
4.00	1.569472	1.569409	0.000063016	0.000069060	1.569345	1.569276
4.25	1.570344	1.570313	0.000030528	0.000035048	1.570277	1.570242
4.50	1.570653	1.570640	0.000013516	0.000016564	1.570621	1.570604
4.75	1.570755	1.570749	0.000005458	0.000007321	1.570741	1.570733
5.00	1.570785	1.570783	0.000002003	0.000003039	1.570780	1.570777

Analog verfahren wir mit dem etwas besseren (5,3)-RKV, \mathbf{b} mit (5.90), und dem KRKV.

t(i)	eta3(i)	eta34(i)	. - .	. - .	eta43(i)	eta4(i)
-2.00	1.000000	1.000000	0.000000000	0.000000000	1.000000	1.000000
-1.75	0.694109	0.694014	0.000095331	0.000095331	0.694109	0.694014
-1.50	0.344466	0.344432	0.000033502	0.000033492	0.344349	0.344315
-1.25	0.007751	0.007751	0.000000319	0.000000311	0.007592	0.007592
-1.00	-0.270142	-0.270136	0.000005236	0.000005240	-0.270296	-0.270291
-0.75	-0.473479	-0.473475	0.000003650	0.000003652	-0.473616	-0.473612
-0.50	-0.607288	-0.607287	0.000001336	0.000001336	-0.607412	-0.607411
-0.25	-0.682180	-0.682179	0.000000199	0.000000199	-0.682295	-0.682295
0.00	-0.706196	-0.706196	0.000000000	0.000000000	-0.706309	-0.706309
0.25	-0.682180	-0.682180	0.000000001	0.000000001	-0.682295	-0.682295
0.50	-0.607289	-0.607289	0.000000116	0.000000116	-0.607411	-0.607410
0.75	-0.473481	-0.473480	0.000000950	0.000000950	-0.473613	-0.473612
1.00	-0.270150	-0.270148	0.000002385	0.000002386	-0.270292	-0.270289
1.25	0.007734	0.007734	0.000000204	0.000000200	0.007590	0.007589
1.50	0.344461	0.344443	0.000017472	0.000017466	0.344324	0.344307
1.75	0.694197	0.694133	0.000064247	0.000064236	0.694071	0.694007
2.00	1.000271	1.000109	0.000162077	0.000162064	1.000138	0.999976
2.25	1.229487	1.229214	0.000272987	0.000273047	1.229304	1.229031
2.50	1.381175	1.380855	0.000319655	0.000319997	1.380918	1.380598
2.75	1.472334	1.472040	0.000294275	0.000295118	1.472033	1.471738
3.00	1.522875	1.522642	0.000232094	0.000233478	1.522584	1.522351
3.25	1.548915	1.548751	0.000163458	0.000165239	1.548676	1.548510
3.50	1.561422	1.561317	0.000104602	0.000106535	1.561249	1.561142
3.75	1.567029	1.566968	0.000061281	0.000063110	1.566916	1.566853
4.00	1.569377	1.569344	0.000032992	0.000034530	1.569310	1.569276
4.25	1.570295	1.570278	0.000016363	0.000017524	1.570259	1.570242
4.50	1.570630	1.570623	0.000007489	0.000008282	1.570613	1.570604
4.75	1.570745	1.570742	0.000003167	0.000003660	1.570737	1.570733
5.00	1.570781	1.570780	0.000001239	0.000001520	1.570778	1.570777

Beide (5,3)-RKV mit (5.89) bzw. (5.90) führen auf ähnliche Ergebnisse.

Dazu gewinnt man noch folgende Erkenntnisse. Am Anfang der Rechnung muss sich eine Schrittweite meist erst "einpendeln". Im Verlauf werden dann in Bereichen mit flachem Lösungsverlauf die Kontrollgrößen kleiner, was zu einem Vorschlag für eine Schrittweitenvergrößerung führt. Dort, wo starke Anstiege bzw. Krümmungen auftreten, wachsen die Kontrollgrößen, sodass man h entsprechend verkleinern sollte.

Die Abbildungen und Tabellenwerte sind dem MATLAB-Skriptfile *r_rkv341.m* zu entnehmen.

```
% r_rkv341.m
% Runge-Kutta-Verfahren fuer gDGl
% --> f12.m x'=t cos(x)

clear all
clc
format long
diary f12m.pro
diary on

% Richtungsfeld zur gDGl x'=f(t,x) in Gitter Omega1
figure(1)
dfield('f12',-2:0.5:5,-1:0.2:2);
title('Richtungsfeld zu x=t cos(x) in [-2,5]x[-1,2]');
xlabel('t');
ylabel('x');
print r_f12a.ps -dpsc
pause

% exakte Loesung
tt = -2:0.05:5;
nt = max(size(tt));
for i=1:nt
    xx(i)=f12ex(tt(i));
end;
xxe = xx(nt)
pause

figure(1)
dfield('f12',-2:0.5:5,-1:0.2:2);
title('Richtungsfeld zu x=t cos(x) in [-2,5]x[-1,2] und Loesung x(t)...
      mit AB x(-2)=1');
hold on
plot([-2 -2],[0 1],'k-',[ -2 5],[1.571 1.571],'k-');
plot([-2],[1],'o');
plot(tt,xx,'r-',tt,0.995*xx,'r-',tt,1.005*xx,'r-');
xlabel('t');
text(1.9,0.5,'x(t)');
print r_f12b.ps -dpsc
hold off
pause

t0 = -2
t1 = 5
x0 = 1
N = 28
h = (t1-t0)/N
disp('Test RKV');
[te,xe] = euler1('f12',t0,t1,x0,N);
eu = xe(N+1)
[tc,xc] = collatz1('f12',t0,t1,x0,N);
co = xc(N+1)
[th,xh] = heun1('f12',t0,t1,x0,N);
he = xh(N+1)
```

```

A = [0 0 0; 1/2 0 0; 1/3 2/3 0];
b = [1/3 1/3 1/3]';
c = [0 1/2 1]';
disp('(3,2)-RKV');
disp('rkv1');
[t3,x3] = rkv1(A,b,c,'f12',t0,t1,x0,N);
k3 = x3(N+1)

A = [0 0 0 0 0; 1/2 0 0 0 0; 0 1/2 0 0 0; 0 0 1 0 0; 1/6 1/3 1/3 1/6 0];
b = [1/6 1/3 1/3 0 1/6]';
c = [0 1/2 1/2 1 1]';
disp('(5,3)-RKV');
disp('rkv1_3a');
[t3a,x3a,x4a] = rkv1_3a('f12',t0,t1,x0,N);
k3a = x3a(N+1)
[t3am,xam,x3am] = rkv1_3am('f12',t0,t1,x0,N);
k3am = x3am(N+1)

for i = 1:N+1
    fprintf(1,'%4.2f %9.6f %9.6f %12.9f %12.9f %9.6f %9.6f\n',...
        t3a(i),x3a(i),x4a(i),abs(x3a(i)-x4a(i)),...
        abs(x3am(i)-xam(i)),x3am(i),xam(i) );
end;

A = [0 0 0 0 0; 1/2 0 0 0 0; 0 1/2 0 0 0; 0 0 1 0 0; 1/6 1/3 1/3 1/6 0];
b = [1/6 1/3 1/3 1/12 1/12]';
c = [0 1/2 1/2 1 1]';
disp('(5,3)-RKV');
disp('rkv1_3b');
[t3b,x3b,x4b] = rkv1_3b('f12',t0,t1,x0,N);
k3b = x3b(N+1)
[t3bm,xbm,x3bm] = rkv1_3bm('f12',t0,t1,x0,N);
k3bm = x3bm(N+1)

for i = 1:N+1
    fprintf(1,'%4.2f %9.6f %9.6f %12.9f %12.9f %9.6f %9.6f\n',...
        t3b(i),x3b(i),x4b(i),abs(x3b(i)-x4b(i)),...
        abs(x3bm(i)-xbm(i)),x3bm(i),xbm(i) );
end;

A = [0 0 0 0; 1/2 0 0 0; 0 1/2 0 0; 0 0 1 0];
b = [1/6 1/3 1/3 1/6]';
c = [0 1/2 1/2 1]';
disp('krkv1');
[t,x] = rkv1(A,b,c,'f12',t0,t1,x0,N);
k4 = x(N+1)

% identisch mit ...
disp('krkv1');
[t,x] = krkv1('f12',t0,t1,x0,N);
k1 = x(N+1)

% exakte Loesung auf Gitter zu den ESV
ttn = -2:0.25:5; # ttn = linspace(t0,t1,N+1);
ntn = max(size(ttn));
for i=1:ntn
    xxn(i)=f12ex(ttn(i));
end;
pause

```

```

disp('Endwerte: eu,co,he,k3,k3a,k3b,k4, exakt');
disp(eu),disp(co),disp(he),disp(k3),disp(k3a),disp(k3b),...
disp(k4),disp(xxe);

disp('absolute Fehler bei t=5');
disp(abs(eu-xxe)),disp(abs(co-xxe)),disp(abs(he-xxe)),...
disp(abs(k3-xxe)),disp(abs(k3a-xxe)),disp(abs(k3b-xxe)),disp(abs(k4-xxe));

disp('relative Fehler bei t=5');
disp(abs((eu-xxe)/xxe)),disp(abs((co-xxe)/xxe)),disp(abs((he-xxe)/xxe)),...
disp(abs((k3-xxe)/xxe)),disp(abs((k3a-xxe)/xxe)),disp(abs((k3b-xxe)/xxe)),...
disp(abs((k4-xxe)/xxe));

disp('max(absolute Fehler) auf [-2,5]');
disp(max(abs(xe-xxn))),disp(max(abs(xc-xxn))),disp(max(abs(xh-xxn))),...
disp(max(abs(x3-xxn))),disp(max(abs(x3a-xxn))),disp(max(abs(x3b-xxn))),...
disp(max(abs(x-xxn)));

disp('max(relative Fehler) auf [-2,5]');
disp(max(abs(xe-xxn)./xxn)),disp(max(abs(xc-xxn)./xxn)),...
disp(max(abs(xh-xxn)./xxn)),disp(max(abs(x3-xxn)./xxn)),...
disp(max(abs(x3a-xxn)./xxn)),disp(max(abs(x3b-xxn)./xxn)),...
disp(max(abs(x-xxn)./xxn));
diary off
pause

figure(2)
plot(tt,xx,'k-',t,x,'b:',te,xe,'r-.',tc,xc,'g--',th,xh,'m--',t3,x3,'c:');
title('x=t cos(x) in [-2,5]x[-1,2], Loesung x(t) mit AB x(-2)=1 ...
      sowie PZV, HV, MPZV, (3,2)-RKV, KRKV');
xlabel('t');
text(1.8,1.2,'x(t)');
text(1,-0.6,'PZV');
text(0.2,-0.1,'HV');
text(0.2,-0.2,'MPZV');
text(0.2,-0.3,'(3,2)-RKV');
text(0.2,-0.4,'KRKV');
text(3,1.65,'PZV');
text(3,1.4,'KRKV');
text(3,1.3,'(3,2)-RKV');
text(3,1.2,'MPZV');
text(3,1.1,'HV');

axis([-2 5 -1 2]);
hold on
plot([-2 5],[0 0],'k-',[1.571 1.571],'k-',[0 0],[-1 2],'k-');
plot([-2],[1],'o');
print r_f12c.ps -dpsc
hold off
pause

plot(tt,xx,'k-',t,x,'b:',t3,x3,'r--',t3a,x3a,'g--',t3b,x3b,'m--');
axis([-2 5 -1 2]);
pause

plot(tt,xx,'k-',t,x,'b:',t3a,x3a,'r--');
axis([-2 5 -1 2]);
pause
diary off

```

(b) Gegeben sei das AWP

$$x' = f(t, x) = \frac{1}{x + 10^{-p}}, \quad p = 1, \quad x(0) = 0, \quad t \in [t_0, T] = [0, 5]. \quad (5.93)$$

Die beiden allgemeinen Lösungen sind

$$x_1(t) = -10^{-p} + \sqrt{10^{-2p} + 2(t + C)}, \quad x_2(t) = -10^{-p} - \sqrt{10^{-2p} + 2(t + C)}. \quad (5.94)$$

Die exakte Lösung des AWP ist

$$x(t) = -10^{-p} + \sqrt{10^{-2p} + 2t}, \quad p = 1. \quad (5.95)$$

Zunächst interessieren wir uns für das Richtungsfeld zur gDGI und das AWP mit ausgewählten AB. Wir nutzen das Maple-Tool `DEplot` mit dem KRKV.

Dabei verwenden wir in `DEplot` die Option `stepsize=0.2`, also eine grobe Schrittweite im KRKV, um das Problem der Ungenauigkeiten zu Beginn der Rechnung wegen des steilen Anstiegs bei der AB $x(0) = 0$ der Lösung zu verdeutlichen.

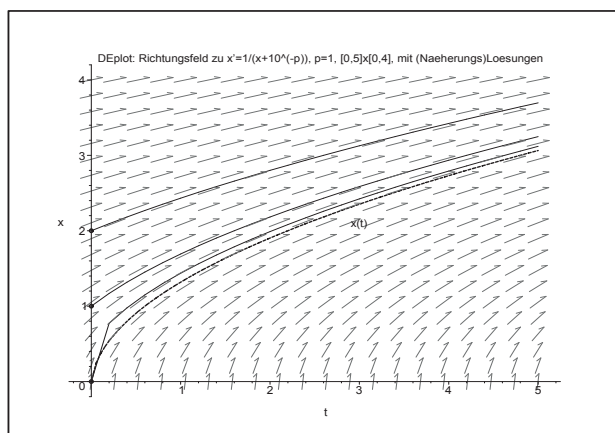


Abb. 5.8

Datei `richt11.ps`,
Richtungsfeld zur gDGI
 $x' = 1/(x + 10^{-p})$, $p = 1$,
sowie Näherungslösungen
zu den AB $x(0) = 0$,
 $x(0) = 1$, $x(0) = 2$,
sowie $x(t)$ mit AB $x(0) = 0$
(untere Kurve)

Wir zeichnen das Richtungsfeld auch mit MATLAB und tragen die Trajektorie (5.95) ein.

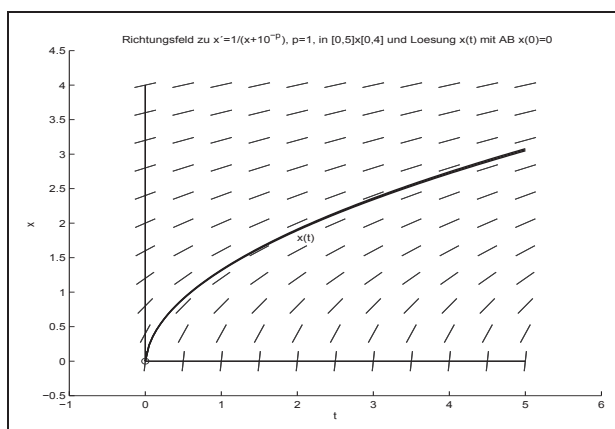


Abb. 5.9

Datei `r_f11b.ps`,
Richtungsfeld zur gDGI
 $x' = 1/(x + 10^{-p})$, $p = 1$,
sowie Lösung $x(t)$
zur AB $x(0) = 0$

Nun vergleichen wir die Werte verschiedener ESV $\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h)$, $i = 0, 1, \dots, N-1$, $h = (T - t_0)/N = 5/N$, an der Stelle $t = 5$ bei $N = 25$.

ESV	η_N	$ \eta_N - x(5) $	$\frac{ \eta_N - x(5) }{ x(5) }$
PZV	3.658 726 899	0.594 868 495	0.194 156 653
MPZV	2.985 706 265	0.078 152 138	0.025 507 751
HV	3.204 139 610	0.140 281 206	0.045 785 799
(3,2)-RKV (5.18)	3.127 357 066	0.063 498 662	0.020 725 064
(5,3)-RKV, b mit (5.89)	3.121 312 066	0.057 453 662	0.018 752 062
(5,3)-RKV, b mit (5.90)	3.120 058 858	0.056 200 454	0.018 343 032
KRKV	3.118 811 604	0.054 953 200	0.017 935 946
$x(5)$	3.063 858 403		

Tab. 5.7 Numerische Berechnung von η_N mit ESV, $h = \frac{5}{N} = 0.2$, zu $x(5)$ und Fehler

ESV	$\max_{i=0(1)N} \eta_i - x(t_i) $	$\max_{i=1(1)N} \frac{ \eta_i - x(t_i) }{ x(t_i) }$
PZV	1.459 687 576	2.701 562 118
MPZV	0.358 494 241	0.663 494 352
HV	0.507 306 623	0.938 913 490
(3,2)-RKV (5.18)	0.262 045 627	0.484 989 084
(5,3)-RKV, b mit (5.89)	0.241 051 231	0.446 133 052
(5,3)-RKV, b mit (5.90)	0.236 465 096	0.437 645 122
KRKV	0.231 878 962	0.429 157 191

Tab. 5.8 Maximale Fehler im Intervall $[0, 5]$

Wir bemerken, dass die (5,3)-RKV durchaus mit dem KRKV vergleichbar sind.

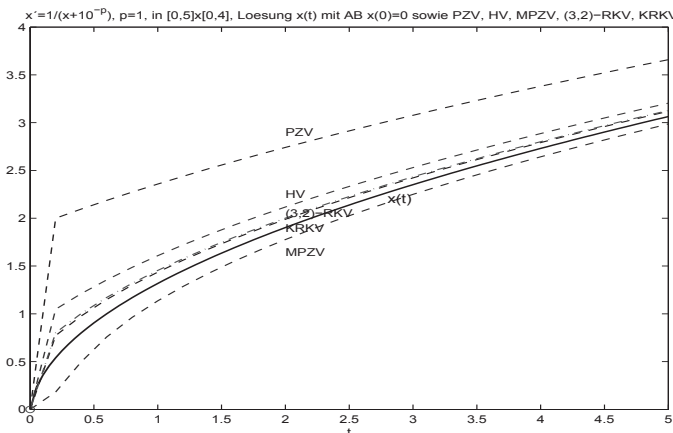


Abb. 5.10 Datei *r_f11c.ps*, $x' = 1/(x + 10^{-p})$, $p = 1$, und Lösung $x(t)$ zur AB $x(0) = 0$, dazu Näherungen mit PZV, HV, MPZV, (3,2)-RKV, KRKV bei $h = 0.2$

Die Schrittweite h soll wieder kontrolliert, aber noch nicht gesteuert werden.

Wir rechnen hier an äquidistanten Knoten $t_i = t_0 + ih = \frac{i}{5}$, $i = 0, 1, \dots, N = 25$, und beobachten, wie sich die Fehler verhalten.

Für die Schrittweitenkontrolle benutzen wir wiederum das (5,3)-RKV der Ordnung $p = 3$ mit (5.89), (5.90) und das KRKV als Kontrollverfahren.

Die nachfolgenden Tabellen sind wie im Teil (a) des Beispiels zu verstehen.

(5,3)-RKV, \mathbf{b} mit (5.89), und KRKV

t(i)	eta3(i)	eta34(i)	. - .	. - .	eta43(i)	eta4(i)
0.0	0.000000	0.000000	0.000000000000	0.000000000000	0.000000	0.000000
0.2	0.781364	0.772191	0.009172268485	0.009172268485	0.781364	0.772191
0.4	0.984821	0.984814	0.000006774776	0.000007295126	0.977384	0.977376
0.6	1.155725	1.155724	0.000001469632	0.000001548596	1.149299	1.149298
0.8	1.306004	1.306004	0.000000472197	0.000000491732	1.300267	1.300267
1.0	1.441704	1.441703	0.000000191092	0.000000197522	1.436473	1.436472
1.2	1.566388	1.566388	0.000000090056	0.000000092610	1.561550	1.561550
1.4	1.682372	1.682372	0.000000047292	0.000000048451	1.677849	1.677849
1.6	1.791256	1.791256	0.000000026927	0.000000027509	1.786994	1.786994
1.8	1.894204	1.894204	0.000000016326	0.000000016641	1.890163	1.890163
2.0	1.992093	1.992093	0.000000010408	0.000000010589	1.988240	1.988240
2.2	2.085601	2.085601	0.000000006913	0.000000007023	2.081914	2.081914
2.4	2.175269	2.175269	0.000000004751	0.000000004820	2.171728	2.171728
2.6	2.261536	2.261536	0.000000003361	0.000000003406	2.258124	2.258124
2.8	2.344760	2.344760	0.000000002437	0.000000002468	2.341464	2.341464
3.0	2.425243	2.425243	0.000000001806	0.000000001827	2.422052	2.422052
3.2	2.503239	2.503239	0.000000001363	0.000000001378	2.500144	2.500144
3.4	2.578965	2.578965	0.000000001046	0.000000001057	2.575957	2.575957
3.6	2.652608	2.652608	0.000000000815	0.000000000823	2.649681	2.649681
3.8	2.724332	2.724332	0.000000000643	0.000000000649	2.721480	2.721480
4.0	2.794279	2.794279	0.000000000513	0.000000000518	2.791496	2.791496
4.2	2.862575	2.862575	0.000000000414	0.000000000418	2.859856	2.859856
4.4	2.929332	2.929332	0.000000000338	0.000000000341	2.926673	2.926673
4.6	2.994649	2.994649	0.000000000278	0.000000000280	2.992046	2.992046
4.8	3.058615	3.058615	0.000000000230	0.000000000232	3.056065	3.056065
5.0	3.121312	3.121312	0.000000000192	0.000000000194	3.118812	3.118812

(5,3)-RKV, \mathbf{b} mit (5.90), und KRKV an ausgewählten Stellen

t(i)	eta3(i)	eta34(i)	. - .	. - .	eta43(i)	eta4(i)
0.0	0.000000	0.000000	0.000000000000	0.000000000000	0.000000	0.000000
0.2	0.776778	0.772191	0.004586134242	0.004586134242	0.776778	0.772191
0.4	0.981096	0.981092	0.000003514864	0.000003647563	0.977380	0.977376
0.6	1.152507	1.152507	0.000000754292	0.000000774298	1.149299	1.149298
0.8	1.303131	1.303131	0.000000240934	0.000000245866	1.300267	1.300267
1.0	1.439083	1.439083	0.000000097141	0.000000098761	1.436472	1.436472
1.2	1.563965	1.563965	0.000000045663	0.000000046305	1.561550	1.561550
1.4	1.680106	1.680106	0.000000023934	0.000000024225	1.677849	1.677849
1.6	1.789121	1.789121	0.000000013608	0.000000013754	1.786994	1.786994
1.8	1.892179	1.892179	0.000000008242	0.000000008321	1.890163	1.890163
2.0	1.990162	1.990162	0.000000005249	0.000000005295	1.988240	1.988240
3.0	2.423644	2.423644	0.000000000908	0.000000000913	2.422052	2.422052
4.0	2.792884	2.792884	0.000000000258	0.000000000259	2.791496	2.791496
5.0	3.120059	3.120059	0.000000000097	0.000000000097	3.118812	3.118812

Auch die Ergebnisse sind analog zum Teil (a). Nach einer notwendigen Anpassung der Schrittweite am Anfang des Intervalls durch den starken Anstieg der Lösung kann anschließend eine schnelle Schrittweitenvergrößerung vorgenommen werden.

5.4.2 Schrittweitenkontrolle mittels Einbettung

Die Idee der Einbettung bei ESV haben wir im Abschnitt 5.4.1 schon erwähnt. Die Methode geht auf E. Fehlberg zurück. Basierend auf zwei "benachbarten" RKV, kann sie mit sehr geringem Zusatzaufwand durchgeführt und zur SWS verwendet werden. Das würde bedeuten, dass man zum RKV der Ordnung p mit der Näherungslösung η_{i+1} seine SWS durch den Vergleich mit einem RKV der Ordnung $p + 1$ und seiner Lösung $\hat{\eta}_{n+1}$ (als sogenannten Repräsentant der exakten Lösung) erhält. Damit gilt

$$|\eta_{n+1} - \hat{\eta}_{n+1}| \approx h |\tau(t_n, \eta_n, h)| = \mathcal{O}(h^{p+2})$$

und $|\eta_{n+1} - \hat{\eta}_{n+1}|$ kann als Maß für die Wahl der Schrittweite h genommen werden. Es besteht natürlich auch die Möglichkeit, die genaueren Werte für den Fortgang der Rechnung zu verwenden.

Diese Vorgehensweise ist z.B. in den eingebetteten Runge-Kutta-Fehlberg-Verfahren realisiert.

Der Ausgangspunkt des RKFV sind zwei explizite RKV gegeben durch ihre Butcher-Tableaus

$c_1 = 0$							
c_2	a_{21}						
c_3	a_{31}	a_{32}					
...				
c_{r-1}	$a_{r-1,1}$	$a_{r-1,2}$...	$a_{r-1,r-2}$			
c_r	$a_{r,1}$	$a_{r,2}$...	$a_{r,r-2}$	$a_{r,r-1}$		
c_{r+1}	$a_{r+1,1}$	$a_{r+1,2}$...	$a_{r+1,r-2}$	$a_{r+1,r-1}$	$a_{r+1,r}$	
1	\hat{b}_1	\hat{b}_2	...	\hat{b}_{r-2}	\hat{b}_{r-1}	\hat{b}_r	\hat{b}_{r+1}

und

$c_1 = 0$						
c_2	a_{21}					
c_3	a_{31}	a_{32}				
...			
c_{r-1}	$a_{r-1,1}$	$a_{r-1,2}$...	$a_{r-1,r-2}$		
c_r	$a_{r,1}$	$a_{r,2}$...	$a_{r,r-2}$	$a_{r,r-1}$	
1	b_1	b_2	...	b_{r-2}	b_{r-1}	b_r

Das erste $(r + 1)$ -RKV ist von der Ordnung $p + 1 \leq r + 1$, das zweite von der Ordnung $p \leq r$. In beiden Schemata sind die Koeffizienten c_i und a_{ij} dieselben für $i = 1, 2, \dots, r$.

Gewöhnlich notiert man das gemeinsame Tableau des RKFV als

$$\begin{array}{c|cccccc}
 c_1 = 0 & & & & & & & \\
 c_2 & a_{21} & & & & & & \\
 c_3 & a_{31} & a_{32} & & & & & \\
 \dots & \dots & \dots & \ddots & & & & \\
 c_{r-1} & a_{r-1,1} & a_{r-1,2} & \dots & a_{r-1,r-2} & & & \\
 c_r & a_{r,1} & a_{r,2} & \dots & a_{r,r-2} & a_{r,r-1} & & \\
 c_{r+1} & a_{r+1,1} & a_{r+1,2} & \dots & a_{r+1,r-2} & a_{r+1,r-1} & a_{r+1,r} & \\
 \hline
 1 & b_1 & b_2 & \dots & b_{r-2} & b_{r-1} & b_r & \\
 \hline
 1 & \hat{b}_1 & \hat{b}_2 & \dots & \hat{b}_{r-2} & \hat{b}_{r-1} & \hat{b}_r & \hat{b}_{r+1}
 \end{array} \tag{5.96}$$

Die entsprechenden Näherungslösungen sind

$$\begin{aligned}
 \hat{\eta}_{n+1} &= \eta_n + h \Psi(t_n, \eta_n, h) = \eta_n + h \sum_{i=1}^{r+1} \hat{b}_i k_i, \\
 \eta_{n+1} &= \eta_n + h \Psi(t_n, \eta_n, h) = \eta_n + h \sum_{i=1}^r b_i k_i.
 \end{aligned} \tag{5.97}$$

Mit dem lokalen Diskretisierungsfehler τ_{n+1} aus der Beziehung (5.35) hat man

$$h \tau_{n+1} = x_{n+1} - x_n - h \Psi(t_n, x_n, h) = \mathcal{O}(h^q), \quad q = p + 2, p + 1, \tag{5.98}$$

bzw.

$$\begin{aligned}
 x_{n+1} &= x_n + h \sum_{i=1}^{r+1} \hat{b}_i k_i + \hat{\psi}(t_n, x_n) h^{p+2} + \mathcal{O}(h^{p+3}), \\
 x_{n+1} &= x_n + h \sum_{i=1}^r b_i k_i + \psi(t_n, x_n) h^{p+1} + \mathcal{O}(h^{p+2}),
 \end{aligned} \tag{5.99}$$

wobei

$$\begin{aligned}
 k_1 &= f(t_n, x_n), \\
 k_i &= f\left(t_n + c_i h, x_n + h \sum_{j=1}^{i-1} a_{ij} k_j\right), \quad i = 2, 3, \dots, r + 1,
 \end{aligned}$$

sowie $\psi(t_n, x_n)$ und $\hat{\psi}(t_n, x_n)$ die Hauptteile der Fehler sind.

Subtrahieren wir jeweils die beiden Seiten in (5.99), erhält man mit

$$\psi(t_n, x_n) h^{p+1} = \sum_{i=1}^{r+1} (\hat{b}_i - b_i) k_i + \mathcal{O}(h^{p+2}), \quad b_{r+1} = 0. \tag{5.100}$$

Damit findet man die Formel, die den Fehler des RKFV beschreibt, also

$$R(h) = \sum_{i=1}^{r+1} (\hat{b}_i - b_i) k_i = \frac{1}{h} (\hat{\eta}_{n+1} - \eta_{n+1}), \quad b_{r+1} = 0. \tag{5.101}$$

$R(h)$ ist ein asymptotischer Fehlerschätzer der Ordnung $p + 1$ für den lokalen Diskretisierungsfehler τ_{n+1} des eingebetteten RKFV.

Beim RKFV sind also zunächst die Anstiege k_1, k_2, \dots, k_{r+1} zu berechnen, dann der Fehler $R(h)$ (5.101) auszuwerten. Wenn dieser hinreichend klein ist, wird der Schritt akzeptiert und $\hat{\eta}_{n+1}$ als Näherungswert genommen. Sonst ist der Schritt mit kleinerem h zu wiederholen. Für die SWS sind unterschiedliche Strategien möglich. Die duale SWS basiert auf Grob- und Feinrechnung (siehe Abschnitt 5.4.4), die variable SWS ist im Abschnitt 8.5.2 mit den eingebetteten Verfahren dargestellt.

Wegen (5.100) ist der Ausdruck (5.101) ein Schätzer des enthaltenen RKV der Ordnung p . Die Verwendung dieser Abschätzung für das genauere RKV dient zugleich zur Verbesserung des Ergebnisses η_{n+1} , das dann bei Berücksichtigung von $R(h)$ mit der Näherung $\hat{\eta}_{n+1}$ vergleichbar ist. Der Aufwand für die Schätzer ist marginal, da keine zusätzlichen Funktionsauswertungen für $f(t, x)$ notwendig sind.

Gebäuchliche RKFV der Ordnungen $p(p+1)$ sind die folgenden (siehe auch [39]). Will man die Stufenzahl s noch vermerken, dann notiert man $\text{RKFp}(p+1)_s$.

RKF1(2)₃

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	$\frac{1}{256}$	$\frac{255}{256}$	
	$\frac{1}{256}$	$\frac{255}{256}$	
	$\frac{1}{512}$	$\frac{255}{256}$	$\frac{1}{512}$

RKF2(3)₃

0			
1	1		
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	
	$\frac{1}{2}$	$\frac{1}{2}$	
	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{4}{6}$

RKF2(3)₄

0				
$\frac{1}{4}$	$\frac{1}{4}$			
$\frac{27}{40}$	$-\frac{189}{800}$	$\frac{729}{800}$		
1	$\frac{214}{800}$	$\frac{1}{33}$	$\frac{650}{891}$	
	$\frac{214}{891}$	$\frac{1}{33}$	$\frac{650}{891}$	
	$\frac{533}{2106}$	0	$\frac{800}{1053}$	$-\frac{1}{78}$

Verf. RKF3(4)₄ von Ceschino

0				
$\frac{1}{4}$	$\frac{1}{4}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	1	-2	2	
	1	-2	2	
	$\frac{1}{6}$	0	$\frac{4}{6}$	$\frac{1}{6}$

RKF4(5)₆

0						
$\frac{1}{4}$	$\frac{1}{4}$					
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$				
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$			
1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$		
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	
	$\frac{25}{216}$	0	$\frac{1408}{2565}$	$\frac{2197}{4104}$	$-\frac{1}{5}$	
	$\frac{16}{135}$	0	$\frac{6656}{12825}$	$\frac{28561}{56430}$	$-\frac{9}{50}$	$\frac{2}{55}$

Eingebettetes Verfahren 4(5)₆ von Sarafyan

0						
$\frac{1}{2}$	$\frac{1}{2}$					
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$				
1	0	-1	2			
$\frac{2}{3}$	$\frac{7}{27}$	$\frac{10}{27}$	0	$\frac{1}{27}$		
$\frac{1}{5}$	$\frac{28}{625}$	$-\frac{125}{625}$	$\frac{546}{625}$	$\frac{54}{625}$	$-\frac{378}{625}$	
	$\frac{1}{6}$	0	$\frac{4}{6}$	$\frac{1}{6}$	0	
	$\frac{14}{336}$	0	0	$\frac{35}{336}$	$\frac{162}{336}$	$\frac{125}{336}$

Eingebettetes RKV Fehlberg5(6)₈

0								
$\frac{1}{6}$	$\frac{1}{6}$							
$\frac{4}{15}$	$\frac{4}{75}$	$\frac{16}{75}$						
$\frac{2}{3}$	$\frac{5}{6}$	$-\frac{8}{3}$	$\frac{5}{2}$					
$\frac{4}{5}$	$-\frac{8}{5}$	$\frac{144}{25}$	-4	$\frac{16}{25}$				
1	$\frac{361}{320}$	$-\frac{18}{5}$	$\frac{407}{128}$	$-\frac{11}{80}$	$\frac{55}{128}$			
0	$-\frac{11}{640}$	0	$\frac{11}{256}$	$-\frac{11}{160}$	$\frac{11}{256}$	0		
1	$\frac{93}{640}$	$-\frac{18}{5}$	$\frac{803}{256}$	$-\frac{11}{160}$	$\frac{99}{256}$	0	1	
	$\frac{31}{384}$	0	$\frac{1125}{2816}$	$\frac{9}{32}$	$\frac{125}{768}$	$\frac{5}{66}$		
	$\frac{7}{1408}$	0	$\frac{1125}{2816}$	$\frac{9}{32}$	$\frac{125}{768}$	0	$\frac{5}{66}$	$\frac{5}{66}$

Eingebettetes RKV Fehlberg5(6)₈ modifiziert $c_5 = 0.80246913580$, $a_{8,7} = 0.010304142834$, $b_6 = 0.0728409090909$, $\hat{b}_7 = 0.00083226803025$, $\hat{b}_8 = 0.075385551948$

0								
$\frac{1}{6}$	$\frac{1}{6}$							
$\frac{4}{15}$	$\frac{4}{75}$	$\frac{16}{75}$						
$\frac{2}{3}$	$\frac{5}{6}$	$-\frac{8}{3}$	$\frac{5}{2}$					
c_5	-1.6668571864	5.9926501719		-4.1802147369	0.65689088723			
1	1.2002177068	-3.8490566038		3.3706489291	-0.14308176101	0.42127172888		
$\frac{1}{16}$	-2.0977589744	1.6533333333		2.9213133641	-6.4930909091	4.0828698527	0	
1	1.1968785589	-3.7954239569		3.2777419424	-0.063718159613	0.37421747248	0	$a_{8,7}$
	0.08076923076877	0	0.39929828236678	0.284090909090909	0.160557486862298	b_6		
	0.080288461538	0	0.39875279291	0.284722222222222	0.16001870335	0	\hat{b}_7	\hat{b}_8

8-stufige RKV der Ordnungen 5(6) stammen von J.H. Verner [113], wo man mit den darin berechneten Steigungswerten k_1, k_2, \dots, k_8 auch Näherungen der Ordnungen 1,2,3 und 4 ermitteln kann und somit eine komplette Menge eingebetteter RKV (sogenannte CSIRK-Verfahren, complete set of imbedded Runge-Kutta-methods) erhält.

Eingebettetes RKV Verner5(6)₈

0								
$\frac{1}{6}$	$\frac{1}{6}$							
$\frac{4}{15}$	$\frac{4}{75}$	$\frac{16}{75}$						
$\frac{2}{3}$	$\frac{5}{6}$	$-\frac{8}{3}$	$\frac{5}{2}$					
$\frac{3}{4}$	$-\frac{219}{512}$	$\frac{27}{16}$	$-\frac{435}{512}$	$\frac{87}{256}$				
1	$\frac{1}{12}$	0	$\frac{25}{58}$	$-\frac{1}{4}$	$\frac{64}{87}$			
$\frac{1}{15}$	$-\frac{517}{1500}$	$\frac{124}{75}$	$-\frac{212}{145}$	$\frac{291}{500}$	$-\frac{1312}{3625}$	0		
1	$\frac{787}{1812}$	$-\frac{108}{151}$	$\frac{9185}{8758}$	$-\frac{517}{1812}$	$\frac{390368}{538617}$	0	$-\frac{3850}{18573}$	
	$\frac{23}{288}$	0	$\frac{375}{928}$	$\frac{3}{16}$	$\frac{64}{261}$	$\frac{1}{12}$		
	$\frac{13}{144}$	0	$\frac{2125}{5104}$	$\frac{1}{6}$	$\frac{2816}{10701}$	0	$-\frac{125}{6888}$	$\frac{151}{1848}$

Eingebettetes RKV Verner5(6)₈ modifiziert

0								
$\frac{1}{18}$	$\frac{1}{18}$							
$\frac{1}{6}$	$-\frac{1}{12}$	$\frac{1}{4}$						
$\frac{2}{9}$	$-\frac{2}{81}$	$\frac{4}{27}$	$\frac{8}{81}$					
$\frac{2}{3}$	$\frac{40}{33}$	$-\frac{4}{11}$	$-\frac{56}{11}$	$\frac{54}{11}$				
1	$-\frac{369}{73}$	$\frac{72}{73}$	$\frac{5380}{219}$	$-\frac{12285}{584}$	$\frac{2695}{1752}$			
$\frac{8}{9}$	$-\frac{8716}{891}$	$\frac{656}{297}$	$\frac{39520}{891}$	$-\frac{416}{11}$	$\frac{52}{27}$	0		
1	$\frac{3015}{256}$	$-\frac{9}{4}$	$-\frac{4219}{78}$	$\frac{5985}{128}$	$-\frac{539}{384}$	0	$\frac{693}{3328}$	
	$\frac{3}{80}$	0	$\frac{4}{25}$	$\frac{243}{1120}$	$\frac{77}{160}$	$\frac{73}{700}$		
	$\frac{57}{640}$	0	$-\frac{16}{65}$	$\frac{1377}{2240}$	$\frac{121}{320}$	0	$\frac{891}{8320}$	$\frac{2}{35}$

Eingebettetes RKV Prince-Dormand5(6)₈

0								
$\frac{1}{10}$	$\frac{1}{10}$							
$\frac{2}{9}$	$-\frac{2}{81}$	$\frac{20}{81}$						
$\frac{3}{7}$	$\frac{615}{1372}$	$-\frac{1080}{1372}$	$\frac{1053}{1372}$					
$\frac{3}{5}$	$\frac{3243}{5500}$	$-\frac{54}{55}$	$\frac{50949}{71500}$	$\frac{4998}{17875}$				
$\frac{4}{5}$	$-\frac{26492}{37125}$	$\frac{72}{55}$	$\frac{2808}{23375}$	$-\frac{24206}{37125}$	$\frac{338}{459}$			
1	$\frac{5561}{2376}$	$-\frac{35}{11}$	$-\frac{24117}{31603}$	$\frac{899983}{200772}$	$-\frac{5225}{1836}$	$\frac{3925}{4056}$		
1	$\frac{465467}{266112}$	$-\frac{2945}{1232}$	$-\frac{5610201}{14158144}$	$\frac{10513573}{3212352}$	$-\frac{424325}{205632}$	$\frac{376225}{454272}$	0	
	$\frac{821}{10800}$	0	$\frac{19683}{71825}$	$\frac{175273}{912600}$	$\frac{395}{3672}$	$\frac{785}{2704}$	$\frac{3}{50}$	
	$\frac{61}{864}$	0	$\frac{98415}{321776}$	$\frac{16807}{146016}$	$\frac{1375}{7344}$	$\frac{1375}{5408}$	$-\frac{37}{1120}$	$\frac{1}{10}$

Während man bei den bisher genannten Verfahren die Fehlerkonstanten des Verfahrens p -ter Ordnung minimiert, haben J.R. Dormand und P.J. Prince ein 7-stufiges Verfahrenspaar der Ordnungen 5(4) konstruiert, bei dem der Fehlerterm des genaueren Verfahrens 5. Ordnung minimiert wird und die Näherung η_n des "schlechteren" Verfahrens der Fehlerschätzung dient. Nachfolgend steht das Parameterschema dieses häufig benutzten Verfahrens DOPRI5(4), das wegen $a_{7,i} = b_i, i = 1(1)6$, besonders effizient ist. Weitere eingebettete Verfahren, z.B. DOPRI8(7), findet man in der angegebenen Literatur.

Eingebettetes RKV DOPRI5(4)

0								
$\frac{1}{5}$	$\frac{1}{5}$							
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$						
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$					
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$				
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$			
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$		
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0	
	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$	

5.4.3 Schrittweitenkontrolle mittels Schrittkenzahl

In numerischen Berechnungen beobachtet man, dass der Wert der partiellen Ableitung $f_x(t, x)$ großen Einfluss auf die Wahl der Schrittweite h hat. Große Beträge $|f_x|$ erfordern kleine h .

Für das KRKV kann man diese Ableitung näherungsweise aus den in diesem Verfahren berechneten Anstiegen k_1, k_2, k_3 erhalten. Es gilt nämlich bei Berücksichtigung der Lipschitz-Bedingung (4.18) für f

$$\begin{aligned} \kappa &= 2 \left| \frac{k_3 - k_2}{k_2 - k_1} \right| \\ &= 2 \left| \frac{f(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_2) - f(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_1)}{k_2 - k_1} \right| \\ &\approx 2 \left| \frac{\frac{h}{2}(k_2 - k_1)}{k_2 - k_1} \right| \left| \frac{\partial f}{\partial x} \right| = h \left| \frac{\partial f}{\partial x} \right|. \end{aligned} \quad (5.102)$$

Die Größe κ heißt Schrittkenzahl.

Zahlreiche Experimente führten auf die Faustregel, dass für moderate Genauigkeiten der Lösung der Wert κ zwischen den Grenzen 0.1 und 0.2 liegen sollte.

Die Kennzahl kann man natürlich im Laufe der Integration berechnen und kontrollieren. Mehr noch, es gelingt mittels geeigneter Wahl von h , den angegebenen Bereich 0.1..0.2 einzuhalten. Man könnte bei zu großem κ den RK-Schritt verwerfen und mit halber Schrittweite einen neuen Versuch machen, oder bei zu kleinem κ den RK-Schritt zwar akzeptieren, aber mit doppelter Schrittweite den nächsten Schritt durchführen. Zusätzliche Kontrollen dienen zur weiteren Verbesserung dieser dualen Strategie der Verdopplung bzw. Halbierung.

Kennt man die exakte Lösung des AWP, so bestimmt man natürlich aus der Beziehung (5.102) umgekehrt die günstige Schrittweite

$$h = h(t) = \frac{\kappa}{|f_x(t, x(t))|}, \quad \kappa \in [0.1, 0.2], \quad (5.103)$$

an jeder Stelle t des Lösungsintervalls.

Für lineare gDGL der Form

$$x' = \alpha x + \beta t + \gamma \quad (5.104)$$

mit reellen Koeffizienten gilt sofort $f_x = \alpha = \text{const}$ und

$$\kappa = 2 \left| \frac{k_3 - k_2}{k_2 - k_1} \right| = h|\alpha| = h|f_x|, \quad (5.105)$$

sodass eine konstante Schrittweite h im ganzen Intervall genommen werden kann. Sie wird lediglich durch die gewünschte Genauigkeit der Ordnung $\mathcal{O}(h^4)$ bestimmt.

Beispiel 5.9 Wir überprüfen die Bedingung (5.102) bei mehreren AWP.

(a) Gegeben sei das AWP

$$x' = f(t, x) = -x + t, \quad x(0) = 1, \quad t \in [t_0, T] = [0, 2]. \quad (5.106)$$

Die exakte Lösung ist $x(t) = 2e^{-t} + t - 1$. Die Schrittkenzahl beträgt

$$\kappa = 2 \left| \frac{k_3 - k_2}{k_2 - k_1} \right| = h|f_x| = h \cdot 1, \quad (5.107)$$

sodass man bei $\kappa = 0.2$ durchweg mit $h = 0.2$ rechnen kann.

(b) Für das AWP

$$x' = f(t, x) = tx, \quad x(0) = 1, \quad t \geq 0, \quad (5.108)$$

mit der stark wachsenden exakten Lösung $x(t) = e^{t^2/2}$ wird wegen $f_x(t, x) = t$ die Schrittkenzahl κ bei konstanter Schrittweite h stetig zunehmen.

Sei $[t_0, T] = [0, 2]$ und $N = 5$. Bei $h = 0.4$ erhalten wir mit dem KRKV die folgende Tabelle.

t(i)	x(i))	x'(i)		k1	k2	k3	k4		kappa	h f_x(i)
0.0	1.0000	0.0000		0.0000	0.2000	0.2080	0.4333		0.0800	0.0000
0.4	1.0833	0.4333		0.4333	0.7020	0.7342	1.1016		0.2400	0.1600
0.8	1.3771	1.1017		1.1017	1.5974	1.6966	2.4669		0.4000	0.3200
1.2	2.0542	2.4651		2.4651	3.5661	3.8744	5.7664		0.5600	0.4800
1.6	3.5950	5.7521		5.7521	8.5418	9.5461	14.8270		0.7200	0.6400
2.0	7.3787	14.7574		14.7574	22.7264	26.2327	42.8923		0.8800	0.8000

Tab. 5.9 Kontrolle der Schrittkenzahl κ bei konstantem $h = 0.4$

Damit sollte man die Schrittweite h im Laufe der Rechnung geeignet verkleinern. Für die Schrittweitenfolge $(h_0, h_1, \dots, h_{12}) = (0.4, 0.4, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)$,

$\sum_{k=0}^{12} h_k = 2$, bleibt die Schrittkenzahl κ jedoch nahe dem Wert 0.2.

t(i)	x(i))	x'(i)		k1	k2	k3	k4		kappa	h f_x(i)
0.00	1.0000	0.0000		0.0000	0.2000	0.2080	0.4333		0.0800	0.0000
0.40	1.0833	0.4333		0.4333	0.7020	0.7342	1.1016		0.2400	0.1600
0.80	1.3771	1.1017		1.1017	1.3385	1.3599	1.6491		0.1800	0.1600
1.00	1.6487	1.6487		1.6487	1.9949	2.0330	2.4663		0.2200	0.2000
1.20	2.0544	2.4653		2.4653	2.7221	2.7381	3.0266		0.1250	0.1200
1.30	2.3279	3.0263		3.0263	3.3470	3.3686	3.7307		0.1350	0.1300
1.40	2.6644	3.7301		3.7301	4.1338	4.1631	4.6210		0.1450	0.1400
1.50	3.0801	4.6202		4.6202	5.1323	5.1720	5.7557		0.1550	0.1500
1.60	3.5965	5.7545		5.7545	6.4090	6.4630	7.2128		0.1650	0.1600
1.70	4.2417	7.2109		7.2109	8.0540	8.1278	9.0981		0.1750	0.1700
1.80	5.0529	9.0953		9.0953	10.1893	10.2905	11.5558		0.1850	0.1800
1.90	6.0798	11.5516		11.5516	12.9819	13.1213	14.7838		0.1950	0.1900
2.00	7.3888	14.7776		14.7776	16.6618	16.8549	19.0560		0.2050	0.2000

Tab. 5.10 Ungefähres Einhalten der Schranken der Schrittkenzahl κ bei variablem h

(c) Für das AWP

$$x' = f(t, x) = \frac{1}{x + 10^{-p}}, \quad p = 1, \quad x(0) = 0, \quad t \in [t_0, T] = [0, 5], \quad (5.109)$$

mit der exakten Lösung $x(t) = -10^{-p} + \sqrt{10^{-2p} + 2t}$, $p = 1$, kann man mit der Ableitung $f_x(t, x) = -(x + 10^{-p})^{-2}$ bei $\kappa = 0.1$ die variable Schrittweite aus (2.63) theoretisch ermitteln.

$$\begin{aligned} h &= \frac{\kappa}{|f_x|} = 0.1 (x(t) + 10^{-p})^2 \\ &= 0.1 (-10^{-p} + \sqrt{10^{-2p} + 2t} + 10^{-p})^2 \\ &= 0.1 (10^{-2p} + 2t) \\ &= 0.2t + 10^{-2p-1}, \end{aligned} \quad (5.110)$$

was der Gestalt

$$h(t) = \alpha t + \beta, \quad \alpha = 0.2, \quad \beta = 10^{-2p-1}, \quad (5.111)$$

entspricht. Bei $p = 1$ ist $h(t) = 0.2t + 10^{-3}$.

Wegen $t_0 = 0$ gilt $h(t_0) = \beta$. Weiter sind

$$\begin{aligned} t_1 &= t_0 + h(t_0) = \beta, & h(t_1) &= \alpha t_1 + \beta = \beta(\alpha + 1), \\ t_2 &= t_1 + h(t_1) = \beta + \beta(\alpha + 1), & h(t_2) &= \alpha t_2 + \beta = \beta(\alpha + 1)^2, \\ t_3 &= t_2 + h(t_2) = \beta + \beta(\alpha + 1) + \beta(\alpha + 1)^2, & h(t_3) &= \alpha t_3 + \beta = \beta(\alpha + 1)^3, \\ &\dots & & \\ t_m &= \beta \sum_{i=0}^{m-1} (\alpha + 1)^i = \beta \frac{(\alpha + 1)^m - 1}{\alpha}, & h(t_m) &= \beta(\alpha + 1)^m. \end{aligned}$$

Der Faktor der Schrittweitenvergrößerung im Prozess beträgt $\alpha + 1$. Aus der Forderung

$$t_m = t_{m-1} + h(t_{m-1}) = T = 5$$

folgt

$$\beta \frac{(\alpha + 1)^m - 1}{\alpha} = T$$

bzw.

$$m = \frac{\lg(\frac{\alpha}{\beta}T + 1)}{\lg(\alpha + 1)} = \frac{\lg(\beta^{-1} + 1)}{\lg(\alpha + 1)} = \frac{\lg(10^{2p+1} + 1)}{\lg(1.2)} \geq 12.629 \cdot 10^{2p+1}. \quad (5.112)$$

Bei $p = 1$ und $h_0 = \beta = 10^{-3}$ ist $m = 37.893$. Somit braucht das KRKV 38 Schritte im Intervall $[0, 5]$. Es gilt

$$\begin{aligned} t_{37} &= 10^{-3} \frac{(1.2)^{37} - 1}{0.2} = 4.247\,811\,249, \\ h(t_{37}) &= 10^{-3} (1.2)^{37} = 0.850\,562\,249, \\ t_{38} &= 10^{-3} \frac{(1.2)^{38} - 1}{0.2} = 5.098\,373\,499. \end{aligned}$$

i	t(i)	h(i)		i	t(i)	h(i)
0	0.0000	0.0010		20	0.1867	0.0383
1	0.0010	0.0012		21	0.2250	0.0460
2	0.0022	0.0014		22	0.2710	0.0552
3	0.0036	0.0017		23	0.3262	0.0662
4	0.0054	0.0021		24	0.3925	0.0795
5	0.0074	0.0025		25	0.4720	0.0954
6	0.0099	0.0030		26	0.5674	0.1145
7	0.0129	0.0036		27	0.6819	0.1374
8	0.0165	0.0043		28	0.8192	0.1648
9	0.0208	0.0052		29	0.9841	0.1978
10	0.0260	0.0062		30	1.1819	0.2374
11	0.0322	0.0074		31	1.4193	0.2849
12	0.0396	0.0089		32	1.7041	0.3418
13	0.0485	0.0107		33	2.0459	0.4102
14	0.0592	0.0128		34	2.4561	0.4922
15	0.0720	0.0154		35	2.9483	0.5907
16	0.0874	0.0185		36	3.5390	0.7088
17	0.1059	0.0222		37	4.2478	0.8506
18	0.1281	0.0266		38	5.0984	1.0207
19	0.1547	0.0319		39	6.1190	1.2248

Tab. 5.11 Entwicklung der variablen Schrittweite h bei Berücksichtigung der Schrittkenzahl $\kappa = 0.1$

Bei großem p ist $|f_x|$ anfänglich sehr groß ($f_x(0,0) = -10^{2p}$) und die Anfangsschrittweite h muss dann klein sein, sodass insgesamt eine hohe Schrittzahl m folgt.

Die Schrittkenzahl ist nicht in jedem Fall für eine Schrittweitensteuerung geeignet. Es gibt Beispiele wie

$$x' = f(t, x) = 10\left(x - \frac{t^2}{1+t^2}\right) + \frac{2t}{(1+t^2)^2}, \quad t \in [t_0, t_1] = [0, 5], \quad x(t_0) = x_0,$$

wo $|f_x| = 10$ ist und $\kappa = 2|k_3 - k_2|/|k_2 - k_1|$ konstant bleibt. Man möchte dann eine Rechnung mit einer konstanten Schrittweite $h = \kappa/|f_x|$ machen. Die exakte beschränkte Lösung $x(t) = x_0 e^{10t} + t^2/(1+t^2)$ mit $x_0 = 0$ hat aber instabiles Verhalten und bei $x_0 \neq 0$ "explodiert" sie gegen $+\infty$ oder $-\infty$. Somit ist bei konstanter Schrittweite in einem größeren Intervall keine gute Näherungslösung zu erwarten (vergl. auch Beispiel 5.10 (c)).

5.4.4 Schrittweitensteuerung mit Grob- und Feinrechnung

Hierbei betrachten wir nur ein ESV und seinen Diskretisierungsfehler.

Damit verbunden sind solche Begriffe wie asymptotische Fehlerschätzung, Extrapolation, adaptives Prinzip bzw. Runge-Prinzip.

Analogien gibt es auch zur numerischen Integration mit Grob- und Feinrechnung. Dort macht man für die näherungsweise Berechnung des bestimmten Integrals I unter Verwendung einer Quadraturformel folgenden Zugang.

Mit der Schrittweite $2h$ liefert die Quadraturformel die Näherung I_G zu I , das ist die Grobrechnung. Bei der Schrittweite h liefert sie die Näherung I_F , das ist die Feinrechnung und entspricht dem Kontrollmechanismus.

Mit den beiden Approximationen I_G und I_F lässt sich eine ungefähre Angabe über den Fehler der Feinrechnung

$$F_F = I - I_F$$

machen. Damit kann $\tilde{I} = I_F + F_F$ für eine bessere Approximation mit höherer Genauigkeitsordnung verwendet werden.

Wir wollen hier mit dem ESV vorab nicht zwei Näherungen η_i und $\hat{\eta}_i$ im gesamten Lösungsintervall bei konstanten Schrittweiten $2h$ bzw. h ermitteln und dann eine Verbesserung vornehmen. Im Zuge der schrittweisen Berechnung soll eine lokale Betrachtung des Diskretisierungsfehlers sowohl für die Korrektur von Näherungslösungen als auch zur SWS durchgeführt werden.

Dazu nutzen wir den globalen Diskretisierungsfehler e_i und die Fehlerordnung p des s -stufigen expliziten RKV in der Darstellung

$$e_i = e(t_i, h) = \eta_i - x(t_i; t_0, x_0) = \eta_i - \Psi^{t_i, t_0} x_0 = Ch^p + \mathcal{O}(h^{p+1}). \quad (5.113)$$

Angenommen der Koeffizient C sei ungefähr konstant. Mit der Berechnung von Ch^p gelangt man zu der gewünschten Genauigkeitsverbesserung und zu einer Kenngröße für die SWS.

Das Runge-Prinzip beinhaltet Folgendes.

Sei auf dem Teilintervall $[t_i, t_{i+2}]$ der Wert η_{i+2} das Ergebnis der Grobrechnung mit einem $2h$ -Schritt des ESV und $\hat{\eta}_{i+1}$, $\hat{\eta}_{i+2}$ die Ergebnisse der Feinrechnung zweier h -Schritte. Dann kann man mit $x_0 = \eta_0$ schreiben

$$e_{i+2} = e(t_{i+2}, 2h) = \eta_{i+2} - x(t_{i+2}; t_0, x_0) = C(2h)^p + \mathcal{O}(h^{p+1}), \quad (5.114)$$

$$\hat{e}_{i+2} = \hat{e}(t_{i+2}, h) = \hat{\eta}_{i+2} - x(t_{i+2}; t_0, x_0) = Ch^p + \mathcal{O}(h^{p+1}). \quad (5.115)$$

Damit gelten

$$\begin{aligned} \eta_{i+2} &= x(t_{i+2}; t_0, x_0) + C(2h)^p + \mathcal{O}(h^{p+1}), \\ \hat{\eta}_{i+2} &= x(t_{i+2}; t_0, x_0) + Ch^p + \mathcal{O}(h^{p+1}), \\ \eta_{i+2} - \hat{\eta}_{i+2} &= C(2^p - 1)h^p + \mathcal{O}(h^{p+1}), \\ \frac{\eta_{i+2} - \hat{\eta}_{i+2}}{2^p - 1} &= Ch^p + \mathcal{O}(h^{p+1}) \\ &= \hat{\eta}_{i+2} - x(t_{i+2}; t_0, x_0) + \mathcal{O}(h^{p+1}) \\ &= \hat{e}_{i+2} + \mathcal{O}(h^{p+1}). \end{aligned}$$

Somit erhalten wir eine erste Beziehung für den Fehler der Feinrechnung

$$\hat{e}_{i+2} = \frac{\eta_{i+2} - \hat{\eta}_{i+2}}{2^p - 1} + \mathcal{O}(h^{p+1}). \quad (5.116)$$

Daraus folgt auch eine zweite für den Lösungswert

$$x(t_{i+2}; t_0, x_0) = \hat{\eta}_{i+2} - \hat{e}_{i+2} = \hat{\eta}_{i+2} + \frac{\hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1} + \mathcal{O}(h^{p+1}) = \frac{2^p \hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1} + \mathcal{O}(h^{p+1}). \quad (5.117)$$

Die Formel (5.117) bedeutet eine Linearkombination von Grob- und Feinrechnung und führt zu einem verbesserten Näherungswert für $x(t_{i+2}; t_0, x_0)$. Die Feinrechnung hat dabei ein größeres Gewicht.

Bemerkung 5.4

(a) Beide Beziehungen (5.116) und (5.117) stellen asymptotische Fehlerschätzungen dar, d.h. je kleiner die Schrittweite h , desto genauer wird der Wert \hat{e}_{i+2} bzw. $x(t_{i+2}; t_0, \eta_0)$ approximiert, falls die Rundungsfehler vernachlässigt werden können und die GKA hinreichend stark ist.

(b) Für den Fehler der Feinrechnung \hat{e}_{i+2} liefert die Größe

$$est = \frac{\hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1} \quad (est = estimate) \quad (5.118)$$

i. Allg. eine gute Näherung.

Zugleich finden wir aus (5.117) eine weitere Formel, die den Fehler des RKV beschreibt, nämlich

$$R(h) = \frac{1}{h} (x(t_{i+2}; t_i, \eta_i) - \hat{\eta}_{i+2}) = \frac{1}{h} \frac{\hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1}. \quad (5.119)$$

$R(h)$ ist ein asymptotischer Fehlerschätzer der Ordnung p für den lokalen Diskretisierungsfehler τ_{i+2} des RKV.

(c) Das Runge-Prinzip beinhaltet eine Extrapolation.

Den aus Fein- und Grobrechnung zusammengesetzten oder extrapolierten Wert $\tilde{\eta}_{i+2}$ findet man in verschiedenen Darstellungen als

$$\tilde{\eta}_{i+2} = \frac{2^p \hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1} = \hat{\eta}_{i+2} + \frac{\hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1}. \quad (5.120)$$

Er ist eine bessere Näherung als der Feinwert $\hat{\eta}_{i+2}$ und zwar um eine Genauigkeitsordnung.

Beim KRKV (5.57) haben wir $p = 4$ und

$$\begin{aligned} \eta_{i+2} &= \eta_i + 2h\Phi(t_i, \eta_i, 2h) = \eta_i + K, \\ \hat{\eta}_{i+1} &= \eta_i + h\Phi(t_i, \eta_i, h) = \eta_i + k', \\ \hat{\eta}_{i+2} &= \hat{\eta}_{i+1} + h\Phi(t_{i+1}, \hat{\eta}_{i+1}, h) = \hat{\eta}_{i+1} + k'' = \eta_i + k' + k'', \end{aligned}$$

mit den Zuwächsen K, k', k'' . Damit ist der Fehlerterm

$$est = \frac{k' + k'' - K}{15}. \quad (5.121)$$

Dieser wird zur dualen SWS verwendet. Ist bei gegebener Toleranz $\varepsilon > 0$ der Fehler $|est| > \varepsilon$, wird der Schritt verworfen und mit halber Schrittweite wiederholt. Sonst wird er akzeptiert und bei Unterschreitung einer unteren Schranke die folgende Schrittweite verdoppelt. Die untere Schranke $\varepsilon/50$ vermeidet ein ständiges abwechselndes Verdoppeln und Halbieren der Schrittweite. Der Faktor $1/50$ ist gleichzeitig ein aus vielen Experimenten gewonnener Wert.

(d) Die Anpassung der Schrittweiten und damit der Gitterstruktur während des Lösungsprozesses heißt, ausgehend von (t_i, η_i) eine neue Schrittweite h_i festzulegen, um dann η_{i+1} an der Stelle $t_{i+1} = t_i + h_i$ zu bestimmen.

Das Ziel der SWS kann nun so formuliert werden: Wähle h_i , sodass die nächste Näherung eine vorgegebene Fehlertoleranz erfüllt.

Im Gegensatz zum Bisherigen steuern die gängigen Verfahren nur den lokalen Fehler und nicht den globalen. Die Schrittweite wird so gewählt, dass für den aktuellen Schritt

$$|\eta_{i+1} - x(t_{i+1}; t_i, \eta_i)| \leq \varepsilon$$

gilt mit vorgegebener Toleranz ε . Die exakte Lösung $x(t_{i+1}; t_i, \eta_i)$, also $x(t)$ zur AB $x(t_i) = \eta_i$ ist hier natürlich nicht bekannt und muss i. Allg. durch eine Approximation ersetzt werden. Man spricht deswegen von der Fehlerschätzung und kommt so zu der im Abschnitt 5.4.1 empfohlenen Vorgehensweise.

Wir wollen aber wieder zur Grob- und Feinrechnung im Teilintervall $[t_i, t_i + 2h]$ zurückkehren. Dort soll der Koeffizient C lokal konstant sein, was für ein kleines Intervall angenommen werden kann. Analog zu den Formeln (5.114), (5.115) sei η_{i+2} das Ergebnis der Grobrechnung mit einem $2h$ -Schritt des ESV und $\hat{\eta}_{i+1}$, $\hat{\eta}_{i+2}$ die Ergebnisse der Feinrechnung zweier h -Schritte.

Dann kann man mit $x_i = \eta_i$ schreiben

$$\begin{aligned} e'_{i+2} &= e'(t_{i+2}, 2h) = \eta_{i+2} - x(t_{i+2}; t_i, \eta_i) = C(2h)^{p+1} + \mathcal{O}(h^{p+2}), \\ \hat{e}'_{i+2} &= \hat{e}'(t_{i+2}, h) = \hat{\eta}_{i+2} - x(t_{i+2}; t_i, \eta_i) = 2Ch^{p+1} + \mathcal{O}(h^{p+2}). \end{aligned}$$

Damit gelten

$$\begin{aligned} \eta_{i+2} &= x(t_{i+2}; t_i, \eta_i) + C(2h)^{p+1} + \mathcal{O}(h^{p+2}), \\ \hat{\eta}_{i+2} &= x(t_{i+2}; t_i, \eta_i) + 2Ch^{p+1} + \mathcal{O}(h^{p+2}), \\ \frac{\eta_{i+2} - \hat{\eta}_{i+2}}{2^p - 1} &= 2Ch^{p+1} + \mathcal{O}(h^{p+2}) = \hat{\eta}_{i+2} - x(t_{i+2}; t_i, \eta_i) + \mathcal{O}(h^{p+2}) = \hat{e}'_{i+2} + \mathcal{O}(h^{p+2}). \end{aligned}$$

Daraus folgt für den Lösungswert

$$x(t_{i+2}; t_i, \eta_i) = \hat{\eta}_{i+2} - \hat{e}'_{i+2} = \hat{\eta}_{i+2} + \underbrace{\frac{\hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1}}_{\substack{est \\ \tilde{\eta}_{i+2}}} + \mathcal{O}(h^{p+2}).$$

Das entspricht dem Ergebnis der Formel (5.117). Damit treffen auch die bisherigen Bemerkungen zum Runge-Prinzip und zur SWS zu.

Im folgenden Flussdiagramm zum KRKV für das skalare AWP

$$x' = f(t, x), \quad x(t_0) = x_0, \quad t \in [t_0, T],$$

wird die Prozedur $KRKV_S(\)$ mit entsprechenden Parametern für den RK-Schritt verwendet.

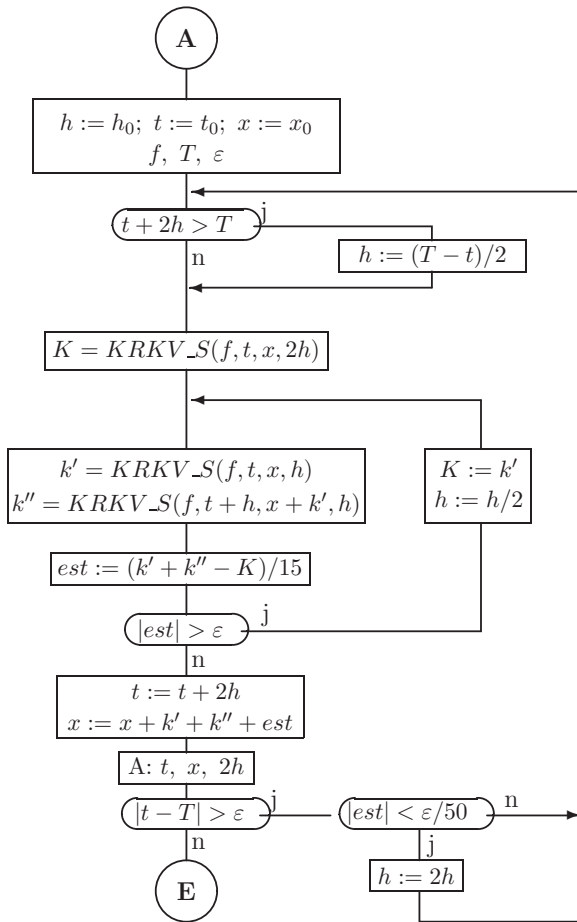


Abb. 5.11 Datei *dsws_rkv.pic*,
Flussdiagramm zum KRKV mit dualer SWS

Die duale SWS mit Verdoppeln und Halbieren der Schrittweite ist überschaubar, aber natürlich noch nicht optimal bzgl. einer möglichst kleinen Anzahl der Funktionswertberechnungen. Die Voraussage einer variablen und damit besser angepassten Schrittweite erfordert jedoch einen aufwändigeren Steuermechanismus.

Folgende MATLAB-Funktionen liegen dazu vor. Dabei kann die Schrittprozedur entweder den Näherungswert am nächsten Knoten bzw. nur den Zuwachs an dieser Stelle zurückgeben.

```
% Datei krkv_s1.m
% Ein Schritt des KRKV --> y=x(i+1)

function y = krkv_s1(f,ti,xi,h);
    k1 = feval(f,ti,xi);
    k2 = feval(f,ti+0.5*h,xi+0.5*h*k1);
    k3 = feval(f,ti+0.5*h,xi+0.5*h*k2);
    k4 = feval(f,ti+h,xi+h*k3);
    y = xi+h/6*(k1+2*k2+2*k3+k4);
% -----

% Datei krkv_s2.m
% Ein Schritt des KRKV --> z=Zuwachs

function z = krkv_s2(f,ti,xi,h);
    k1 = feval(f,ti,xi);
    k2 = feval(f,ti+0.5*h,xi+0.5*h*k1);
    k3 = feval(f,ti+0.5*h,xi+0.5*h*k2);
    k4 = feval(f,ti+h,xi+h*k3);
    z = h/6*(k1+2*k2+2*k3+k4);
```

Um eine Analogie zum Flussdiagramm in Abb. 5.11 zu haben, verwenden wir im Steuermechanismus die Funktion `krkv_s2`. Dazu sind alle Knoten t_i , die Näherungswerte $x(i) = \eta_i$ sowie ihre nicht äquidistanten Abstände $2h_i$ Vektoren und Ergebnisparameter der Funktion `krkv1_sw`, um damit auch verschiedene Auswertungen vorzunehmen.

```
% Datei krkv1_sw.m
% Explizites Klassisches Runge-Kutta-Verfahren, KRKV fuer gDGL
% mit dualer Schrittweitensteuerung
% x'=f(t,x), x(t0)=x0, t in [t0,T], h0 Anfangsschrittweite
% --> ein Schritt: krkv_s1.m oder krkv_s2.m

function [t,x,h,anz] = krkv1_sw(f,t0,T,x0,h0,eps)
    hi = h0;
    ti = t0;
    xi = x0;
    t(1) = t0;
    x(1) = x0;
    anz = 0;

    for i = 1:1000000 % obere Schranke hinreichend gross waehlen
        if ti+2*hi>T
            hi = (T-ti)/2;
        end;
        K = krkv_s2(f,ti,xi,2*hi);
        while 1>0,
            ks = krkv_s2(f,ti,xi,hi);
            kss = krkv_s2(f,ti+hi,xi+ks,hi);
            est = (ks+kss-K)/15;
            if abs(est)<=eps
                break;
            end;
        end;
        t(i+1) = ti+hi;
        x(i+1) = x(i) + est;
        anz = anz + 1;
        ti = t(i+1);
        xi = x(i+1);
        hi = hi/2;
    end;
end;
```

```

else
    hi = 0.5*hi;
    K = ks;
    anz = anz+1;
end;
end;
h(i) = 2*hi;
ti = ti+2*hi;
xi = xi+ks+kss+est;
t(i+1) = ti;
x(i+1) = xi;
if abs(ti-T)<=eps
    break
elseif abs(est)<0.02*eps
    hi = 2*hi;
else
end;
end;
h(i+1) = 2*hi;

```

Ein kleines Problem bei der SWS ist die Vorgabe einer guten Anfangsschrittweite h_0 , um zu Beginn der Rechnung sowohl unnötige Halbierungen als auch mehrere Verdopplungsvorschläge zu vermeiden. Oft bleibt jedoch nur die Wahl von $h_0 = (T - t_0)/2$.

In `krkv1_sw` ist die Anzahl der durchgeführten Schritte mittels der Länge der Ergebnisvektoren gemäß `max(size(t))` abfragbar.

Für Testzwecke haben wir noch einen Zähler `anz` für die Anzahl der nicht akzeptierten Schritte eingebaut, den man später wieder entfernen kann.

Bei Nichtakzeptanz der Schrittweite kann man aber das Ergebnis des Zwischenschrittes für die Wiederholung des Schrittes "retten".

Mögliche Aufrufe der Funktion in einem Rahmenprogramm wären z.B.

```
[t,x,h,anz] = krkv1_sw('f1',t0,T,x0,h0,eps);
```

```
[t,x,h,anz] = krkv1_sw('f1',0,5,0,0.00025,10^(-16));
```

Beispiel 5.10

(a) Wir überprüfen anhand des AWP der Riccati-DGI

$$x' = t - x^2, \quad x(0) = 0, \quad t \in [t_0, T] = [0, 1], \quad (5.122)$$

dass der lokale sowie globale Diskretisierungsfehler des KRKV

$$\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h), \quad i = 0, 1, \dots, N-1, \quad h = (T - t_0)/N = 1/N,$$

von der Ordnung $p = 4$ ist.

Bei einer Schrittweithalbierung sollte sich der Fehler rund 16 Mal verkleinern.

In MATLAB bei der GKA *double* lösen wir das AWP für $N = 5, 10, 20, \dots$ Schritte.

Die Ergebnisse η_i , $i = 0, 1, \dots, N$, und entsprechenden Fehler sind in einer Tabelle zusammengestellt. Für große N wird die Akkumulation der Rundungsfehler sichtbar.

N	h	η_N ($t_N = 1$)	Globaler Fehler $e_N(h)$	$\frac{ e_N(h) }{ e_{2N}(h/2) }$
5	0.2	0.4555 3348 8111	-0.0000 1102 7969	16.1374
10	0.1	0.4555 4384 2704	-0.0000 0068 3376	16.0947
20	0.05	0.4555 4448 3621	-0.0000 0004 2459	16.0596
40	0.025	0.4555 4452 3437	-0.0000 0000 2643	16.1092
80	0.0125	0.4555 4452 5916	-0.0000 0000 0164	17.4061
160	0.0062 5	0.4555 4452 6071	-0.0000 0000 0009 42	40.9884
320	0.0031 25	0.4555 4452 6081 23	-0.0000 0000 0000 64	0.2760
640	0.0015 625	0.4555 4452 6081 83	-0.0000 0000 0000 04	-
	$x(1)$	0.4555 4452 6081 8735		16

Tab. 5.12 Ergebnisse η_N bei $t_N = 1$ für $N = 5, 10, 20, 40, 80, 160, 320, 640$ sowie Relation der globalen Fehler $e_N = \eta_N - x(t_N)$ bei Schrittweithalbung

(b) Gegeben sei das AWP

$$x' = f(t, x) = t \cos(x), \quad x(-2) = 1, \quad t \in [t_0, T] = [-2, 5], \quad (5.123)$$

mit der exakten Lösung

$$x(t) = \arctan \left(\frac{(1 + \sin(1)) e^{t^2-4} - 1 + \sin(1)}{(1 + \sin(1)) e^{t^2-4} + 1 - \sin(1)}, \frac{2 \cos(1) e^{t^2/2-2}}{(1 + \sin(1)) e^{t^2-4} + 1 - \sin(1)} \right). \quad (5.124)$$

Das Richtungsfeld, die Trajektorie $x(t)$ und Näherungslösungen mit ESV haben wir im Bsp. 5.8 (a) demonstriert. Nun vergleichen wir die Fehler des KRKV für verschiedene Werte N .

N	η_N	$ e_N = \eta_N - x(5) $	$\max_{i=0(1)N} \eta_i - x(t_i) $
7	2.8087 3448 8267	1.2379 5432 0301	1.2379 5432 0301
14	1.5698 9151 6710	0.0008 8865 1255	0.0076 0346 3175
28	1.5707 7650 2388	0.0000 0366 5577	0.0003 0295 4367
56	1.5707 8003 4019	0.0000 0013 3946	0.0000 1501 4776
112	1.5707 8016 1308	0.0000 0000 6658	0.0000 0083 1907
224	1.5707 8016 7594	0.0000 0000 0371	0.0000 0004 8904
70	1.5707 8011 7959	0.0000 0005 0007	0.0000 0584 0025
700	1.5707 8016 7962	0.0000 0000 0004	0.0000 0000 0491
7 000	1.5707 8016 7966 22	0.0000 0000 0000	0.0000 0000 0000 049
14 000	1.5707 8016 7966 22	0.0000 0000 0000 0011	0.0000 0000 0000 0051
35 000	1.5707 8016 7966 22	0.0000 0000 0000 0011	0.0000 0000 0000 0042
70 000	1.5707 8016 7966 22	0.0000 0000 0000 0015	0.0000 0000 0000 0093
$x(5)$	1.5707 8016 7966 22		

Tab. 5.13 Fehler bei $t = 5$ und maximale Fehler im Intervall $[-2, 5]$

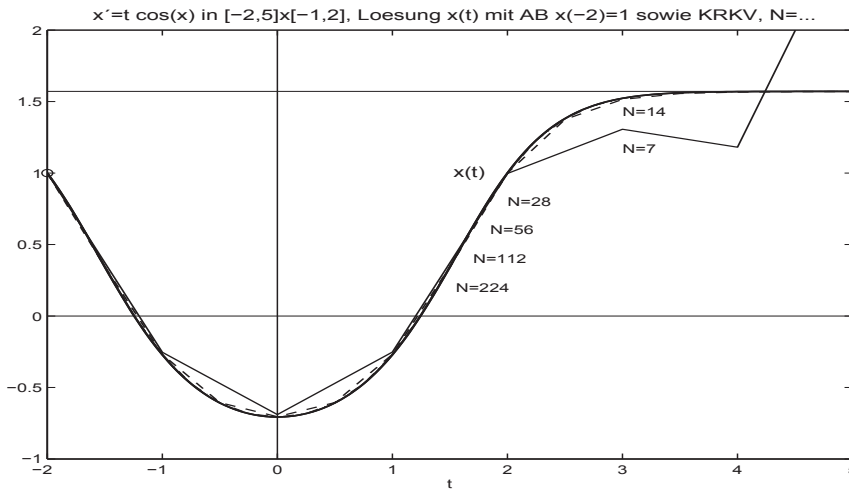


Abb. 5.12 Datei *r_f12f.ps*, Näherungslösungen zur gDGl $x' = t \cos(x)$, $x(-2) = 1$, mit KRKV bei verschiedenen N

Die Näherungslösungen zeigen ein moderates Verhalten.

Mit der Schrittweite $h = 1$ ($N = 7$) ist natürlich kein vernünftiges Ergebnis zu erwarten. Andererseits stößt man mit der Schrittweite $h = 10^{-3}$ und der Verfahrensordnung $\mathcal{O}(h^4)$ schon fast an die MATLAB-Genauigkeit *double*. Außerdem wachsen mit N die Rundungsfehler an.

Wir verwenden nun das KRKV mit SWS zu verschiedenen Toleranzen $\varepsilon = 10^{-k}$ und nehmen dazu eine möglichst große und runde Anfangsschrittweite h_0 , wo das erste Teilintervall $[t_0, t_0 + 2h_0]$ und weitere solche der Länge $2h_0$ akzeptiert werden. Dann werten wir die Ergebnisse unter folgenden Gesichtspunkten aus.

ε	h_0	1. Schritt mit $2h_0$	Anzahl Schritte	Anzahl n. akz.	$2h_{max}$	$2h_{min}$	$\max_i \eta_i - x(t_i) $
10^{-3}	0.50	1.00	11	1	1.00	0.50	1.094e-03
	0.55	n. akz.					
10^{-4}	0.20	0.40	21	2	0.80	0.20	1.655e-05
	0.25	n. akz.					
10^{-5}	0.25	0.50	31	4	0.50	0.125	1.796e-06
	0.30	n. akz.					
10^{-6}	0.15	0.30	44	2	0.30	0.10	2.960e-07
	0.20	n. akz.					
10^{-7}	0.05	0.10	72	2	0.20	0.05	1.992e-08
	0.075	n. akz.					
10^{-8}	0.030	0.060	104	1	0.120	0.040	2.693e-09
	0.035	n. akz.					
10^{-9}	0.020	0.040	175	3	0.080	0.020	2.096e-10
	0.025	n. akz.					

ε	h_0	1. Schritt mit $2h_0$	Anzahl Schritte	Anzahl n. akz.	$2h_{max}$	$2h_{min}$	$\max_i \eta_i - x(t_i) $
10^{-10}	0.013	0.026	281	3	0.052	0.013	1.525e-11
	0.014	n. akz.					
10^{-11}	0.008	0.016	434	3	0.064	0.008	2.054e-12
	0.009	n. akz.					
10^{-12}	0.0050	0.010	620	2	0.040	0.010	3.281e-13
	0.0055	n. akz.					
10^{-13}	0.0030	0.006	1 012	2	0.024	0.006	3.064e-14
	0.0035	n. akz.					
10^{-14}	0.0020	0.004	1 550	2	0.016	0.004	4.218e-15
	0.0025	n. akz.					
10^{-15}	0.0013	0.0026	2 778	3	0.0104	0.0013	3.469e-14
	0.0014	n. akz.					
10^{-16}	0.0008	0.0016	4 285	4	0.0064	0.0008	5.373e-14
	0.0009	n. akz.					
10^{-17}	0.0005	0.0010	6 161	2	0.0040	0.0010	1.572e-13
	0.0006	n. akz.					

Tab. 5.14 Ergebnisse zum KRKV mit SWS bei gegebener Toleranz ε
(n. akz. = nicht akzeptierter Schritte)

Die Genauigkeitsordnung des Verfahrens ist durch die Korrektur von der Ordnung $\mathcal{O}(h^5)$ und wird erreicht. Bei Rechnungen nahe der Genauigkeit der GKA *double* treten sehr große Schrittzahlen auf und mit den dabei in jedem akzeptierten Schritt gemachten zwölf Funktionswertberechnungen machen sich Rundungsfehler bemerkbar. Diese lassen dann keine Verbesserung der Lösungsgenauigkeit mehr zu.

Nun noch zum Vergleich zwischen KRKV ohne und mit SWS.

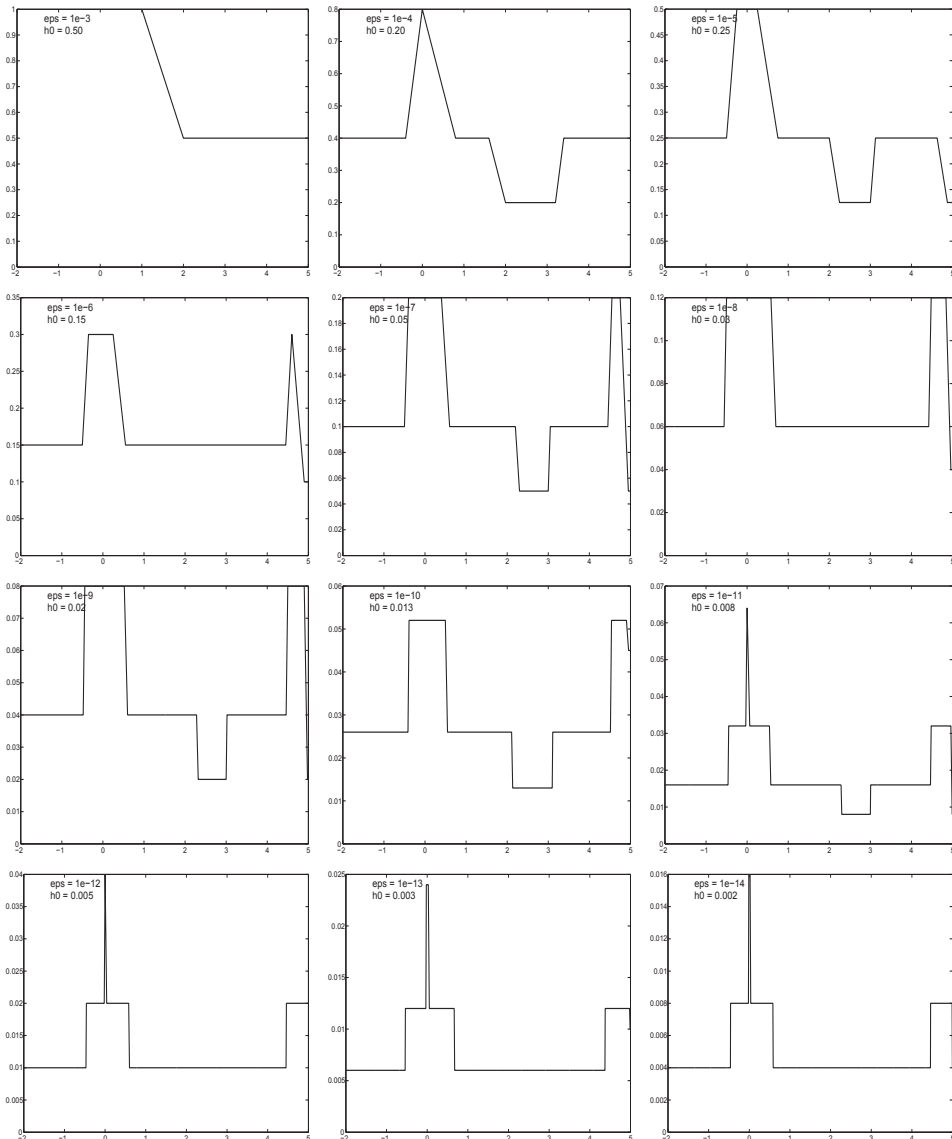
Wir wählen Varianten bei ähnlicher Lösungsgenauigkeit $\max_i |\eta_i - x(t_i)|$ aus und untersuchen dabei die Schrittweiten h und die Anzahl *anzf* der Funktionswertberechnungen.

$\max_i \eta_i - x(t_i) $	KRKV ohne SWS			KRKV mit SWS		
	N	h	<i>anzf</i>	a/n.S.	h	<i>anzf</i>
$\mathcal{O}(10^{-5})$	56	0.125	224	21/2	0.1...0.4	268
$\mathcal{O}(10^{-8})$	224	0.03125	896	72/2	0.025...0.1	880
$\mathcal{O}(10^{-10})$	700	0.01	2 800	175/3	0.01...0.04	2 124
$\mathcal{O}(10^{-14})$	7 000	0.001	28 000	1 012/2	0.003...0.012	12 160
$\mathcal{O}(10^{-15})$	14 000	0.0005	56 000	1 550/2	0.002...0.008	18 616

Tab. 5.15 Vergleich von KRKV ohne und mit SWS
(a/n.S. = Anzahl der akzeptierten bzw. nicht akzeptierten Schritte)

Für geringe Genauigkeitsanforderungen reicht das KRKV ohne SWS aus und ist schneller. Das Verhältnis ändert sich jedoch mit kleiner werdender Toleranz ε zu Gunsten des KRKV mit SWS, wo der Bereich der Schrittlängen nicht so klein wird.

Bei der Betrachtung der akzeptierten Schrittweiten $2h_i$ versus t_i ergeben sich zwei auffällige Situationen. Bei flachem Lösungsverlauf nahe $t = 0$ und bei $t > 3$ erfolgt eine Schrittweitenvergrößerung, die dann aber bei t rechts von Null wieder schnell auf das Niveau wie zu Beginn der Rechnung "zurückgenommen" werden muss. Im Verlauf einer Phase eines starken Funktionsanstiegs bzw. Krümmung kann es, wie zwischen $t = 2$ und $t = 3$, zu einer weiteren Schrittweitenverkleinerung kommen.



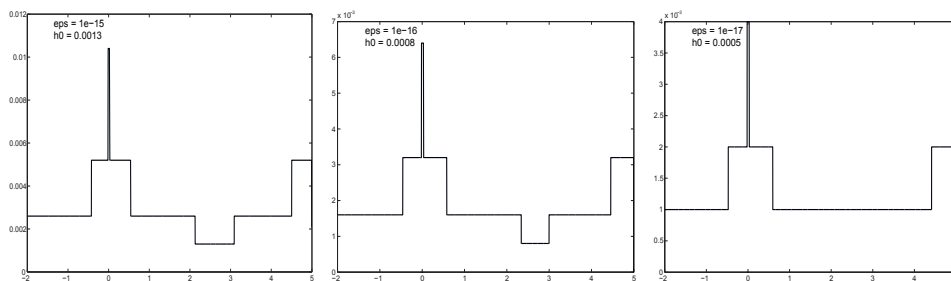


Abb. 5.13 Dateien *r_f12g3.ps*, *r_f12g4.ps*, ..., *r_f12g17.ps* (v.l.o. n.r.u.), Schrittweitenverhalten $2h_i(t_i)$ beim KRKV mit SWS bei $\varepsilon = 10^{-k}$, $k = 3, 4, \dots, 17$, und geeigneter Wahl von h_0

(c) Gegeben sei das AWP

$$x' = f(t, x) = 10 \left(x - \frac{t^2}{1+t^2} \right) + \frac{2t}{(1+t^2)^2}, \quad t \in [t_0, T] = [0, 5], \quad x(t_0) = x_0. \quad (5.125)$$

Die allgemeine Lösung der gDGl ist

$$x(t) = Ce^{10t} + \frac{t^2}{1+t^2}, \quad (5.126)$$

die exakte Lösung des AWP mit der AB $x(t_0) = x_0$ lautet

$$x(t) = x_0 e^{10t} + \frac{t^2}{1+t^2}. \quad (5.127)$$

Zunächst interessieren wir uns für das Richtungsfeld zur gDGl und AWP mit ausgewählten AB. Wir nutzen MATLAB mit der Funktion `dfield` aus Abschnitt 4.1.

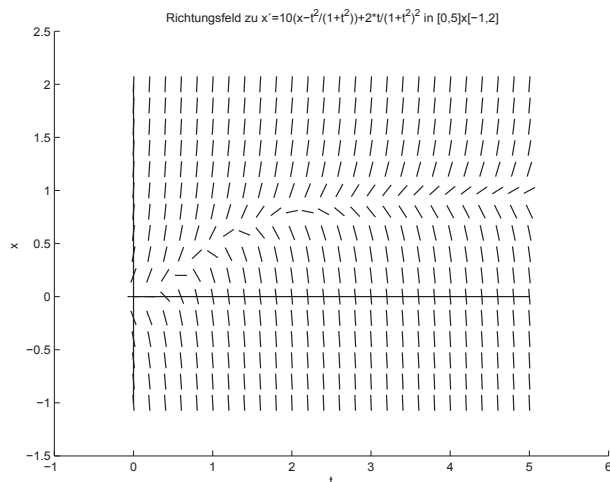


Abb. 5.14

Datei *r_f1a.ps*,
Richtungsfeld zur gDGl
 $x' = f(t, x)$ (5.125)
in $[0, 5] \times [-1, 2]$

Wir zeichnen das Richtungsfeld und tragen Trajektorien (5.127) mit gegebenen AB ein.

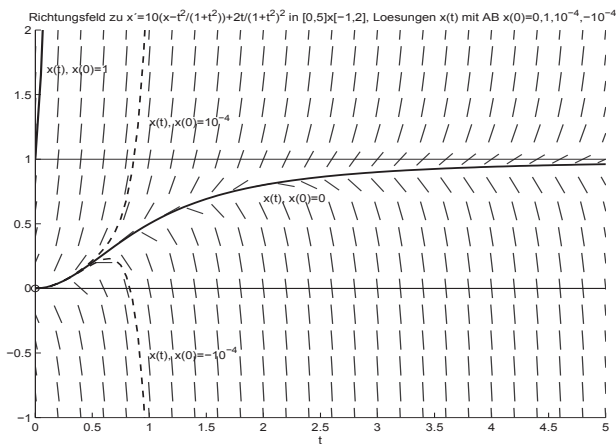


Abb. 5.15

Datei *r_f1b.ps*,
 Richtungsfeld zur gDGL
 $x' = f(t, x)$ (5.125)
 sowie Lösungen $x(t)$
 mit den AB $x(0) = 0,$
 $x(0) = 1, 10^{-4}, -10^{-4}$

Dabei bemerkt man, dass die Lösungskurve $x(t) = t^2/(1 + t^2)$ mit der AB $x(0) = 0$ für $t \rightarrow \infty$ gegen die Asymptote $x = 1$ strebt. Aber sie ist eine instabile Kurve (instabile Gleichgewichtslage), weil bei noch so geringfügiger Störung der AB die Nachbartrajektorien sich bald von ihr entfernen. Ist $x_0 > 0$, so tendiert $x(t)$ schnell gegen unendlich. Somit ist es interessant, wie sich die ESV verhalten, die in der Nachbarschaft zu $x(t)$ arbeiten.

Wir vergleichen also mehrere ESV $\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h), i = 0, 1, \dots, N - 1,$ bei Wahl einer kleinen konstanten Schrittweite $h = (T - t_0)/N = 5/N, N = 1000.$

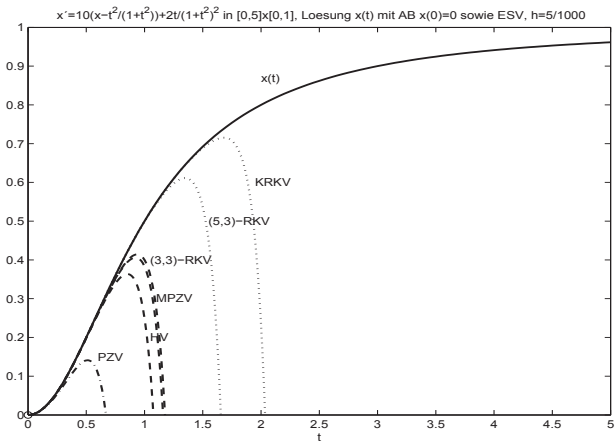


Abb. 5.16

Datei *r_f1c.ps*,
 gDGL $x' = f(t, x), t \in [0, 5],$
 und Lösung $x(t)$
 mit den AB $x(0) = 0,$
 sowie Näherungslösungen
 mittels ESV bei $N = 1000$

Keines des ESV kann über ein längeres Lösungsintervall an der exakten Lösung “dran bleiben“. Würden wir die Schrittkenzahl κ (5.102) zum KRKV auswerten, so kann man nur konstant bleibende Werte erkennen, sodass die im Abschnitt 5.4.3 beschriebene Methode nicht greift.

Wir vergleichen noch Rechnungen mit dem KRKV bei verschiedenen Schrittweiten $h = 5/N$.

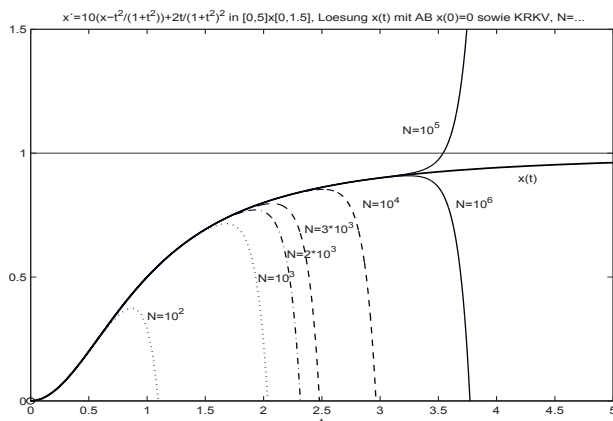


Abb. 5.17

Datei *r_f1d.ps*,
 gDGI $x' = f(t, x)$, $t \in [0, 5]$,
 und Lösung $x(t)$
 mit den AB $x(0) = 0$,
 sowie Näherungslösungen
 mit KRKV bei $N = 10^2, 10^3$,
 $N = 2 \cdot 10^3, 3 \cdot 10^3, 10^4, 10^5, 10^6$

Damit ist man eigentlich geneigt, die SWS mit Grob-/Feinrechnung zu verwenden. Der Funktionsaufruf ist

```
h0 = 0.00025
eps = 10^(-16)
[t,x,h,anz] = krkv1_sw('f1',t0,T,x0,h0,eps);
```

Dabei verwenden wir die kleinste Toleranz $\varepsilon = 10^{-16}$, die bei der GKA *double* von MATLAB noch Sinn macht (Rechnungen mit kleineren Toleranzen finden kein Ende). Dazu nehmen wir noch eine möglichst große und runde Anfangsschrittweite $h_0 = 0.00025$, wo das erste Teilintervall $[t_0, t_0 + 2h_0]$ und weitere solche der Länge $2h_0$ akzeptiert werden.

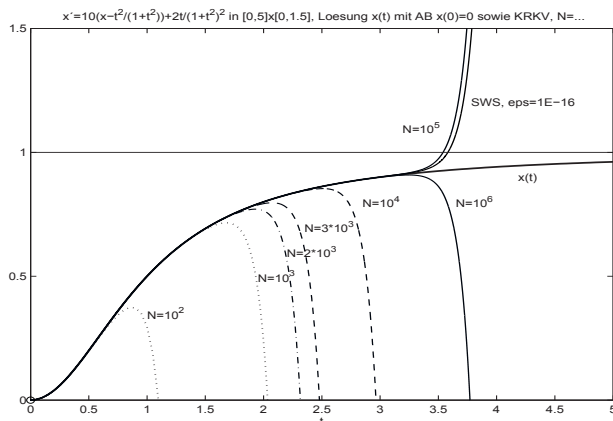


Abb. 5.18

Datei *r_f1e.ps*,
 gDGI $x' = f(t, x)$, $t \in [0, 5]$,
 und Lösung $x(t)$
 mit den AB $x(0) = 0$,
 sowie Näherungslösungen
 mit KRKV bei $N = 10^2, 10^3$,
 $N = 2 \cdot 10^3, 3 \cdot 10^3, 10^4, 10^5, 10^6$,
 dazu das KRKV mit SWS
 bei $h_0 = 0.00025$, $\varepsilon = 10^{-16}$

Wir wollen das Ergebnis erläutern.

Auch das KRKV mit SWS wird meistens etwas "später" die Umgebung der exakten Trajektorie verlassen und dann sich schnell von ihr entfernen.

Mit der Anfangsschrittweite $h_0 = 0.00025$ erzielen wir das beste Ergebnis bzgl. des Aufwands an Funktionsauswertungen. Auch andere h_0 , wie z.B. $h_0 = 0.001$, die nach einigen anfänglichen Halbierungen/Verdopplungen auf die Teilintervalllänge 0.0005 kommen, zeigen ähnliches Verhalten. Mit der Anfangsschrittweite $h_0 = 0.001$ haben wir z.B. eine Situation, wo zunächst zwei Schrittweithalbierungen vorgenommen werden müssen, um dann mit der günstigen Schrittweite 0.0005 weiter zu rechnen.

Aber Rechnungen mit $h_0 \approx 0.00025$, also knapp daneben, weichen schon deutlich ab.

In der ersten Hälfte des Bereiches $[t_0, T] = [0, 5]$ kann man bei $h_0 = 0.00025$ fast mit konstanter Schrittweite rechnen. Die meisten Schrittweithalbierungen und damit verbunden die Nichtakzeptanz eines Schrittes sind in der zweiten Hälfte.

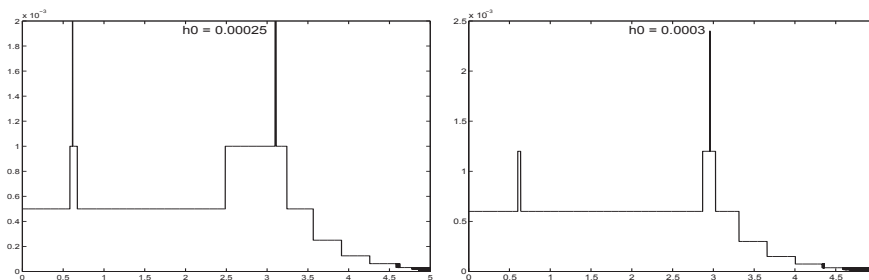
Trotz des strengen Abbruchkriteriums $|est| < 10^{-16}$ wird der Fehler der Näherungslösung dabei ständig größer.

Zum KRKV mit SWS erfolgen noch einige vergleichenden Rechnungen.

h_0	1. Schritt mit $2h_0$	Anzahl Schritte	Anzahl nicht akz.	$2h_{max}$	$2h_{min}$
0.0004, nicht akz.	0.0004	80 809	41 745	0.0016	0.000000195
0.0002, akzeptiert	0.0004	80 809	41 744	0.0016	0.000000195
0.000225, akzeptiert	0.00045	81 639	41 629	0.0018	0.000000219
0.00024, akzeptiert	0.00048	53 010	21 907	0.00192	0.000000937
0.000245, akzeptiert	0.00049	72 417	35 112	0.00196	0.000000239
0.00025 , akzeptiert	0.0005	34 522	9 457	0.0020	0.000001953
0.000255, akzeptiert	0.00051	74 360	36 121	0.00204	0.000000249
0.00026, akzeptiert	0.00052	58 304	25 821	0.00208	0.000000507
0.000275, akzeptiert	0.00055	55 208	23 961	0.0022	0.000000537
0.0003, akzeptiert	0.0006	56 113	24 968	0.0024	0.000000585
0.01, nicht akz.	0.0003125	71 333	35 092	0.0025	0.000000305
0.001, nicht akz.	0.0005	34 522	9 459	0.0020	0.000001953

Tab. 5.16 Ergebnisgrößen zum KRKV mit SWS zu verschiedenen h_0

Wir illustrieren die akzeptierten Schrittweiten $2h_i$ versus t_i . Dabei zeigt sich der vergleichsweise günstige Verlauf, wenn $h_0 = 0.00025$ ist. Die hohen Schrittzahlen der Rechnungen entstehen jedoch am Ende des rechten Teils des Intervalls $[t_0, T]$.



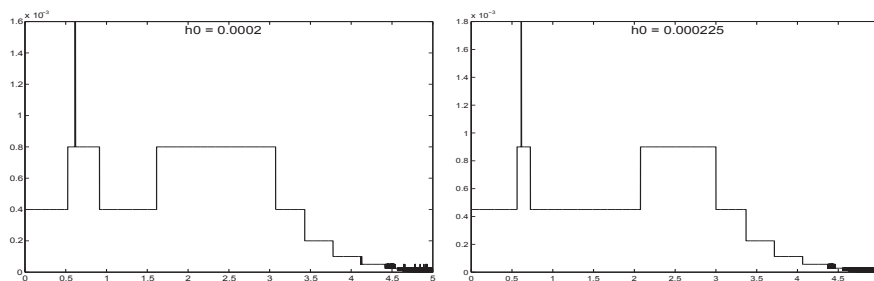


Abb. 5.19 Dateien *swh5.ps*, *swh3.ps*, *swh2.ps*, *swh6.ps* (v.l.o. n.r.u.),
Schrittweitenverhalten $2h_i(t_i)$ beim KRKV mit SWS bei $\varepsilon = 10^{-16}$ und
 $h_0 = 0.000\ 25, 0.000\ 3, 0.000\ 2, 0.000\ 225$

Wir betrachten an ausgewählten Stellen den Fehlerverlauf des KRKV mit SWS bei $h_0 = 0.000\ 25$ und $\varepsilon = 10^{-16}$. Weil an denselben Punkten auch das KRKV bei konstanter Schrittweite mit $N = 10^6$ und $h = 5 \cdot 10^{-6}$ gerechnet wurde, können wir einen Vergleich anstellen (MATLAB mit Nummerierung der Feldgrenzen von 1).

i	t(i)	exakt x(t(i))	KRKV mit SWS eta(i) eta(i)-x(t(i))	KRKV ohne SWS eta(i) eta(i)-x(t(i))
1	0	0	0	0
1000	0.499500	1.996800e-001	1.996800e-001	1.804e-015
2000	1.045500	5.222329e-001	5.222329e-001	6.928e-013
3000	1.545500	7.048903e-001	7.048903e-001	1.038e-010
4000	2.045500	8.071014e-001	8.071014e-001	1.540e-008
5000	2.604000	8.714788e-001	8.714829e-001	4.104e-006
6000	3.427000	9.215337e-001	9.369295e-001	1.540e-002
7000	3.746500	9.334941e-001	1.309307e+000	3.758e-001
8000	3.954250	9.398897e-001	3.940565e+000	3.001e+000
9000	4.079250	9.433116e-001	1.141670e+001	1.047e+001
10000	4.204250	9.464545e-001	3.750217e+001	3.656e+001
11000	4.293937	9.485541e-001	9.058056e+001	8.963e+001
12000	4.356437	9.499463e-001	1.684046e+002	1.675e+002
13000	4.418938	9.512837e-001	3.137977e+002	3.128e+002
14000	4.481438	9.525691e-001	5.854266e+002	5.845e+002
15000	4.543938	9.538050e-001	1.092895e+003	1.092e+003
16000	4.602125	9.549134e-001	1.954864e+003	1.954e+003
17000	4.635563	9.555327e-001	2.730698e+003	2.730e+003
18000	4.666813	9.561002e-001	3.732071e+003	3.731e+003
19000	4.698063	9.566570e-001	5.100786e+003	5.100e+003
20000	4.729313	9.572035e-001	6.971598e+003	6.971e+003
21000	4.760563	9.577399e-001	9.528694e+003	9.528e+003
22000	4.791625	9.582632e-001	1.299943e+004	1.300e+004
23000	4.821750	9.587617e-001	1.756901e+004	1.757e+004
24000	4.847531	9.591813e-001	2.273571e+004	2.273e+004
25000	4.869555	9.595348e-001	2.833684e+004	2.834e+004
26000	4.889078	9.598443e-001	3.444594e+004	3.444e+004
27000	4.906031	9.601103e-001	4.080963e+004	4.081e+004
28000	4.921703	9.603539e-001	4.773350e+004	4.773e+004
29000	4.937484	9.605970e-001	5.589322e+004	5.589e+004
30000	4.952406	9.608248e-001	6.488779e+004	6.489e+004
31000	4.964438	9.610070e-001	7.318352e+004	7.318e+004
32000	4.974961	9.611654e-001	8.130466e+004	8.130e+004
33000	4.984898	9.613141e-001	8.979930e+004	8.980e+004
34000	4.994336	9.614545e-001	9.868681e+004	9.869e+004
34522	5.000000	9.615385e-001	1.044378e+005	1.044e+005

Die absoluten Fehler $|\eta_i - x(t_i)|$ sind beim KRKV mit SWS etwa nur halb so groß.

Mit der in diesem Algorithmus eingebauten Korrektur der Lösung aus Grob- und Feinrechnung möchte man schon eine höhere Genauigkeit erwarten.

Der Aufwand an Funktionsauswertungen $f(t, x)$ über das ganze Intervall ist bzgl. der Anzahl beim KRKV ohne SWS $4 \cdot 10^6$, bei KRKV mit SWS bei 35422 Schritten und 9457 nicht akzeptierten Schritten $12 \cdot 34522 + 8 \cdot 9457 \approx 4.9 \cdot 10^5$, also rund acht Mal weniger. Rechnet man aber z.B. nur bis zur Stelle $t(5000) = 2.604$, wo beide Verfahren noch die Lösungsgenauigkeit $\mathcal{O}(10^{-6})$ aufweisen, dann haben bis dahin das KRKV ohne SWS ca. $2.1 \cdot 10^6$ Funktionswerte berechnet, das KRKV mit SWS braucht bei wenigen nicht akzeptierten Schritten nur $(4 + 8) \cdot 5000 = 6 \cdot 10^4$ Funktionswerte. Das ist ein gewaltiger Unterschied.

Für beide Varianten beginnt nach der Mitte des Lösungsintervalls die rapide Fehlerzunahme.

Bei $t(5000) = 2.604$ hat man für das KRKV mit SWS den Näherungswert $eta(5000) = 0.87148287902258$, den Schrittweitevorschlag $2h = 0.001$ sowie zum Vergleich den exakten Lösungswert $x(t(5000)) = 0.87147877549090$.

Nur wenn man mit dem exakten Wert, also mit dem Funktionsaufruf

```
[ts, xs, hs, anz] = krkv1_sw('f1', 2.604, 5, 0.87147877549090, 0.0005, 10^(-16))
```

weiter rechnet, kommt man in der zweiten Intervallhälfte auf ein annehmbares Ergebnis mit nur einer Nichtakzeptanz der Schrittweite und damit Schrittweithalbung nahe der rechten Intervallgrenze 5, sowie

```
max(size(ts)) = 2362,
max(hs) = 0.004,
min(hs) = 0.001,
max(abs(xs-xexakt)) = abs(xs(2362)-xexakt(5)) = 1.522e-04.
```

Wir haben bei unseren Rechnungen mit expliziten ESV die Erkenntnis gewonnen, dass die näherungsweise Integration an Trajektorien, die instabile Gleichgewichtslagen sind, nur in sehr beschränkten Intervallen und bei hohen Genauigkeitsanforderungen möglich ist.

(d) Gegeben sei zur Riccati-DGI das AWP

$$x' = t - x^2, \quad t \in [t_0, T] = [0, 15], \quad x(t_0) = x_0 = 0. \quad (5.128)$$

Die gDGI wurde schon in Abschnitt 1.8.1 genannt, sowie im Abschnitt 4.2 Bsp. 4.2 (d) mittels des PZV gelöst. Die exakte Lösung des AWP ist

$$x(t) = \frac{\sqrt{3} \operatorname{AiryAi}(1, t) + \operatorname{AiryBi}(1, t)}{\sqrt{3} \operatorname{AiryAi}(t) + \operatorname{AiryBi}(t)}. \quad (5.129)$$

Wir zeigen das Richtungsfeld zur gDGL zusammen mit der Lösungstrajektorie in einem ausgewählten Bereich.

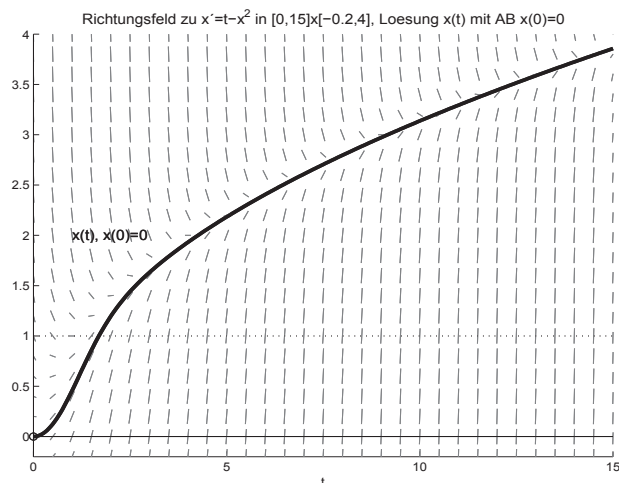


Abb. 5.20

Datei *r_f14b.ps*,
 Richtungsfeld zur gDGL
 $x' = t - x^2$
 in $[0, 15] \times [-0.2, 4]$,
 Lösung $x(t)$ zur AB $x(0) = 0$

Man bemerkt, dass die Lösungskurve $x(t)$ mit der AB $x(0) = 0$ für $t \rightarrow \infty$ stetig wächst. Aber sie ist eine stabile Kurve (stabile Gleichgewichtslage), weil die Linienelemente auf die Kurve zulaufen.

Andererseits wird die numerische Lösung mittels ESV eine Schrittweisenbeschränkung nach sich ziehen. Die partielle Ableitung der rechten Seite der gDGL ist $f_x = -2x$ und sie wird mit wachsendem x betragsmäßig immer größer. In Anlehnung an die Bem. 4.3 haben wir die Beziehung für die Schrittweite $h \leq 1/\max |f_x|$, die somit im Laufe der Integration eher langsam kleiner werden muss, im günstigsten Fall konstant bleiben wird. Wir vergleichen mehrere ESV $\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h)$, $i = 0, 1, \dots, N-1$, bei Wahl einer konstanten Schrittweite $h = (T - t_0)/N = 15/N$, $N = 30$.

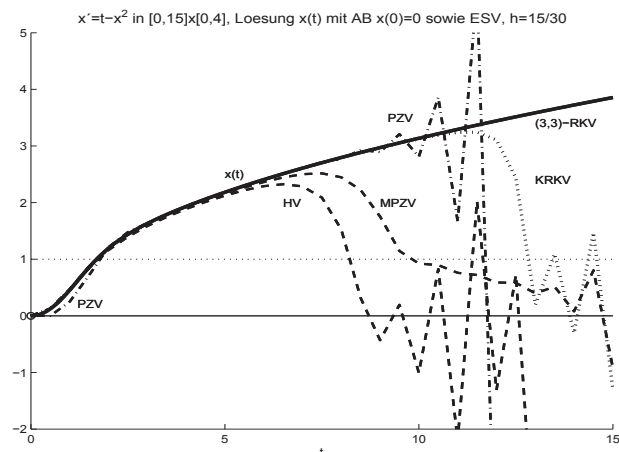


Abb. 5.21

Datei *r_f14c.ps*,
 gDGL $x' = t - x^2$, $t \in [0, 15]$,
 und Lösung $x(t)$
 zur AB $x(0) = 0$,
 sowie Näherungslösungen
 mittels ESV bei $N = 30$:
 PZV, MPZV, HV, KRKV,
 (3,3)-RKV

Keines der expliziten ESV kann über ein längeres Intervall an der exakten Lösung “dran bleiben“. Die beste Approximation liefert noch das (3,3)-RKV.

In der Tat muss die Schrittweite kleiner gewählt werden.

Würden wir die Schrittkenzahl κ (5.102) zum KRKV auswerten, so kann man langsam wachsende Werte erkennen, sodass die im Abschnitt 5.4.3 beschriebene Methode zur Schrittweitenkontrolle angewendet werden kann.

Wir vergleichen noch Rechnungen mit dem KRKV bei verschiedenen Schrittweiten $h = 15/N$.

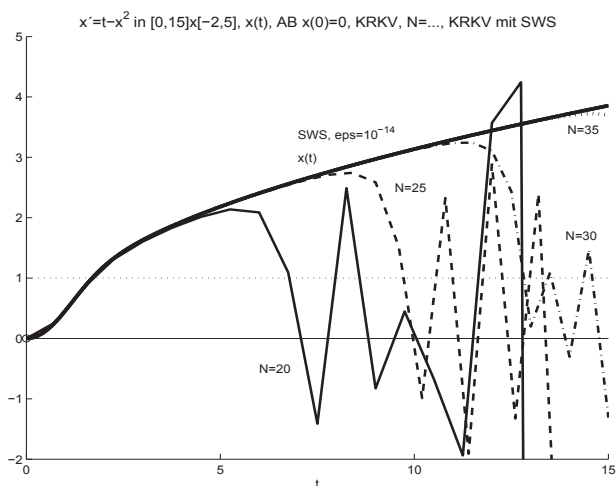


Abb. 5.22

Datei *r_f14e.ps*,
gDGI $x' = t - x^2$, $t \in [0, 15]$,
und Lösung $x(t)$
zur AB $x(0) = 0$,
sowie Näherungslösungen
mit KRKV bei
 $N = 20, 25, 30, 35$,
bzw. KRKV mit SWS bei
 $h_0 = 0.002$ und $\varepsilon = 10^{-14}$

Natürlich kann man die SWS mit Grob-/Feinrechnung anwenden.

Der Funktionsaufruf mit dem KRKV ist

```
h0 = 0.002
eps = 10^(-14)
[t,x,h,anz] = krkv1_sw('f14',t0,T,x0,h0,eps);
```

Dabei verwenden wir eine kleine Toleranz $\varepsilon = 10^{-14}$, die bei der GKA *double* von MATLAB noch Sinn macht. Dazu nehmen wir noch eine möglichst große und runde Anfangsschrittweite $h_0 = 0.002$, wo das erste Teilintervall $[t_0, t_0 + 2h_0]$ und weitere der Länge $2h_0$ akzeptiert werden.

Die sehr genaue Näherungslösung mittels KRKV mit SWS (die Genauigkeit beträgt $\mathcal{O}(10^{-14})$) behält die Schrittweite $2h = 0.004$ fast konstant über das ganze Integrationsintervall $[0, 15]$, macht keine nicht akzeptierten Schritte und braucht somit eine hohe Anzahl von Schritten ($\#Schritte=3752$). Das “Verlassen“ der Umgebung der Lösungstrajektorie wurde nicht bemerkt.

Vergleichen wir das KRKV ohne/mit SWS, wo beide ungefähr gleich genau sein sollen. Für den maximalen Fehler $\approx 2 \cdot 10^{-5}$ brauchen beide Versionen ca. 400 Auswertungen der rechten Seite.

```

t0 = 0; x0=0; T = 15;
N = 100;
[tt2,xx2] = krkv1('f14',t0,T,x0,N);
xxx = f14exv(tt2);
err1 = max(abs(xxx-xx2))

h0 = 0.2;
[ts,xs,hs,anz] = krkv1_sw('f14',t0,T,x0,h0,10^(-4));
nicht_akz = anz
anz_schritte = length(ts)
xxx = f14exv(ts);
err2 = max(abs(xxx-xs))
%-----
err1 = 1.331942702043065e-005
nicht_akz = 1
anz_schritte = 39
err2 = 1.941902481883417e-005

```

Rechnet man wie zuletzt das KRKV mit SWS bei gleichen h_0 und ε über ein Intervall $[0, mT]$, $m > 1$, so erhöht sich proportional die Anzahl der Schritte, die Schrittweite bleibt aber dieselbe.

Zum Schluss testen wir das KRKV mit SWS bei $h_0 = 0.2$ in $[0, 50]$, wobei $\varepsilon = 10^{-k}$, $k = 4, 5, 6, 7, 8, 9$, ist. Wir zeigen die Schrittweite h als Funktion des Integrationssteps (ihre Anzahl ist die Abszissenlänge in der Abb. 5.23). Die Anzahl der nicht akzeptierten Schritte ist sehr gering. Die Genauigkeit ist i. Allg. von der Ordnung $\mathcal{O}(10^{-k-1})$.

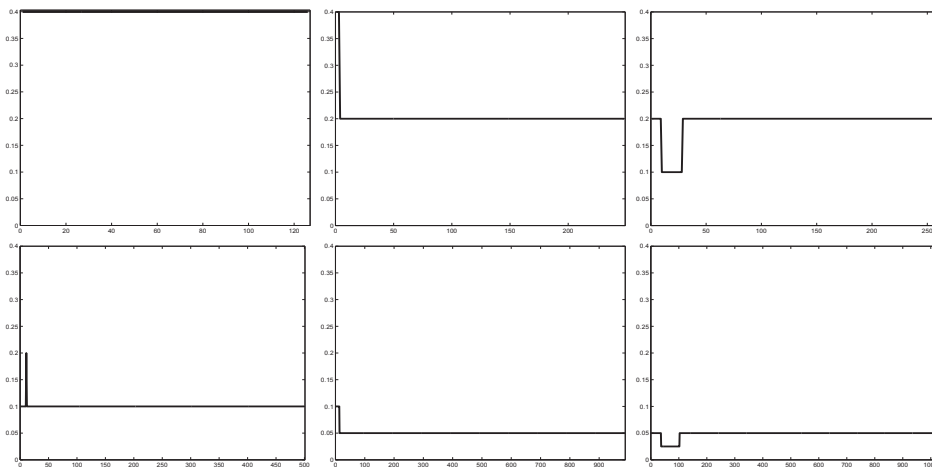


Abb. 5.23 Dateien *r_f14f1.6.ps*, AWP (5.128),
 Schrittweite h als Funktion des Integrationssteps i
 bei Rechnungen mit dem KRKV mit SWS,
 $[t_0, T] = [0, 50]$, $h_0 = 0.2$, $\varepsilon = 10^{-k}$, $k = 4, 5, 6, 7, 8, 9$ (v.l.o.n.r.u.)

5.4.5 Variable Schrittweitensteuerung

Im Abschnitt 5.4.2 wurde die asymptotische Fehlerschätzung bei der Kontrolle der Schrittweite in ESV eingeführt. Dazu ist der asymptotische Fehlerschätzer $R(h)$ (5.101) der Ordnung $p+1$ für den lokalen Diskretisierungsfehler τ_{n+1} des eingebetteten RKV hergeleitet worden.

Im Abschnitt 8.5.2 ist die SWS zu den eingebetteten ROW-Verfahren der Ordnung $p(p+1)$ beschrieben. Dort wird aber in der Formel des Fehlerschätzers nicht die Schrittweite h verwendet. Jetzt soll $R(h)$ bei einem SysgDGL 1. Ordnung mit n Gleichungen genutzt werden, um die Integrationsschrittweite h so zu wählen, dass eine vorgegebene Toleranz $\varepsilon > 0$ in jedem Schritt eingehalten wird. Aus dem Vergleich des maximalen absolut-relativen Fehlers (Fehlerschätzer)

$$err = \max_{k=1(1)n} \left(\frac{|\hat{\eta}_{k,i+1} - \eta_{k,i+1}|/h}{1 + |\hat{\eta}_{k,i+1}|} \right) \quad (5.130)$$

mit der Toleranz ε bestimmt man den Vorschlag für die neue Schrittweite

$$h_{neu} = \delta \rho h, \quad \rho = \sqrt[q]{\frac{\varepsilon}{err}}. \quad (5.131)$$

Dabei bezeichnet $\delta \in (0, 1]$ einen geeigneten Sicherheitsfaktor. Mit δ nahe der Eins wird ein häufiges Rechnen an der Schrittweitengrenze vermieden. Die Größe $\delta \rho$ bezeichnet man als Schrittweitenveränderungsfaktor. Für das "bessere" Verfahren ist in (5.131) offenbar $q = p+1$ zu setzen.

Die Vorgehensweise ist grob folgendermaßen. Ergibt sich $err > \varepsilon$, so muss der letzte Schritt mit der neuen Schrittweite h_{neu} gegebenenfalls mehrfach wiederholt werden. Andernfalls wird der Schritt akzeptiert und mit dem Vorschlag weitergerechnet. Dabei sind kleine Modifikationen durchaus noch möglich.

Auf diese Weise entstehen adaptive RKV mit automatischer SWS bei variabler oder freier Schrittweitenwahl. Sie erlauben eine weitgehende Anpassung der Integrationsschrittweite h an den Lösungsverlauf und lassen stärkere Vergrößerungen von h zu. Bei Verkleinerung von h erweisen sich Halbierungen oft als recht günstig.

Damit ergeben sich folgende drei Varianten der SWS.

(1) Variante 1

Falls $err \geq \varepsilon$ ist, dann einfach $h := \frac{h}{2}$ und Wiederholung des Schrittes mit h .

Falls $err < \varepsilon$ ist, dann Akzeptierung des Schrittes und Vorschlag für die neue Schrittweite $h := \delta \rho h$, $\delta \in (0, 1]$.

Der Sicherheitsfaktor δ ist oft nahe der Eins.

(2) Variante 2

Analog zur Variante 1, nur dass im ersten Fall

$$h := \max\left(\frac{h}{2}, \delta \rho h\right). \quad (5.132)$$

Man "fällt" also dabei nicht unter eine Schrittweithalbierung.

(3) Variante 3

Man initialisiert den Sicherheitsfaktor δ , z.B. $\delta = 0.5$, und berechnet

$$h_{neu} = \delta \rho h.$$

Falls $h_{neu} < h$ (d.h. $err \geq \varepsilon$) ist, wird $h := h_{neu}$ gesetzt und der aktuelle Schritt mit h wiederholt.

Falls $h_{neu} \geq h$ ist, dann Akzeptierung des Schrittes und Vorschlag für die neue Schrittweite in zwei Versionen.

- (a) Ist $err < \delta_1 \varepsilon$, $\delta_1 \approx 0.1$, dann neue Schrittweite $h := h_{neu}$.
- (b) Ist $err \geq \delta_1 \varepsilon$, dann Beibehaltung der Schrittweite h .

Zum SysgDGL mit $n = 2$ Gleichungen als AWP

$$\begin{aligned} x'_1 &= f_1(t, x_1, x_2) = t - x_2, & x_1(0) &= 1, & t \in [t_0, t_1] &= [0, 1], \\ x'_2 &= f_2(t, x_1, x_2) = -t + x_1 x_2, & x_2(0) &= 1, \end{aligned}$$

wird die Implementation zur Variante 1 als Prozeduren in der Programmiersprache Dev-Pascal [25] mit dem eingebetteten RKF2(3)₄ notiert.

0				
$\frac{1}{4}$		$\frac{1}{4}$		
$\frac{27}{40}$	$-\frac{189}{800}$	$\frac{729}{800}$		
1	$\frac{214}{800}$	$\frac{1}{33}$	$\frac{650}{891}$	
	$\frac{214}{891}$	$\frac{1}{33}$	$\frac{650}{891}$	
	$\frac{533}{2106}$	0	$\frac{800}{1053}$	$-\frac{1}{78}$

```
{ Dev-Pascal }
type vektor=array[1..nmax] of float;
procedure vf(n:integer; t:float; var x,xs:vektor);
begin
  xs[1]:= t-x[2];
  xs[2]:=-t+x[1]*x[2];
end;

{spezielle Version: 1 Schritt}
procedure rk23step(n:integer; t,h,eps:float;
                  var tneu,err,rho:float;
                  var x,xneu:vektor);
{ vf ist global }

var z,k1,k2,k3,k4:vektor;
    i:integer;
    q,errh:float;
begin
  q:=3;
  vf(n,t,x,k1);
  for i:=1 to n do z[i]:=x[i]+h*k1[i]/4;
```

```

vf(n,t+h/4,z,k2);
for i:=1 to n do z[i]:=x[i]+h*(-189*k1[i]+729*k2[i])/800;
vf(n,t+27*h/40,z,k3);
for i:=1 to n do z[i]:=x[i]+h*(214*k1[i]/891+k2[i]/33+650*k3[i]/891);
vf(n,t+h,z,k4);
err:=0;
for i:=1 to n do
  begin
    xneu[i]:=x[i]+h*(533*k1[i]/2106+800*k3[i]/1053-k4[i]/78);
    z[i] :=x[i]+h*(214*k1[i]/891+k[2]/33+650*k3[i]/891);
    errh :=abs(xneu[i]-z[i])/h/(1+abs(xneu[i]));
    if errh>err then err:=errh;
  end;
tneu:=t+h;
if err=0 then rho:=1 else rho:=exp(ln(eps/err)/q);
end;

```

Prozedur rk23 mit obiger Schrittprozedur

Dabei erfolgt eine Reihe von Zwischenausgaben und Abfragen.

```

procedure rk23(n:integer; eps,hstart,t0,t1:float; var x0:vektor);
{ rk23step global }

var x,xneu:vektor;
    i,anz:integer;
    t,h,err,maxerr,tneu,rho,delta:float;

begin
  delta:=0.9;
  t:=t0; x:=x0;
  h:=hstart; rho:=1;
  err:=0; maxerr:=0; anz:=0;
  writeln;
  writeln('t':10,'x':30,'err':20);
  while t<=t1-1E-15 do
    begin
      writeln;
      write(t,' ');
      for i:=1 to n do write(x[i],' ');
      writeln(err);
      h:=2*h;
      {hier Zaehler anf fuer anfaengliche Halbierungen}
      repeat
        h:=0.5*h;
        rk23step(n,t,h,eps,tneu,err,rho,x,xneu);
      until err<eps;
      if err>maxerr then maxerr:=err;
      t:=tneu;
      x:=xneu;
      anz:=anz+1;
      h:=delta*rho*h;
      if t+h>t1 then h:=t1-t;
    end;
  writeln;
  writeln('maxerr=',maxerr);
end;

```

Wir betrachten nun das vierstufige eingebettete RKF2(3)₄ für die Riccati-DGI

$$x' = t - x^2, \quad x(0) = 0, \quad t \in [0, 1],$$

(vergl. Abschnitte 1.8.1 sowie 2.2.3 mit Bsp. 2.4 (20)).

Die exakte Lösung ist

$$x(t) = \frac{\sqrt{3} \operatorname{AiryAi}(1, t) + \operatorname{AiryBi}(1, t)}{\sqrt{3} \operatorname{AiryAi}(t) + \operatorname{AiryBi}(t)}$$

und $x(1) = 0.455\,544\,526\,081\,8736$.

Die SWS erfolgt mittels der Variante 1.

Die Parameter sind $\delta = 0.9$, $q = 3$ bzw. $q = 2$, die Toleranzen $\varepsilon = 10^{-m}$, $m = 1(1)10$, die Startschrittweite $h = 1$, welche bei kleinen Toleranzen durch das Verfahren automatisch "heruntergeregelt" wird.

Als Ergebnisse folgen

- der Näherungswert η_n bei $t_n = 1$ zu $x(1)$,
- **anz** als Anzahl der akzeptierten Schritte, damit ist die Zahl der Funktionswertberechnungen $\geq 4 \cdot \text{anz}$,
- **maxest** als geschätzter maximaler Fehler aller akzeptierten Schritte,
- **anf** als Anzahl der anfänglichen Halbierungen der Schrittweite,
- **hm** als mittlere Schrittweite nach Einpegeln des Verfahrens,
- $e_n = x(1) - \eta_n$ als Fehler bei $t_n = 1$.

ε	η_n	anz	maxerr	anf	hm	e_n
10^{-1}	0.463 309 023 23	1	7.99e-04	0	1	-7.76e-03
10^{-2}	0.463 309 023 23	1	7.99e-04	0	1	-7.76e-03
10^{-3}	0.463 309 023 23	1	7.99e-04	0	1	-7.76e-03
10^{-4}	0.455 573 339 20	3	8.77e-05	1	0.45	-2.88e-05
10^{-5}	0.455 550 665 67	6	8.51e-06	3	0.18	-6.14e-06
10^{-6}	0.455 545 204 60	15	9.73e-07	4	0.06	-6.78e-07
10^{-7}	0.455 544 610 96	41	9.08e-08	5	0.02	-8.49e-08
10^{-8}	0.455 544 532 99	121	9.24e-09	6	0.004	-6.91e-09
10^{-9}	0.455 544 526 54	372	9.43e-10	7	0.002	-4.60e-10
10^{-10}	0.455 544 526 12	1165	9.75e-11	7	0.0005	-4e-11

0.455 544 526 08 = $x(1)$

Tab. 5.17 Ergebnisse mit $q = p + 1 = 3$ für das RKF2(3)₄ mit SWS Variante 1 zur Riccati-DGI $x' = t - x^2$, $x(0) = 0$, $t \in [0, 1]$

Es gibt nur geringe Unterschiede, wenn in der SWS die kleinere Genauigkeitsordnung $q = p = 2$ genommen wird.

ε	η_n	anz	maxerr	anf	hm	e_n
10^{-1}	0.463 309 023 23	1	7.99e-04	0	1	-7.76e-03
10^{-2}	0.463 309 023 23	1	7.99e-04	0	1	-7.76e-03
10^{-3}	0.463 309 023 23	1	7.99e-04	0	1	-7.76e-03
10^{-4}	0.455 519 827 78	3	8.77e-05	1	0.45	-2.47e-05
10^{-5}	0.455 529 632 58	7	9.41e-06	3	0.18	1.49e-05
10^{-6}	0.455 544 396 92	18	9.63e-07	4	0.06	1.29e-07
10^{-7}	0.455 544 565 06	45	9.90e-08	5	0.02	-3.90e-08
10^{-8}	0.455 544 530 61	123	9.91e-09	6	0.004	-4.53e-09
10^{-9}	0.455 544 526 36	362	9.95e-10	7	0.002	-2.81e-10
10^{-10}	0.455 544 526 11	1114	9.97e-11	7	0.0005	-3.24e-11

0.455 544 526 08 = $x(1)$

Tab. 5.18 Ergebnisse mit $q = p = 2$ für das RKF2(3)₄ mit SWS Variante 1 zur Riccati-DGI $x' = t - x^2$, $x(0) = 0$, $t \in [0, 1]$

Wir testen auch das eingebettete RKV von Verner5(6)₈ für die Riccati-DGI. Aus der Anzahl **anz** der akzeptierten Schritte ergibt sich die Zahl der Funktionswertberechnungen $\geq 8 \cdot \mathbf{anz}$. Die mittlere Schrittweite **hm** nach Einpegeln des Verfahrens ist auch ungefähr die Schrittweite am Ende des Integrationsintervalls.

ε	η_n	anz	maxerr	hm	e_n
10^{-1}	0.454 628 776 491	1	1.56e-03	1	9.16e-04
10^{-2}	0.454 628 776 491	1	1.56e-03	1	9.16e-04
10^{-3}	0.455 537 055 604	2	7.45e-05	0.5	7.47e-06
10^{-4}	0.455 537 055 604	2	7.45e-05	0.5	7.47e-06
10^{-5}	0.455 544 266 914	4	8.94e-06	0.3	2.59e-07
10^{-6}	0.455 544 509 652	5	8.49e-07	0.2	1.64e-08
10^{-7}	0.455 544 523 299	7	8.50e-08	0.15	2.79e-09
10^{-8}	0.455 544 526 016	12	8.31e-09	0.10	5.97e-11
10^{-9}	0.455 544 526 069	17	9.72e-10	0.06	5.98e-12
10^{-10}	0.455 544 526 081	28	8.92e-11	0.04	-6.01e-12

0.455 544 526 0818 = $x(1)$

Tab. 5.19 Ergebnisse mit $q = p + 1 = 6$ für das Verner5(6)₈ mit SWS Variante 1 zur Riccati-DGI $x' = t - x^2$, $x(0) = 0$, $t \in [0, 1]$

Interessant ist auch ein Vergleich mehrerer eingebetteter RKVp(p+1) mit $q = p + 1$ bei gleicher Toleranz $\varepsilon = 10^{-6}$ und Startschrittweite $h = 1$.

Dazu werden die Verfahren von RKF2(3)₃, RKF2(3)₄, Sarafyan4(5)₆, Fehlberg5(6)₈, Fehlberg5(6)₈ modifiziert, Verner5(6)₈ und Prince-Dormand5(6)₈ herangezogen.

Verfahren	η_n	anz	maxerr	hm	e_n
RKF2(3) ₃	0.455 545 008 763	84	9.19e-07	0.004	-4.83e-07
RKF2(3) ₄	0.455 545 204 600	15	9.73e-07	0.06	-6.78e-07
Sarafyan4(5) ₆	0.455 544 453 019	8	5.48e-07	0.18	7.31e-08
Fehlberg5(6) ₈	0.455 443 892 335	3	5.11e-07	0.25	1.37e-07
Fehlberg5(6) ₈ mod.	0.455 543 963 672	5	8.33e-07	0.22	5.62e-07
Verner5(6) ₈	0.455 544 509 652	5	8.49e-07	0.20	1.64e-08
Prince-Dormand5(6) ₈	0.455 544 560 985	3	4.70e-07	0.25	-3.49e-08

0.455 544 526 081 = $x(1)$

Tab. 5.20 Resultate mit $q=p+1$ für 7 eingebettete RKV_{p(p+1)} mit SWS Variante 1, $\varepsilon = 10^{-6}$, Startschrittweite $h = 1$, zur Riccati-DGI

Zum Abschluss sollen noch einige Modifikationen des Fehlerschätzers err untersucht werden. Die Formel (5.130) ist natürlich auch dabei. Dabei wollen wir feststellen, welchen Einfluss die Berücksichtigung der Schrittweite h in der Formel err im Schritt $i + 1$ hat.

Es sind folgende vier Versionen. Dabei ist n die Dimension des SysgDGI.

$$err = \left\{ \begin{array}{l} (1) \max_{k=1(1)n} \left(\frac{|\hat{\eta}_{k,i+1} - \eta_{k,i+1}|/h}{1 + |\hat{\eta}_{k,i+1}|} \right) \\ (2) \max_{k=1(1)n} \left(\frac{|\hat{\eta}_{k,i+1} - \eta_{k,i+1}|/h}{\max(1, |\hat{\eta}_{k,i+1}|)} \right) \\ (3) \max_{k=1(1)n} \left(\frac{|\hat{\eta}_{k,i+1} - \eta_{k,i+1}|/h}{\underbrace{\max(1, |\hat{\eta}_{k,i+1}|, s_i)}_{= s_{i+1}, s_0 = 1}} \right) \\ (4) \max_{k=1(1)n} \left(\frac{|\hat{\eta}_{k,i+1} - \eta_{k,i+1}|}{\max(1, |\hat{\eta}_{k,i+1}|, s_i)} \right) \end{array} \right\} = Ch^q + \mathcal{O}(h^{q+1}) \approx \varepsilon. \quad (5.133)$$

Bei einer gDGI 1. Ordnung wie für das Testbeispiel der Riccati-DGI ist $n = 1$. In den Versionen (3) und (4) werden im jeweiligen Schritt im Nenner der vorhergehende Nennerwert mit einbezogen.

Als eingebettetes RKV mit diesen vier Fehlerschätzern $err_{(m)}$, $m = 1, 2, 3, 4$, wählen wir das RKF7(8)₁₃, sodass $q = 7$ bzw. 8 sein kann.

Im Schrittweitemvorschlag

$$h_{neu} = \delta \rho h, \quad \rho = \sqrt[q]{\frac{\varepsilon}{err}},$$

wird mit $\delta = 0.9$ gerechnet.

Man beachte, dass im Fall $q \gg 1$ und $\frac{\varepsilon}{err} > 1$ die q -te Wurzel nahe der Eins sein kann und dann $h_{neu} \lesssim h$ wird, also eine deutliche Schrittweitenvergrößerung wirklich erst für $err \ll \varepsilon$ eintritt.

Nun zur Bewertung und zum Vergleich der Versionen (1) – (4) nach mehreren Testrechnungen mit dem eingebetteten Verfahren RKF7(8)₁₃.

```

{ Dev-Pascal }
{ vf ist global }
{ Version RKF7(8)_13, 1 Schritt }
procedure rkf7813step(variante,n:integer; t,h,eps:float;
                      var tneu,err,rho:float;
                      var x,xneu,s:vektor);

var z,k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12,k13,diffxz:vektor;
    i:integer;
    q,erri:float;

begin
  {step}
  vf(n,t,x,k1);
  for i:=1 to n do z[i]:=x[i]+h*2/27*k1[i];
  vf(n,t+h*2/27,z,k2);
  for i:=1 to n do z[i]:=x[i]+h*(1/36*k1[i]+1/12*k2[i]);
  vf(n,t+h/9,z,k3);
  for i:=1 to n do z[i]:=x[i]+h*(1/24*k1[i]+1/8*k3[i]);
  vf(n,t+h/6,z,k4);
  for i:=1 to n do z[i]:=x[i]+h*(5/12*k1[i]-25/16*k3[i]+25/16*k4[i]);
  vf(n,t+h*5/12,z,k5);
  for i:=1 to n do z[i]:=x[i]+h*(k1[i]/20+k4[i]/4+k5[i]/5);
  vf(n,t+h/2,z,k6);
  for i:=1 to n do z[i]:=x[i]+h*(-25/108*k1[i]+125/108*k4[i]
                                -65/27*k5[i]+125/54*k6[i]);

  vf(n,t+h*5/6,z,k7);
  for i:=1 to n do z[i]:=x[i]+h*(31/300*k1[i]+61/225*k5[i]
                                -2/9*k6[i]+13/900*k7[i]);

  vf(n,t+h*1/6,z,k8);
  for i:=1 to n do z[i]:=x[i]+h*(2*k1[i]-53/6*k4[i]+704/45*k5[i]
                                -107/9*k6[i]+67/90*k7[i]+3*k8[i]);

  vf(n,t+h*2/3,z,k9);
  for i:=1 to n do z[i]:=x[i]+h*(-91/108*k1[i]+23/108*k4[i]-976/135*k5[i]
                                +311/54*k6[i]-19/60*k7[i]+17/6*k8[i]-1/12*k9[i]);

  vf(n,t+h/3,z,k10);
  for i:=1 to n do z[i]:=x[i]+h*(2383/4100*k1[i]-341/164*k4[i]
                                +4496/1025*k5[i]-301/82*k6[i]+2133/4100*k7[i]
                                +45/82*k8[i]+45/164*k9[i]+18/41*k10[i]);

  vf(n,t+h,z,k11);
  for i:=1 to n do z[i]:=x[i]+h*(3/205*k1[i]-6/41*k6[i]-3/205*k7[i]
                                -3/41*k8[i]+3/41*k9[i]+6/41*k10[i]);

  vf(n,t,z,k12);
  for i:=1 to n do z[i]:=x[i]+h*(-1777/4100*k1[i]-341/164*k4[i]
                                +4496/1025*k5[i]-289/82*k6[i]+2193/4100*k7[i]
                                +51/82*k8[i]+33/164*k9[i]+12/41*k10[i]+k12[i]);

  vf(n,t+h,z,k13);

  for i:=1 to n do
  begin
    xneu[i]:=x[i]+h*(34/105*k6[i]+9/35*k7[i]+9/35*k8[i]+
                    9/280*k9[i]+9/280*k10[i]+41/840*k12[i]+41/840*k13[i]);
    { z[i] :=x[i]+h*(41/840*k1[i]+34/105*k6[i]+9/35*k7[i]+9/35*k8[i]+
                    9/280*k9[i]+9/280*k10[i]+41/840*k11[i]); }
    diffxz[i]:=h*41/840*(-k1[i]-k11[i]+k12[i]+k13[i]);
  end;
end;

```

```

err:=1E-20; {0;}
case variante of
  11:begin
    q:=7;
    for i:=1 to n do
      begin
        {s[i]:=max2(max2(1,abs(xneu[i])),s[i]);}
        {s[i]:=max2(1,abs(xneu[i]));}
        s[i]:=abs(xneu[i])+1;
        erri:=abs(diffxz[i])/h/s[i];
        if erri>err then err:=erri;
      end;
    tneu:=t+h;
    if err=0 then rho:=1 else rho:=exp(ln(eps/err)/q);
  end;
  12:begin
    q:=8;
    for i:=1 to n do
      begin
        {s[i]:=max2(max2(1,abs(xneu[i])),s[i]);}
        {s[i]:=max2(1,abs(xneu[i]));}
        s[i]:=abs(xneu[i])+1;
        erri:=abs(diffxz[i])/h/s[i];
        if erri>err then err:=erri;
      end;
    tneu:=t+h;
    if err=0 then rho:=1 else rho:=exp(ln(eps/err)/q);
  end;
  21:begin
    q:=7;
    for i:=1 to n do
      begin
        s[i]:=max2(max2(1,abs(xneu[i])),s[i]);
        erri:=abs(diffxz[i])/s[i];
        if erri>err then err:=erri;
      end;
    tneu:=t+h;
    if err=0 then rho:=1 else rho:=exp(ln(eps/err)/q);
  end;
  22:begin
    q:=8;
    for i:=1 to n do
      begin
        s[i]:=max2(max2(1,abs(xneu[i])),s[i]);
        erri:=abs(diffxz[i])/s[i];
        if erri>err then err:=erri;
      end;
    tneu:=t+h;
    if err=0 then rho:=1 else rho:=exp(ln(eps/err)/q);
  end;
end;
end;

```

(a) Zwischen den drei Versionen (1) – (3) mit $q = 7, 8$ gibt es keine signifikanten Unterschiede bzgl. Anzahl der Schritte, Schrittweiten und Genauigkeit.

(b) In der Version (4) gibt es zwischen den Fällen $q = 7$ und $q = 8$ keine wesentlichen Unterschiede.

(c) So konzentrieren wir uns auf den Vergleich der Versionen (1) und (4) mit dem Parameter $q = 8$ und einigen Fällen bzgl. der Toleranz ε .

Das Rahmenprogramm in Dev-Pascal enthält die obige Prozedur zum RKF7(8)₁₃. Zu den Fehlerschätzern $err_{(m)}$ im i -ten Schritt wird zusätzlich ihr Maximum über alle Schritte bestimmt.

```

program ERKV2;
{ ERKV2.PAS
  Eingebettete (r+1)-stufige Einschrittverfahren der Ordnung p(p+1)
  vom Runge-Kutta-Fehlberg-Typ mit freier Schrittweitensteuerung
  fuer SysgDGl 1. Ordnung (AWP) x'=f(t,x), x(t0)=x0, t0<=t<=t1,
  aktuell r+1=13, p=7
  Varianten:
    q = 7,8
    RKF7(8)_13
    RKF7(8)_13 mit etwas besserer SWS
}
const nmax=10;
type float=double; vektor=array[1..nmax] of float;
var version,n,i:integer; x0:vektor; t0,t1,eps,hstart:float;

function max2(x1,x2:float):float;
begin
  if x1>x2 then max2:=x1 else max2:=x2;
end;
procedure vf(n:integer; t:float; var x,xs:vektor);
begin
  xs[1]:=t-sqr(x[1]);
end;
procedure xexakt(n:integer; t:float; var xex:vektor);
begin
  xex[1]:=0.4555445260818736;
end;

{ Version RKF7(8)_13, 1 Schritt }
procedure rkf7813step
...

{ Prozedur mit obiger Schrittprozedur }
procedure EIN_RKV(variante,n:integer; eps,hstart,t0,t1:float;
  var x0:vektor);
{ Steps sind global }
var x,xneu,xex,s:vektor;
  i,anz:integer;
  t,h,h_alt,err,maxerr,tneu,rho:float;

begin
  t:=t0; x:=x0;
  h:=hstart;
  rho:=1;
  err:=0; maxerr:=0; anz:=0;
  for i:=1 to n do s[i]:=1;
  h_alt:=0;
  writeln; writeln('t':6,'h':12,'eta':20,'err':17);

  while t<=t1-1e-15 do
    begin
      write(t:11,' ',h_alt:11,' ');
      for i:=1 to n do write(x[i]:23,' ');
      writeln(err:11);

      h:=2*h;

```



```

repeat
  h:=0.5*h;
  case variante of
    11,12,21,22: rkf7813step(variante,n,t,h,eps,tneu,err,rho,x,xneu,s);
  end;
until err<eps;
if err>maxerr then maxerr:=err;
t:=tneu;
x:=xneu;
anz:=anz+1;
h_alt:=h;
h:=0.9*rho*h;
if t+h>t1 then h:=t1-t;
end;
write(t:11,' ',h_alt:11,' ');
for i:=1 to n do write(x[i]:23,' ');
writeln(err:11);

xexakt(n,t1,xex);
write(' ':25);
for i:=1 to n do write(xex[i]:23,' ');
writeln; writeln;
write('Fehler bei t=1: ');
for i:=1 to n do write(xex[i]-x[i],' ');
writeln; writeln('maxerr = ',maxerr);
writeln('Anzahl der Schritte = ',anz);
end;

{Block}
begin
  writeln('Eingebettete (r+1)-stufige ESV der Ordnung p(p+1)');
  writeln('vom Runge-Kutta-Fehlberg-Typ mit variabler SWS');
  writeln('fuer SysgDGl 1. Ordnung (AWP) x''=f(t,x), x(t0)=t0, t0<=t<=t1');
  writeln;
  writeln('Varianten RKFp(p+1)_(r+1)');
  writeln('(11) RKF7(8)_13, q=7');
  writeln('(12) RKF7(8)_13, q=8');
  writeln('(21) RKF7(8)_13, q=7, mit besserer SWS');
  writeln('(22) RKF7(8)_13, q=8, mit besserer SWS');
  writeln;
  write('Auswahl: ');
  readln(variante);

  { Initialisierungen, AB }
  n:=1;
  t0:=0; t1:=1;
  for i:=1 to n do x0[i]:=0;
  hstart:=t1-t0; {=1}
  eps:=1e-13; {eps:=1e-16;}
  writeln('Ergebnisse, ev. mit Zwischenausgaben');
  writeln('Riccati-DGl x''=t-x*x, x(0)=0, 0<=t<=1, eps=1E-16');
  case variante of
    11: writeln('(11) RKF7(8)_13, q=7');
    12: writeln('(12) RKF7(8)_13, q=8');
    21: writeln('(21) RKF7(8)_13, q=7, mit besserer SWS');
    22: writeln('(22) RKF7(8)_13, q=8, mit besserer SWS');
  end;

  EIN_RKV(variante,n,eps,hstart,t0,t1,x0);
  readln;
end.

```

Gemäß der Definition (5.133) sowie auch für das Testbeispiel gilt i. Allg. $err_{(1)} > err_{(4)}$. Wir betrachten nur zwei Fälle zum Verhalten der Schrittweite und Schrittzahl. Folgendes kann man feststellen.

(a) Sei $\varepsilon > err_{(1)} > err_{(4)}$.

Dann gilt $\rho_{(m)} = \sqrt[m]{\frac{\varepsilon}{err_{(m)}}} > 1$ und $1 < \rho_{(1)} < \rho_{(4)}$.

Damit lässt der Faktor $\rho_{(4)}$ manchmal zu große Schrittweitenveränderungen zu als $\rho_{(1)}$ und man muss dann wieder Verkleinerungen der Schrittweite an mehreren Stellen vornehmen. Somit sind insgesamt mehr Schritte zu machen und es folgt $anz_{(4)} \geq anz_{(1)}$.

(b) Sei $\varepsilon < err_{(4)} > err_{(1)}$.

Dann gilt $\rho_{(m)} = \sqrt[m]{\frac{\varepsilon}{err_{(m)}}} < 1$ und $\rho_{(1)} < \rho_{(4)} < 1$.

Damit kann beim Faktor $\rho_{(1)}$ die Schrittweite oft zu klein werden, sodass insgesamt mehr Schritte zu machen sind als bei $\rho_{(4)}$. Es folgt also $anz_{(1)} \geq anz_{(4)}$.

ε	<i>anz</i>	
	<i>err</i> ₍₁₎	<i>err</i> ₍₄₎
10 ⁻¹⁰	6	5
10 ⁻¹¹	8	7
10 ⁻¹²	10	9
10 ⁻¹³	13	12
10 ⁻¹⁴	19	13
10 ⁻¹⁵	23	18
10 ⁻¹⁶	30	23
10 ⁻¹⁷	51	29
10 ⁻¹⁸	75	36

Tab. 5.21

Riccati-DGI $x' = t - x^2$, $x(0) = 0$, $t \in [0, 1]$,
 Ergebnisse zur Schrittzahl für das RKF7(8)₁₃
 mit variabler SWS, Versionen (1) und (4),
 in Abhängigkeit von der Toleranz ε ,
 Startschrittweite $h = 1$

Weitere vergleichende Rechnungen und Ergebnisse für $err_{(1)}$ (12) bzw. $err_{(4)}$ (22)

```

eps=1E-13
(12) RKF7(8)_13, q=8
      t          h          eta          err
0.000E+000  0.000E+000  0.000000000000000E+000  0.000E+000
1.250E-001  1.250E-001  7.810974493529048E-003  1.491E-014
1.964E-001  7.136E-002  1.926322611159285E-002  6.991E-016
2.561E-001  5.971E-002  3.273082416928727E-002  1.495E-016
3.167E-001  6.060E-002  4.997966260287092E-002  4.129E-017
3.899E-001  7.222E-002  7.517516537657150E-002  4.838E-016
5.155E-001  1.266E-001  1.310587783359795E-001  1.571E-014
5.872E-001  7.179E-002  1.690206061373135E-001  1.494E-014
6.692E-001  8.194E-002  2.174350242097448E-001  6.715E-014
7.467E-001  7.751E-002  2.677425036744369E-001  7.653E-014
8.188E-001  7.213E-002  3.180124517968657E-001  6.795E-014
8.870E-001  6.813E-002  3.680975660043853E-001  6.111E-014
9.522E-001  6.521E-002  4.179831582275155E-001  5.677E-014
1.000E+000  4.783E-002  4.555445260818967E-001  7.301E-015
                        x(1)=4.555445260818736E-001
Fehler bei t=1: -2.3037127760971998E-0014
maxerr = 7.652609647636989E-014
Anzahl der Schritte = 13
    
```

```

eps=1E-13
(22) RKF7(8)_13, q=8, mit besserer SWS
  t           h           eta           err
0.000E+000  0.000E+000  0.000000000000000E+000  0.000E+000
1.250E-001  1.250E-001  7.810974493529048E-003  1.879E-015
2.174E-001  9.245E-002  2.361734716487190E-002  6.653E-016
2.953E-001  7.784E-002  4.348678139039374E-002  1.642E-016
3.734E-001  7.808E-002  6.934160811555169E-002  5.373E-017
4.634E-001  9.004E-002  1.063200407856630E-001  7.597E-016
5.380E-001  7.457E-002  1.425040021936496E-001  9.929E-016
6.574E-001  1.195E-001  2.101850169712884E-001  9.342E-014
7.117E-001  5.422E-002  2.444952028586285E-001  3.716E-016
8.099E-001  9.820E-002  3.115982299868878E-001  6.977E-014
9.023E-001  9.245E-002  3.796810471907427E-001  7.018E-014
9.893E-001  8.697E-002  4.470559901787230E-001  6.190E-014
1.000E+000  1.073E-002  4.555445260819030E-001  5.816E-020
x(1)=4.555445260818736E-001

```

```

Fehler bei t=1: -2.9420910152566648E-0014
maxerr = 9.341608054920965E-014
Anzahl der Schritte = 12
-----

```

```

eps=1E-16
(12) RKF7(8)_13, q=8
  t           h           eta           err
0.000E+000  0.000E+000  0.000000000000000E+000  0.000E+000
6.250E-002  6.250E-002  1.953077317739325E-003  1.454E-017
9.829E-002  3.579E-002  4.830237029777594E-003  1.000E-020
1.492E-001  5.093E-002  1.113047478020010E-002  2.747E-017
2.031E-001  5.388E-002  2.060791454646464E-002  6.106E-017
2.547E-001  5.157E-002  3.237606517279502E-002  2.625E-017
3.095E-001  5.486E-002  4.776517104615795E-002  1.552E-017
3.407E-001  3.116E-002  5.781018348444526E-002  7.684E-018
3.794E-001  3.865E-002  7.156476294992341E-002  4.046E-017
4.183E-001  3.895E-002  8.685663669078768E-002  6.482E-017
4.553E-001  3.701E-002  1.026914562135447E-001  6.880E-017
4.902E-001  3.490E-002  1.187654580846073E-001  6.297E-017
5.235E-001  3.328E-002  1.350997533685981E-001  6.206E-017
5.553E-001  3.180E-002  1.515974604979440E-001  5.647E-017
5.860E-001  3.074E-002  1.683508991239710E-001  5.102E-017
6.161E-001  3.009E-002  1.854955361799221E-001  5.485E-017
6.453E-001  2.919E-002  2.028071993774763E-001  5.406E-017
6.737E-001  2.837E-002  2.202493159303462E-001  5.329E-017
7.013E-001  2.763E-002  2.377927315196081E-001  4.816E-017
7.286E-001  2.724E-002  2.556100024352868E-001  5.179E-017
7.552E-001  2.662E-002  2.734940575381486E-001  5.106E-017
7.812E-001  2.606E-002  2.914313637835756E-001  4.196E-017
8.074E-001  2.614E-002  3.098312738204417E-001  5.378E-017
8.328E-001  2.542E-002  3.280927553321492E-001  4.896E-017
8.578E-001  2.502E-002  3.463933848953307E-001  4.830E-017
8.825E-001  2.466E-002  3.647320761925693E-001  4.765E-017
9.068E-001  2.435E-002  3.831099196081596E-001  4.702E-017
9.309E-001  2.408E-002  4.015298626348372E-001  4.640E-017
9.548E-001  2.386E-002  4.199964400712813E-001  4.579E-017
9.784E-001  2.367E-002  4.385155462255019E-001  4.144E-017
1.000E+000  2.157E-002  4.555445260818736E-001  2.978E-017
x(1)=4.555445260818736E-001

```

```

Fehler bei t=1: 0.000000000000000E+0000
maxerr = 6.880006151182192E-017
Anzahl der Schritte = 30

```

```

eps=1E-16
(22) RKF7(8)_13, q=8, mit besserer SWS
  t           h           eta           err
0.000E+000  0.000E+000  0.000000000000000E+000  0.000E+000
6.250E-002  6.250E-002  1.953077317739325E-003  9.102E-019
1.131E-001  5.061E-002  6.395526732943032E-003  8.055E-019
1.547E-001  4.161E-002  1.196334906090303E-002  1.691E-019
1.963E-001  4.158E-002  1.924986437084060E-002  1.690E-019
2.378E-001  4.155E-002  2.824563092286752E-002  5.629E-020
2.855E-001  4.764E-002  4.065506942421402E-002  3.227E-019
3.294E-001  4.391E-002  5.405731250074270E-002  1.785E-018
3.948E-001  6.537E-002  7.744478974018228E-002  4.464E-017
4.273E-001  3.254E-002  9.058961285454700E-002  8.816E-019
4.802E-001  5.290E-002  1.140407889554802E-001  4.816E-017
5.324E-001  5.217E-002  1.396125227157767E-001  7.321E-017
5.812E-001  4.881E-002  1.656542478576782E-001  6.719E-017
6.274E-001  4.617E-002  1.920775371294269E-001  6.155E-017
6.715E-001  4.415E-002  2.188881397819977E-001  5.814E-017
7.140E-001  4.253E-002  2.460503614556533E-001  5.577E-017
7.552E-001  4.117E-002  2.735168787708678E-001  5.399E-017
7.952E-001  4.002E-002  3.012369028011975E-001  5.292E-017
8.342E-001  3.900E-002  3.291380614584491E-001  5.178E-017
8.723E-001  3.811E-002  3.571699598291049E-001  5.039E-017
9.097E-001  3.737E-002  3.853154976961794E-001  5.002E-017
9.464E-001  3.667E-002  4.134995158930803E-001  4.909E-017
9.825E-001  3.608E-002  4.416957721718116E-001  4.887E-017
1.000E+000  1.753E-002  4.555445260818737E-001  1.900E-019
x(1)=4.555445260818736E-001

```

```

Fehler bei t=1: -5.5511151231257827E-0017
maxerr = 7.321489097377977E-017
Anzahl der Schritte = 23
-----

```

Für die Riccati-DGL lässt das RKF7(8)₁₃ mit dem Fehlerschätzer $err_{(4)}$ etwas größere Schrittweiten zu, sodass die Schrittzahl kleiner ist. Dem Leser überlassen wir weitere Testrechnungen.

Kapitel 6

Lineare Mehrschrittverfahren

6.1 Das Konzept der linearen Mehrschrittverfahren

Das Konzept der ESV besteht in der Berechnung einer Approximation η_{i+1} aus dem vorhergehenden Wert η_i . Um hierbei Verfahren höherer Ordnung zu erzielen, wie z.B. bei RKV, ist es in jedem Schritt nötig, die Funktion $f(t, x)$ an mehreren Zwischenstellen auszuwerten. Die linearen Mehrschrittverfahren (MSV) dagegen stellen Berechnungsverfahren für η_{i+1} dar, für welche die bereits bekannten Werte $\eta_i, \eta_{i-1}, \dots, \eta_{i-k}$ verwendet werden. Die zugrunde liegenden Ideen entstammen der Interpolationstheorie sowie der numerischen Integration unter Verwendung von entsprechenden Knoten (t_j, η_j) .

Dies illustrieren die folgenden Beispiele zur Lösung des skalaren AWP

$$x' = f(t, x), \quad x(t_0) = x_0, \quad t \in [t_0, T]. \quad (6.1)$$

Die exakte Lösung werde wieder mit $x(t)$ bezeichnet. Wie zuvor seien zu einer vorgegebenen Schrittweite h die diskreten äquidistanten Stützstellen $t_i = t_0 + i h$ definiert.

Beispiel 6.1

Durch Integration der gDgl des AWP folgt

$$\int_{t_i}^{t_{i+1}} x'(t) dt = x(t_{i+1}) - x(t_i) = \int_{t_i}^{t_{i+1}} f(t, x(t)) dt. \quad (6.2)$$

Bezeichnen wir mit

$$F(t) = f(t, x(t))$$

den Integranden, so erhalten wir die exakte Formel

$$x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} F(t) dt. \quad (6.3)$$

Zu ihrer weiteren Umformung erinnert man sich an die Ideen zur Herleitung der Newton-Côtes-Formeln zur numerischen Integration.

(a) Wir ersetzen den Integranden $F(t)$ durch das lineare Interpolationspolynom an den Punkten (t_{i-1}, F_{i-1}) und (t_i, F_i) , wobei $F_k = F(t_k)$, gemäß

$$\begin{aligned} p_1(t) &= \frac{t-t_{i-1}}{t_i-t_{i-1}}F_i + \frac{t-t_i}{t_{i-1}-t_i}F_{i-1} = \frac{t-t_{i-1}}{h}F_i - \frac{t-t_i}{h}F_{i-1} = \\ &= F_{i-1} + \frac{F_i-F_{i-1}}{t_i-t_{i-1}}(t-t_{i-1}) = F_i + \frac{F_{i-1}-F_i}{t_{i-1}-t_i}(t-t_i) = F_i + \frac{F_i-F_{i-1}}{h}(t-t_i). \end{aligned}$$

Setzen wir $p_1(t)$ anstelle von $F(t)$ in die Gleichung (6.3) ein und integrieren, so folgt nach einfacher Rechnung

$$x(t_{i+1}) \approx x(t_i) + h \left[\frac{3}{2}f(t_i, x(t_i)) - \frac{1}{2}f(t_{i-1}, x(t_{i-1})) \right].$$

Dies führt zum expliziten MSV

$$\eta_{i+1} = \eta_i + h \left[\frac{3}{2}f(t_i, \eta_i) - \frac{1}{2}f(t_{i-1}, \eta_{i-1}) \right], \quad (6.4)$$

explizit deshalb, da η_{i+1} durch diese Formel direkt aus den Werten η_i und η_{i-1} berechnet werden kann.

(b) Ersetzen wir $F(t)$ durch das lineare Interpolationspolynom zu den Punkten (t_i, F_i) und (t_{i+1}, F_{i+1})

$$p_1(t) = F_i + \frac{F_{i+1}-F_i}{h}(t-t_i),$$

so erhalten wir

$$x(t_{i+1}) \approx x(t_i) + h \left[\frac{1}{2}f(t_{i+1}, x(t_{i+1})) + \frac{1}{2}f(t_i, x(t_i)) \right]$$

und das entsprechende implizite MSV

$$\eta_{i+1} = \eta_i + \frac{h}{2}[f(t_{i+1}, \eta_{i+1}) + f(t_i, \eta_i)], \quad (6.5)$$

implizit deshalb, weil die Gleichung nicht unmittelbar nach η_{i+1} aufgelöst werden kann.

(c) Ersetzen wir schließlich $F(t)$ durch das quadratische Interpolationspolynom $p_2(t)$ an den Punkten (t_{i-2}, F_{i-2}) , (t_{i-1}, F_{i-1}) und (t_i, F_i) gemäß Lagrange¹ oder Newton², so

¹Joseph Louis de Lagrange (1736-1813), französischer Mathematiker und Physiker, Arbeiten zur Variationsrechnung, Differentialgleichungen, Mechanik, Zahlentheorie, Analysis, Algebra

²Sir Isaac Newton (1642-1727) englischer Mathematiker, Physiker, Astronom und Philosoph, Mitbegründer der Theorie der Differential- und Integralrechnung, u.a. Arbeiten zu algebraischen Gleichungen, symmetrischen Funktionen, Interpolation

folgen zunächst die verschiedenen Darstellungen des Interpolationspolynoms

$$\begin{aligned}
 p_2(t) &= \frac{(t-t_{i-2})(t-t_{i-1})}{(t_i-t_{i-2})(t_i-t_{i-1})}F_i + \frac{(t-t_{i-2})(t-t_i)}{(t_{i-1}-t_{i-2})(t_{i-1}-t_i)}F_{i-1} + \frac{(t-t_{i-1})(t-t_i)}{(t_{i-2}-t_{i-1})(t_{i-2}-t_i)}F_{i-2} \\
 &= \left[\frac{1}{2h^2}(t-t_i)^2 + \frac{3}{2h}(t-t_i)+1 \right] F_i - \left[\frac{1}{h^2}(t-t_i)^2 + \frac{2}{h}(t-t_i) \right] F_{i-1} \\
 &\quad + \left[\frac{1}{2h^2}(t-t_i)^2 + \frac{1}{2h}(t-t_i) \right] F_{i-2} \\
 &= F_i + F[t_i, t_{i-1}](t-t_i) + F[t_i, t_{i-1}, t_{i-2}](t-t_i)(t-t_{i-1}) \\
 &= F_i + \frac{1}{2h^2}[(t-t_i)^2(F_{i-2} - 2F_{i-1} + F_i) + h(t-t_i)(F_{i-2} - 4F_{i-1} + 3F_i)].
 \end{aligned}$$

Daraus ergeben sich die Näherung

$$x(t_{i+1}) \approx x(t_i) + h \left[\frac{23}{12}f(t_i, x(t_i)) - \frac{16}{12}f(t_{i-1}, x(t_{i-1})) + \frac{5}{12}f(t_{i-2}, x(t_{i-2})) \right]$$

sowie das explizite MSV

$$\eta_{i+1} = \eta_i + \frac{h}{12}[23f(t_i, \eta_i) - 16f(t_{i-1}, \eta_{i-1}) + 5f(t_{i-2}, \eta_{i-2})]. \quad (6.6)$$

(d) Nehmen wir in der Formel (6.2) als Intervall $[t_{i-1}, t_{i+1}]$ und ersetzen $F(t)$ durch das quadratische Interpolationspolynom $p_2(t)$ an den Punkten (t_{i-1}, F_{i-1}) , (t_i, F_i) und (t_{i+1}, F_{i+1}) gemäß der Lagrange-Formel

$$p_2(t) = \frac{(t-t_{i-1})(t-t_i)}{(t_{i+1}-t_{i-1})(t_{i+1}-t_i)}F_{i+1} + \frac{(t-t_{i-1})(t-t_{i+1})}{(t_i-t_{i-1})(t_i-t_{i+1})}F_i + \frac{(t-t_i)(t-t_{i+1})}{(t_{i-1}-t_i)(t_{i-1}-t_{i+1})}F_{i-1},$$

so folgen als Quadraturformel die Simpson³-Regel mit den Gewichten $\frac{1}{6}, \frac{4}{6}, \frac{1}{6}$, damit die Näherung

$$x(t_{i+1}) \approx x(t_{i-1}) + 2h \left[\frac{1}{6}f(t_{i+1}, x(t_{i+1})) + \frac{4}{6}f(t_i, x(t_i)) + \frac{1}{6}f(t_{i-1}, x(t_{i-1})) \right]$$

sowie das implizite MSV

$$\eta_{i+1} = \eta_{i-1} + \frac{h}{3}[f(t_{i+1}, \eta_{i+1}) + 4f(t_i, \eta_i) + f(t_{i-1}, \eta_{i-1})]. \quad (6.7)$$

Übung 6.1 Man rechne die Darstellung der Interpolationspolynome $p_2(t)$ mit Hilfe der angegebenen Lagrange- bzw. Newton-Polynome nach.

In den Beispielen sind drei Aspekte deutlich zu erkennen:

- explizite oder implizite Verfahren,
- Anzahl der Stufen oder Schritte eines MSV,
- Wahl des Integrationsintervalls und der Knoten für die Interpolation.

Die oben erhaltenen Ansätze werden somit wie folgt verallgemeinert.

³Thomas Simpson (1710-1761), englischer Mathematiker

Definition 6.1 Ein Verfahren der allgemeinen einheitlichen Form

$$\begin{aligned} a_r \eta_{i+r} + a_{r-1} \eta_{i+r-1} + \dots + a_1 \eta_{i+1} + a_0 \eta_i &= h (b_r f_{i+r} + \dots + b_1 f_{i+1} + b_0 f_i), \\ a_r &= 1, \quad i \geq 0, \quad f_k = f(t_k, \eta_k), \end{aligned} \quad (6.8)$$

heißt lineares r -stufiges MSV bzw. lineares r -Schritt-Verfahren.

Ist $b_r = 0$, so heißt das MSV explizit, andernfalls implizit.

Als Startgrößen werden dabei $\eta_0, \eta_1, \dots, \eta_{r-1}$ benötigt.

Die Anzahl der Stufen r ergibt sich als Differenz von maximalem und minimalem t -Index.

Das ist die Anzahl der Teilintervalle, über die sich die Formel erstreckt.

In den Beispielen 6.1 haben die MSV in (a), (d) die Stufenzahl 2, in (b) 1 und in (c) 3. Somit beschreibt nach dieser Definition (a) das explizite 2-stufige MSV

$$\eta_{i+2} - \eta_{i+1} = h \left[\frac{3}{2} f(t_{i+1}, \eta_{i+1}) - \frac{1}{2} f(t_i, \eta_i) \right].$$

Die Formel (b) beschreibt das implizite 1-Schritt-Verfahren und damit insbesondere also das ESV

$$\eta_{i+1} - \eta_i = \frac{h}{2} [f(t_{i+1}, \eta_{i+1}) + f(t_i, \eta_i)].$$

Durch (c) ist das explizite 3-Schritt-Verfahren definiert gemäß

$$\eta_{i+3} - \eta_{i+2} = \frac{h}{12} [23f(t_{i+2}, \eta_{i+2}) - 16f(t_{i+1}, \eta_{i+1}) + 5f(t_i, \eta_i)].$$

Mit (d) haben wir das implizite 2-Schritt-Verfahren

$$\eta_{i+1} - \eta_{i-1} = \frac{h}{3} [f(t_{i+1}, \eta_{i+1}) + 4f(t_i, \eta_i) + f(t_{i-1}, \eta_{i-1})].$$

Anstelle der Formel des MSV in der Definition 6.1 gibt es auch noch weitere gebräuchliche Darstellungen. Dabei beachte man jedoch die teilweise andere Bezeichnung und Nummerierung der Koeffizienten.

Eine zweite etwas anschaulichere, aber spezielle Darstellung ist (Stufenanzahl $s + p$)

$$\begin{aligned} \eta_{i+p} &= \eta_{i-q} + h \sum_{j=0}^r \beta_j f_{i-j} = \eta_{i-q} + h (\beta_0 f_i + \beta_1 f_{i-1} + \dots + \beta_r f_{i-r}), \\ p \geq 0, \quad q \geq 0, \quad i \geq s = \max(r, q), \quad \eta_0, \eta_1, \dots, \eta_{s+p-1} &\text{ gegeben.} \end{aligned} \quad (6.9)$$

Eine dritte besondere implizite Form ist (Stufenanzahl r)

$$\sum_{j=0}^r a_j \eta_{i+j} = h b_r f_{i+r}, \quad a_r, b_r \neq 0, \quad i \geq 0, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben.} \quad (6.10)$$

Eine vierte Möglichkeit der Notation von impliziten MSV ist (Stufenanzahl r)

$$\begin{aligned} \eta_{m+1} &= \sum_{j=1}^r \alpha_j \eta_{m+1-j} + h \sum_{j=0}^r \beta_j f_{m+1-j}, \quad m \geq r-1, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben,} \\ &= h \beta_0 f(x_{m+1}, \eta_{m+1}) + \sum_{j=1}^r (\alpha_j \eta_{m+1-j} + h \beta_j f_{m+1-j}). \end{aligned} \quad (6.11)$$

Die Bestimmung von η_{m+1} in (6.11) bzw. in den anderen impliziten Versionen von MSV erfolgt meist iterativ bei vorgegebenem $\eta_{m+1}^{(0)}$ (dies z.B. aus einem expliziten MSV) mit

$$\eta_{m+1}^{(k+1)} = h\beta_0 f(t_{m+1}, \eta_{m+1}^{(k)}) + C = g(\eta_{m+1}^{(k)}), \quad k = 0, 1, \dots \quad (6.12)$$

Bei der Herleitung der Formeln (6.8) – (6.11) wird gemäß ihrem interpolativen Charakter und bei Verwendung von Interpolationspolynomen von Lagrange oder Newton, Schema der dividierten Differenzen, Integration von Polynomen, Newton-Cotes-Formeln der Integration, Vorwärts- bzw. Rückwärtsdifferenzen oftmals die Darstellung mittels der Rückwärtsdifferenzen

$$\nabla^j f_i = \nabla^{j-1} f_i - \nabla^{j-1} f_{i-1}, \quad \nabla^0 f_i = f_i \quad (6.13)$$

genommen. Deshalb gehören MSV auch zur Gruppe der Differenzgleichungen.

Jede Summe von Funktionswerten in den MSV kann in eindeutiger Weise mittels Rückwärtsdifferenzen notiert werden.

Aus den Ansätzen (6.8) – (6.11) lassen sich die grundlegenden Klassen von MSV bei entsprechender Wahl der Parameter p, q, r herleiten.

Formel (6.9)

$p \geq 1$: explizite MSV,

$p = 1, q = 0$: explizite $(r + 1)$ -stufige Adams-Bashforth-Formeln der Ordnung $r + 1$,

$p = 1, q = 1$: explizite s -stufige Nyström -Formeln, $s = \max(r + 1, 2)$,

$p = 0, \beta_0 = 0$: explizite MSV,

$p = 0, \beta_0 \neq 0$: implizite MSV,

$p = 0, q = 1$: implizite s -stufige Adams-Moulton-Formeln der Ordnung $r + 1$, $s = \max(r, 1)$,

$p = 0, q = 2$: implizite s -stufige Milne-Simpson-Formeln aus der Klasse vom Adams-Störmer-Typ, $s = \max(r, 2)$,

Formel (6.10) : Rückwärtige Differenzenformeln (engl. Backward Difference Formulas, BDF), steif-stabile Gear-Methoden.

Wir stellen im Folgenden daraus drei wichtige Klassen von MSV vor.

6.1.1 Adams-Bashforth-Verfahren

Adams⁴-Bashforth⁵-Verfahren sind explizite, bei denen in Verallgemeinerung zu den Beispielen 6.1 (a), (c) die Funktion $F(t)$ aus Gleichung (6.3) im Intervall $[t_i, t_{i+1}]$ durch das Interpolationspolynom an den Knoten $t_{i-r}, \dots, t_{i-1}, t_i$ approximiert wird.

⁴John Couch Adams (1819-1892), englischer Mathematiker und Astronom

⁵Francis Bashforth (1819-1912), englischer Physiker, Ballistik, Bashforth Chronograph

Wir suchen also für die Werte $r = 0, 1, 2, \dots$ die spezielle Darstellung als $(r + 1)$ -stufiges MSV

$$\eta_{i+1} = \eta_i + h \sum_{j=0}^r \beta_j f_{i-j} = \eta_i + h (\beta_0 f_i + \beta_1 f_{i-1} + \dots + \beta_r f_{i-r}). \quad (6.14)$$

Dazu brauchen wir das Lagrange-Interpolationspolynom

$$p_r(t) = \sum_{j=0}^r F_{i-j} L_{i-j}(t)$$

mit den zugehörigen Lagrange-Basispolynomen $L_{i-j}(t)$.

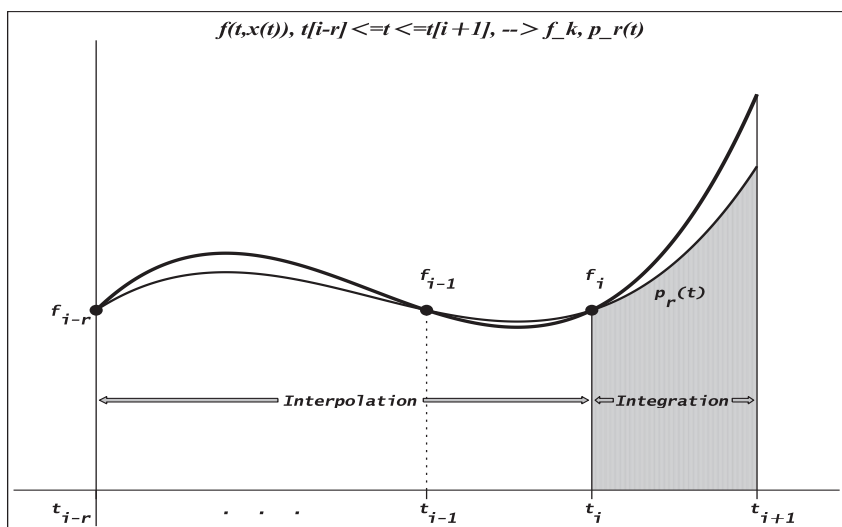


Abb. 6.1 Datei *msv_schritt1_02.ps*,

Interpolation und Integration beim MSV von Adams-Bashforth

Zwecks Vereinfachung der Rechnung seien die Knoten $t_0, \dots, t_{r-1}, t_r, t_{r+1}$ (einfache Indexverschiebung). Dann ist

$$p_r(t) = \sum_{k=0}^r F_k L_k(t)$$

mit

$$L_k(t) = \prod_{\substack{l=0 \\ l \neq k}}^r \frac{t - t_l}{t_k - t_l}, \quad k = 0, 1, \dots, r.$$

Die Integration von $p_r(t)$ anstatt von $F(t)$ liefert mit der Bereichstransformation $t = t_0 + sh$, $t_k = t_0 + kh$,

$$\int_{t_r}^{t_{r+1}} p_r(t) dt = \sum_{k=0}^r F_k \int_{t_r}^{t_{r+1}} \prod_{\substack{l=0 \\ l \neq k}}^r \frac{t - t_l}{t_k - t_l} dt = \sum_{k=0}^r h F_k \int_r^{r+1} \prod_{\substack{l=0 \\ l \neq k}}^r \frac{s - l}{k - l} ds$$

und somit die gesuchten Koeffizienten der MSV

$$\beta_k^{(r)} = \int_r^{r+1} \prod_{\substack{l=0 \\ l \neq k}}^r \frac{s-l}{k-l} ds, \quad k = 0, 1, \dots, r, \quad (6.15)$$

$$\begin{aligned} &= \frac{(-1)^{r-k}}{k!(r-k)!} \int_r^{r+1} \prod_{\substack{l=0 \\ l \neq k}}^r (s-l) ds, \quad s = z+r, \quad z \in [0, 1], \\ &= \frac{(-1)^{r-k}}{k!(r-k)!} \int_0^1 \prod_{\substack{l=0 \\ l \neq k}}^r (z+r-l) dz \\ &= \frac{(-1)^{r-k}}{k!(r-k)!} \int_0^1 \frac{1}{z+r-k} \prod_{l=0}^r (z+l) dz, \quad r-k = j, \\ \beta_{r-j}^{(r)} &= \frac{(-1)^j}{j!(r-j)!} \int_0^1 \frac{1}{z+j} \prod_{l=0}^r (z+l) dz, \quad j = 0, 1, \dots, r. \end{aligned} \quad (6.16)$$

Damit zeigen wir auch den Weg zur Vereinfachung und Auswertung der Formel für die Koeffizienten des MSV.

Wir stellen die Koeffizienten in einer Tabelle zusammen.

$\beta_k^{(r)}$	$k=r-j$					
	0	1	2	3	4	5
$\beta_k^{(0)} = \beta_{0-j}^{(0)}$	1					
$\beta_k^{(1)} = \beta_{1-j}^{(1)}$	$-\frac{1}{2}$	$\frac{3}{2}$				
$\beta_k^{(2)} = \beta_{2-j}^{(2)}$	$\frac{5}{12}$	$-\frac{16}{12}$	$\frac{23}{12}$			
$\beta_k^{(3)} = \beta_{3-j}^{(3)}$	$-\frac{9}{24}$	$\frac{37}{24}$	$-\frac{59}{24}$	$\frac{55}{24}$		
$\beta_k^{(4)} = \beta_{4-j}^{(4)}$	$\frac{251}{720}$	$-\frac{1274}{720}$	$\frac{2616}{720}$	$-\frac{2774}{720}$	$\frac{1901}{720}$	
$\beta_k^{(5)} = \beta_{5-j}^{(5)}$	$-\frac{475}{1440}$	$\frac{2877}{1440}$	$-\frac{7298}{1440}$	$\frac{9982}{1440}$	$-\frac{7923}{1440}$	$\frac{4277}{1440}$

Tab. 6.1 Koeffizienten der $(r+1)$ -stufigen Adams-Bashforth-Verfahren, $r = 0, 1, \dots, 5$

Mit $f_k = f(t_k, \eta_k)$ ist durch

$$\eta_{i+1} - \eta_i = h(\beta_r^{(r)} f_i + \beta_{r-1}^{(r)} f_{i-1} + \dots + \beta_0^{(r)} f_{i-r}) \quad (6.17)$$

das $(r+1)$ -stufige Adams-Bashforth-Verfahren gegeben.

Definieren wir den lokalen Diskretisierungsfehler wie im Fall der ESV, so hat dieses Verfahren die Ordnung $r+1$ (vergl. Abschnitt 6.2).

Wegen der einheitlichen Faktoren (Nenner) in den Koeffizienten $\beta_{r-j}^{(r)}$, $j = 0, 1, \dots, r$, und

aus Gründen ihrer besseren Zuordnung zu den Funktionswerten $f_i, f_{i-1}, \dots, f_{i-r}$ wie in der Formel (6.14) schreiben wir das erweiterte Koeffiziententableau mit $\beta_j = \beta_{r-j}^{(r)}$ und der Konstanten c im Fehlerglied folgendermaßen.

r	Ordnung = Stufen $n=r+1$	Fehler- konstante $ch^n f^{(n+1)}$	K	j						
				0	1	2	3	4	5	
0	1	$\frac{1}{2}$	1	1						
1	2	$\frac{5}{21}$	2	3	-1					
2	3	$\frac{3}{8}$	12	23	-16	5				
3	4	$\frac{251}{720}$	24	55	-59	37	-9			
4	5	$\frac{95}{288}$	720	1901	-2774	2616	-1274	251		
5	6	$\frac{19087}{60480}$	1440	4277	-7923	9982	-7298	2877	-475	

Tab. 6.2 Koeffizienten der n -stufigen Adams-Bashforth(n)-Verfahren (AB(n))

n -ter Ordnung, $n = r + 1$,

$$\eta_{i+1} - \eta_i = h \sum_{j=0}^r \beta_j f_{i-j}, \quad i = r, r+1, \dots, \eta_0, \eta_1, \dots, \eta_r \text{ gegeben}$$

Das Adams-Bashforth(1)-Verfahren ist das PZV.

Das $(r+1)$ -stufige Verfahren kann erstmalig angewendet werden, wenn die Werte η_0, \dots, η_r bekannt sind. Diese Werte müssen durch andere Verfahren, etwa ESV der gleichen Ordnung, berechnet werden. Das Schema des numerischen Verfahrens zur Approximation von $x(t)$ im Intervall $[t_0, T]$ sieht demnach wie folgt aus.

Algorithmus 6.1 $(r+1)$ -stufiges Adams-Bashforth-Verfahren

S1 Initialisierung

Man wähle die Schrittzahl $N \in \mathbb{N}$ und Ordnung $r+1$,
man definiere $h = (T - t_0)/N$ und setze $\eta_0 = x_0$.

S2 Anlaufrechnung

Man wähle ein ESV $\Psi(t, x, h)$ mit derselben Ordnung
und berechne $\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h)$ für $i = 0, 1, \dots, r-1$.

S3 Einsatz des Adams-Bashforth-Verfahrens

Für $i = r, r+1, \dots, N-1$ berechne man $\eta_{i+1} = \eta_i + h \sum_{j=0}^r \beta_j f(t_{i-j}, \eta_{i-j})$.

Beispiel 6.2 Wir betrachten das AWP aus Beispiel 5.10 (c).

$$x' = f(t, x) = 10 \left(x - \frac{t^2}{1+t^2} \right) + \frac{2t}{(1+t^2)^2}, \quad t \in [t_0, T] = [0, 3], \quad x(t_0) = x_0 = 0,$$

mit der exakten Lösung $x(t) = \frac{t^2}{1+t^2}$.

Wir wollen das Adams-Bashforth-Verfahren mit RKV vergleichen. Dabei soll die Anlaufrechnung zum MSV mit den RKV erfolgen. Aus dem Abschnitt 5.2 nutzen wir das MATLAB-File `rkv.m` der vektoriellen Version des expliziten RKV.

Wir testen und vergleichen die folgenden Varianten:

- Adams-Bashforth(3) mit (3,3)-RKV (Kutta-Verfahren),
- Adams-Bashforth(4) mit KRKV,
- (3,3)-RKV,
- KRKV.

MATLAB-File zur vektoriellen Version des Adams-Bashforth-Verfahrens

```
% ab.m
% (r+1)-stufiges Adams-Bashforth-Verfahren der Ordnung r+1
% hier: beta(j)->beta(j+1),j=0,...,r; Ordnung=max(size(beta))

function [t,x] = ab(A,b,c,beta,f,t0,T,x0,N)
    h = (T-t0)/N;
    r = size(beta,1); % max(size(beta))
    n = size(x0,1);
    fr = zeros(n,N);
    % Anlaufrechnung
    [t,x] = rkv(A,b,c,f,t0,t0+(r-1)*h,x0,r-1);
    % Funktionswerte zur Anlaufrechnung
    for i=1:r
        fr(:,i) = feval(f,t(i),x(:,i));
    end;
    % Naechste Schritte
    for i=r:N
        t(i+1) = t0+i*h;
        x(:,i+1) = x(:,i)+h*fr(:,i:-1:i-r+1)*beta;
        fr(:,i+1) = feval(f,t(i+1),x(:,i+1));
    end;
```

Rahmenprogramm zum Test von Adams-Bashforth-Verfahren und RKV

```
% ab_rkv.m
% Test von AB, RKV fuer gDgl
% --> ab.m, rkv.m --> f1.m, flxv.m ( x(t)=t^2/(1+t^2) )

function [t,x,t1,x1,t2,x2,t3,x3,t4,x4] = ab_rkv(N)
% N = Anzahl der Schritte im Intervall [t0,T]

% 3-Schritt Adams-Bashforth -- Ordnung 3
beta1 = [ 23/12 -16/12 5/12]';

% 4-Schritt Adams-Bashforth -- Ordnung 4
beta2 = [ 55/24 -59/24 37/24 -9/24]';

% Kutta-Verfahren, KV -- Ordnung 3
A1 = [ 0 0 0
        0.5 0 0
        -1.0 2 0];
c1 = [ 0 0.5 1]'; b1 = [1/6 2/3 1/6]';
```

```

% KRKV -- Ordnung 4
A2 = [0 0 0 0
      0.5 0 0 0
      0 0.5 0 0
      0 0 1 0];
c2 = [0 0.5 0.5 1]'; b2 = [1/6 1/3 1/3 1/6]';

% Integrationsintervall
t0 = 0;
T = 3;
% Startwert
x0 = 0;
% Numerische Loesung: AB, RKV
[t1,x1] = ab(A1,b1,c1,beta1,'f1',t0,T,x0,N); % Anlaufrechnung mit KV
[t2,x2] = ab(A2,b2,c2,beta2,'f1',t0,T,x0,N); % Anlaufrechnung mit KRKV
[t3,x3] = rkv(A1,b1,c1,'f1',t0,T,x0,N);
[t4,x4] = rkv(A2,b2,c2,'f1',t0,T,x0,N);

% Exakte Loesung x(t)
t = t0:(T-t0)/(10*N):T;
x = flxv(t);

% Grafik
plot(t,x,'k-',t1,x1,'b-',t2,x2,'r:',t3,x3,'g--',t4,x4,'m-.',t,0*t,'k-');
axis([ t0 T -0.1 1]);
xlabel('t');
legend('x(t)', 'AB(3)', 'AB(4)', '(3,3)-KV', 'KRKV', 4);
text(2.2,0.8,'x(t)=t^2/(1+t^2)');

% bei N=100
text(0.6,0.2,'(3,3)-KV'); text(1.3,0.2,'KRKV');
text(1.0,0.8,'AB(3)'); text(1.0,0.2,'AB(4)');
print ab_rkv1.ps -dpsc

```

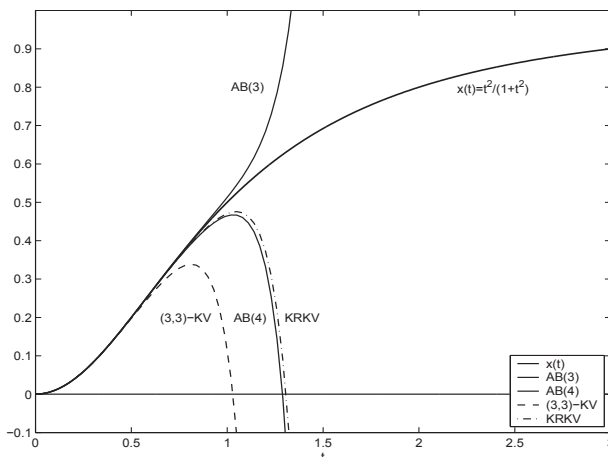


Abb. 6.2

Datei *ab_rkv1.ps*,
 Adams-Bashforth-Verfahren
 mit Anlaufrechnung (RKV)
 und RKV zum AWP
 $x' = f(t, x)$, $x(t_0) = x_0$,
 in $[0, 3] \times [-0.1, 1]$

Auffällig ist, dass für beliebige Schrittweiten $h = (T - t_0)/N$ das Adams-Bashforth(4)-Verfahren und das KRKV sehr ähnliche Lösungsverläufe zeigen.

6.1.2 Adams-Moulton-Verfahren

Die impliziten Adams-Moulton⁶-Verfahren erhält man, wenn man die Funktion $F(t)$ im Intervall $[t_{i-1}, t_i]$ durch das Interpolationspolynom an den Knoten $t_{i-r}, \dots, t_{i-1}, t_i$ approximiert. Das s -Schritt-Adams-Moulton-Verfahren mit $s = \max(r, 1) \geq 1$ hat die Form

$$\eta_i = \eta_{i-1} + h \sum_{j=0}^r \beta_j f_{i-j} = \eta_{i-1} + h (\beta_0 f_i + \beta_1 f_{i-1} + \dots + \beta_r f_{i-r}). \quad (6.18)$$

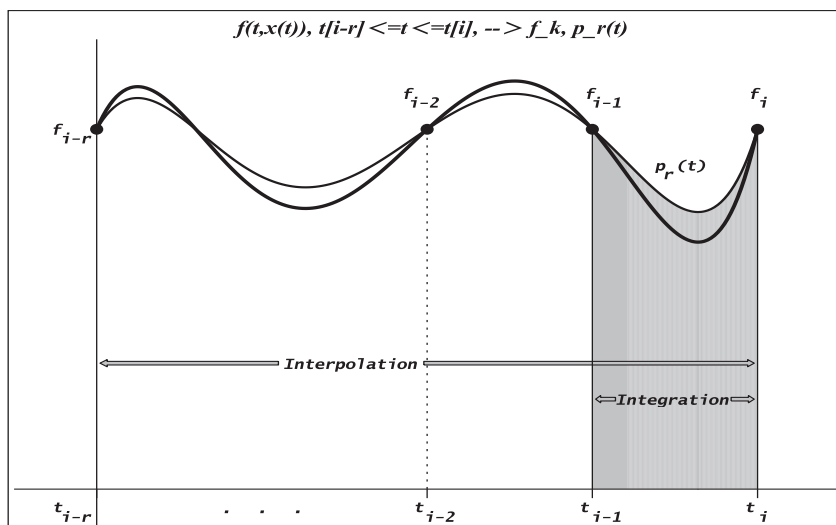


Abb. 6.3 Datei *msv_schritt1_03.ps*,

Interpolation und Integration beim MSV von Adams-Moulton

Bei der Bestimmung der Koeffizienten verfährt man wie beim Adams-Bashforth-Verfahren. In dem Ansatz

$$\eta_i - \eta_{i-1} = h(\beta_r^{(r)} f_i + \beta_{r-1}^{(r)} f_{i-1} + \dots + \beta_0^{(r)} f_{i-r}), \quad f_k = f(t_k, \eta_k), \quad (6.19)$$

werden die Koeffizienten $\beta_k^{(r)}$ berechnet, einschließlich des impliziten ESV für $r = 0$ und $\beta_0^{(0)} = 1$.

$$\beta_k^{(r)} = \int_{r-1}^r \prod_{\substack{l=0 \\ l \neq k}}^r \frac{s-l}{k-l} ds, \quad k = 0, 1, \dots, r. \quad (6.20)$$

⁶Forest Ray Moulton (1872-1952), amerikanischer Physiker und Astronom, Differentialgleichungen und Ballistik

$\beta_k^{(r)}$	k					
	0	1	2	3	4	5
$\beta_k^{(0)}$	1					
$\beta_k^{(1)}$	$\frac{1}{2}$	$\frac{1}{2}$				
$\beta_k^{(2)}$	$-\frac{1}{12}$	$\frac{8}{12}$	$\frac{5}{12}$			
$\beta_k^{(3)}$	$\frac{1}{24}$	$-\frac{5}{24}$	$\frac{19}{24}$	$\frac{9}{24}$		
$\beta_k^{(4)}$	$-\frac{19}{720}$	$\frac{106}{720}$	$-\frac{264}{720}$	$\frac{646}{720}$	$\frac{251}{720}$	
$\beta_k^{(5)}$	$\frac{27}{1440}$	$-\frac{173}{1440}$	$\frac{482}{1440}$	$-\frac{798}{1440}$	$\frac{1427}{1440}$	$\frac{475}{1440}$

Tab. 6.3 Koeffizienten der s -stufigen Adams-Moulton-Verfahren, $s = 1, 2, \dots, 5$

Als klassisches Adams-Moulton-Verfahren wird das Verfahren für $s = r = 4$ bezeichnet. Mit den Kriterien des Abschnitts 6.2 kann festgestellt werden, dass das s -Schritt-Adams-Moulton-Verfahren von der Ordnung $r + 1$ ist.

Wegen der einheitlichen Faktoren (Nenner) in den Koeffizienten $\beta_k^{(r)}$, $k = 0, 1, \dots, r$, aus Gründen ihrer besseren Zuordnung zu den Funktionswerten $f_i, f_{i-1}, \dots, f_{i-r}$ wie in der Formel (6.18) und der Angabe der Stufenzahl s notieren wir das erweiterte Koeffiziententableau mit $\beta_j = \beta_{r-j}^{(r)}$ und der Konstanten c im Fehlerglied.

r	Stuf. s	Ordnung $n=r+1$	Fehler- konstante $ch^n f^{(n+1)}$	K	j						
					0	1	2	3	4	5	
					$K \cdot \beta_j$						
0	1	1	$-\frac{1}{2}$	1	1						
1	1	2	$-\frac{1}{12}$	2	1	1					
2	2	3	$-\frac{1}{24}$	12	5	8	-1				
3	3	4	$-\frac{19}{720}$	24	9	19	-5	1			
4	4	5	$-\frac{3}{160}$	720	251	646	-264	106	-19		
5	5	6	$-\frac{863}{60480}$	1440	475	1427	-798	482	-173	27	

Tab. 6.4 Koeffizienten der s -stufigen Adams-Moulton(n)-Verfahren (AM(n)),
 $s = \max(r, 1) = 1, 2, \dots, 5$, der Ordnung $n = r + 1$, $r = 0, 1, \dots, 5$,
 $\eta_i = \eta_{i-1} + h \sum_{j=0}^r \beta_j f_{i-j}$, $i = r, r + 1, \dots$, $\eta_0, \eta_1, \dots, \eta_{r-1}$ gegeben

Das Adams-Moulton(1)-Verfahren ist das IEV.

Als Schwierigkeit gegenüber den Adams-Bashforth-Verfahren ergibt sich hier, dass η_i nicht direkt aus Gleichung (6.18) berechnet werden kann, da diese unbekannte Größe auch in $f_i = f(t_i, \eta_i)$ auftaucht. Es bietet sich hier an, einen Löser für nichtlineare Gleichungen auf die Gleichung $\varphi(z) = 0$ anzuwenden, wobei

$$\varphi(z) = z - \eta_{i-1} - h (\beta_0 f(t_i, z) + \beta_1 f_{i-1} + \dots + \beta_r f_{i-r}) \quad (6.21)$$

ist.

Konvergenzbetrachtung zur iterativen Lösung von impliziten MSV

(a) Fixpunktgleichung, Fixpunktiteration

Wir betrachten die allgemeine Notation von impliziten MSV (6.11) mit Stufenanzahl r .

$$\begin{aligned}
 \eta_i &= \sum_{j=1}^r \alpha_j \eta_{i-j} + h \sum_{j=0}^r \beta_j f_{i-j} \\
 &= h \beta_0 f(t_i, \eta_i) + \sum_{j=1}^r (\alpha_j \eta_{i-j} + h \beta_j f_{i-j}) \\
 &= h \beta_0 f(t_i, \eta_i) + C, \\
 z &= h \beta_0 f(t_i, z) + C \\
 &= g(z), \quad t_i, h, \beta_0, C \text{ Parameter,} \\
 z^{(k+1)} &= g(z^{(k)}).
 \end{aligned}$$

Konvergenzbedingungen

$$|g'(z)| < 1, \quad h |\beta_0| \left| \frac{\partial f(t, z)}{\partial z} \right| < 1, \quad h |\beta_0| \left| \frac{\partial f(t, x)}{\partial x} \right| < 1.$$

Die Bedingung

$$h < \frac{1}{|\beta_0| \max |\partial f(t, x) / \partial x|} \quad (6.22)$$

ist für hinreichend kleines $h \leq h_0$ in der Umgebung des Fixpunktes i. Allg. erfüllbar.

(b) Nullstellengleichung, Newtonsches Iterationsverfahren

$$\begin{aligned}
 \varphi(z) &= z - h \beta_0 f(t_i, z) - C = 0, \quad \varphi'(z) = 1 - h \beta_0 f_z(t_i, z), \\
 z^{(k+1)} &= z^{(k)} - \frac{\varphi(z^{(k)})}{\varphi'(z^{(k)})} \\
 &= z^{(k)} - \frac{z^{(k)} - h \beta_0 f(t, z^{(k)}) - C}{1 - h \beta_0 f_z(t, z^{(k)})}.
 \end{aligned}$$

Falls der Startwert $z^{(0)}$ hinreichend gut ist, dann ist lokal die Konvergenz des Newton-Verfahrens gesichert und es hat quadratische Konvergenzordnung. Somit sind Einschränkungen für h i. Allg. nicht zu befürchten.

Modifikationen des Newton-Verfahrens arbeiten mit Approximationen der Ableitungen durch Differenzenquotienten (Sekantenverfahren) bzw. mit Berechnung der Ableitung nur im ersten Schritt bzw. Neuberechnung jeweils nach m Schritten.

Zur numerischen Lösung wird der Algorithmus 6.1 wie folgt modifiziert.

Algorithmus 6.2 s -stufiges Adams-Moulton-Verfahren

S1 Initialisierung

Man wähle die Schrittzahl $N \in \mathbb{N}$ und Ordnung $n = r + 1$, damit r, s ,
man definiere $h = (T - t_0)/N$ und setze $\eta_0 = x_0$.

S2 Anlaufrechnung

Man wähle ein ESV $\Psi(t, x, h)$ mit derselben Ordnung
und berechne $\eta_i = \eta_{i-1} + h\Psi(t_{i-1}, \eta_{i-1}, h)$ für $i = 1, 2, \dots, r - 1$.

S3 Einsatz des Adams-Moulton-Verfahrens

Für $i = r, r + 1, \dots, N$ berechne man $\eta_i = \eta_{i-1} + h \sum_{j=0}^r \beta_j f(t_{i-j}, \eta_{i-j})$

als Lösung von $\varphi(z) = 0$ (z.B. mit dem Newton-Verfahren) mit $\varphi(\cdot)$ gegeben
durch (6.21). Dazu braucht man in der Regel einen Startwert $z^{(0)} \approx \eta_i$ für
die Iteration (z.B. $z^{(0)} = \tilde{\eta}_i$ aus einem expliziten MSV oder als η_{i-1}).

Beispiel 6.3 Wir betrachten das Adams-Moulton(3)-Verfahren ($s = 2$) für die gDGL

$$x'(t) = t + x(t)^n. \quad (6.23)$$

Aus dem Ansatz

$$\eta_i = \eta_{i-1} + \frac{h}{12} (5f(t_i, \eta_i) + 8f(t_{i-1}, \eta_{i-1}) - f(t_{i-2}, \eta_{i-2}))$$

folgt die Gleichung für die Unbekannte η_i

$$\eta_i - \frac{5h}{12} \eta_i^n = C_i, \quad (6.24)$$

mit der Konstanten

$$C_i = \eta_{i-1} + h \left(\frac{2}{3} \eta_{i-1}^n - \frac{1}{12} \eta_{i-2}^n + t_i \right) + \frac{h^2}{3}.$$

Für $n = 1, 2$ lässt sich die Gleichung leicht nach η_i auflösen, mit dem Ergebnis

$$\begin{aligned} \eta_i &= \frac{12}{12 - 5h} C_i, \quad n = 1, \\ \eta_i &= \frac{6}{5h} \left(1 - \sqrt{1 - \frac{5h}{3} C_i} \right), \quad n = 2. \end{aligned}$$

Für andere Werte n bietet sich das Newton-Verfahren zur Lösung von (6.24) an.

Übung 6.2

Überlegen Sie sich, warum für $n = 2$ die andere Lösung der quadratischen Gleichung (6.24) nicht in Frage kommt. Entwickeln Sie hierzu η_i in eine Reihe bzgl. h .

6.1.3 Prädiktor-Korrektor-Verfahren

Die Lösung der Gleichung $\varphi(z) = 0$ in (6.21) kann eventuell sehr aufwändig sein.

Prädiktor-Korrektor-Verfahren (PC-Verfahren, PCV)⁷ verkürzen diesen Weg, indem die Nullstelle von $\varphi(z)$ nicht exakt, sondern nur näherungsweise berechnet wird.

Hierbei wird zunächst mit Hilfe eines expliziten MSV (6.8) ein Schätzwert $\bar{\eta}_{i+r}$ für η_{i+r} ermittelt (Prädiktor). Anschließend wird in der Funktion $\varphi(z)$ des impliziten MSV (6.8) der Ausdruck $f(t_{i+r}, \eta_{i+r})$ ersetzt durch $f(t_{i+r}, \bar{\eta}_{i+r})$ und η_{i+r} ermittelt als (explizite) Nullstelle der so veränderten Funktion (Korrektor).

Wird die Iteration so lange durchgeführt, bis die Korrektorformel numerisch exakt erfüllt ist, dann spricht man von "Iteration bis zur Konvergenz". Die Konsistenzordnung eines derartigen PCV ist offensichtlich gleich der des Korrektors.

Die Genauigkeit der Prädiktorformel beeinflusst hier lediglich die Anzahl der erforderlichen Iterationsschritte.

In der Praxis wird man allerdings nur eine feste Anzahl von Korrektor-Iterationen ausführen, denn es zeigt sich, dass nach einer gewissen Anzahl von Korrektor-Schritten die Konsistenz- und Konvergenzordnung des PCV nicht mehr verbessert werden kann. Weiterhin sind nach jedem der beiden Schritte noch Nachbesserungen bzw. Modifikationen möglich. Offenbar hat man in den PCV einen großen Spielraum.

Hier bietet sich die einfache Version an, als Prädiktor ein r -Schritt-Adams-Bashforth-Verfahren und als Korrektor ein r -Schritt-Adams-Moulton-Verfahren zu verwenden.

Es zeigt sich, dass die Ordnung dieses Verfahrens gleich der des entsprechenden impliziten Adams-Moulton-Verfahrens ist (vergl. Abschnitt 6.2).

Beispiel 6.4

(a) Wir benutzen zwei einfache ESV, die zugleich MSV sind.

Hier ist eine Anlaufrechnung nicht notwendig.

Prädiktor P : PZV, Adams-Bashforth(1)-Verfahren, Fehler $\mathcal{O}(h)$

$$\eta_{i+1}^{(0)} = \eta_i + h f(t_i, \eta_i)$$

Korrektor K : implizites (2,2)-RKV (vergl. HV), Adams-Moulton(2)-V., Fehler $\mathcal{O}(h^2)$

$$\eta_{i+1}^{(m+1)} = \eta_i + \frac{h}{2} [f(t_{i+1}, \eta_{i+1}^{(m)}) + f(t_i, \eta_i)], \quad m = 0, 1, \dots, M-1$$

Mit mindestens $M = 1$, also einen Korrektorschritt hat dieses PCV die Fehlerordnung $\mathcal{O}(h^2)$.

⁷*praedicare* (lat.) = vorhersagen, *corrigere* (lat.) = verbessern

(b) Für 3-stufige MSV der Fehlerordnung $\mathcal{O}(h^3)$ notieren wir einen Schritt des PCV.

P: Man bestimme den Prädiktorwert $\eta_{i+1}^{(P)}$ durch das Adams-Bashforth(3)-Verfahren

$$\eta_{i+1}^{(P)} = \eta_i + \frac{h}{12} [23f(t_i, \eta_i) - 16f(t_{i-1}, \eta_{i-1}) + 5f(t_{i-2}, \eta_{i-2})].$$

K: Man berechne η_{i+1} aus der Gleichung für das Adams-Moulton(4)-Verfahren durch

$$\eta_{i+1} = \eta_i + \frac{h}{24} [9f(t_{i+1}, \eta_{i+1}^{(P)}) + 19f(t_i, \eta_i) - 5f(t_{i-1}, \eta_{i-1}) + f(t_{i-2}, \eta_{i-2})].$$

Man macht in K nur eine Iteration. Der Gesamtfehler ist von der Ordnung $\mathcal{O}(h^4)$.

Was die Lösungsgenauigkeit und Stabilitätsgebiete (vergl. Kapitel 7) der Adams-Verfahren betrifft, hat unter den Varianten

- n -stufiges Adams-Bashforth(n)-Verfahren n -ter Ordnung,
- Adams-Bashforth(n)-Prädiktor und $(n-1)$ -stufiger Adams-Moulton(n)-Korrektor,
- Adams-Bashforth(n)-Prädiktor und n -stufiger Adams-Moulton($n+1$)-Korrektor
(beide haben dabei die Stufenanzahl n),

das dritte Verfahren i. Allg. die besten Eigenschaften.

(c) Wir betrachten das PCV aus Teil (a) für das AWP

$$x'(t) = t - x(t), \quad x(0) = x_0 = 1, \quad t \in [0, 1] = [t_0, T]. \quad (6.25)$$

Zu Vergleichen nutzen wir die exakte Lösung $x(t) = 2e^{-t} + t - 1$ und speziell $x(1) = \frac{2}{e}$. Dabei variieren wir die Anzahl der Iterationen im Korrektor gemäß $M = 1, 2, \dots, 5$.

Wir wählen eine grobe Integrationsrittweite h , um auch die Veränderungen im Korrektorprozess zu verdeutlichen.

MATLAB-File mit tabellarischer Ausgabe und Grafik

```
% pcv1.m
% Praediktor: 1-Schritt Adams-Bashforth -- Ordnung 1, PZV
% Korrektor: 1-Schritt Adams-Moulton -- Ordnung 2, implizites HV

% Ohne Anlaufrechnung
% f10.m, Dgl x'=t-x, x(0)=1
% f10ex.m, exakte Loesung x(t)=2*exp(-t)+t-1

clear all
clc
diary pcv1.pro
diary on
disp('PC-Verfahren mit Adams-Bashforth(1) + Adams-Moulton(2)')
disp(' ')

% Integrationsintervall und Anfangswert
t0 = 0;
T = 1;
x0 = 1;
```

```

% Schrittweite
n = 5;
h = (T-t0)/n;
% exakte Loesung
for i=0:n
    xex(i+1) = f10ex(t0+h*i);
end;
t = (t0:h:T)';          % t = linspace(t0,T,n+1)'
xex = f10ex(t);
tt = (t0:h/10:T)';    % Feinunterteilung fuer Grafik
xexg = f10ex(tt);
xex1 = f10ex(1);

% numerische Loesung mit PC-Verfahren
eta = zeros(n+1,1);
eta(1) = x0;
% Korrektorschritte, Anzahl = M
for M = 1:5
    disp(['M=' num2str(M)]);
    % disp(['M=',num2str(1)]); disp(strcat('M=',num2str(1)));
    for i = 1:n
        k1 = f10(t(i),eta(i));
        eta0 = eta(i)+h*k1;
        for k = 0:M-1
            k2 = f10(t(i+1),eta0);
            eta1 = eta(i)+h/2*(k2+k1);
            eta0 = eta1;
        end;
        eta(i+1) = eta0;
    end;
end;

format short e
err1 = eta(n+1)-xex1    % Fehler bei t=1
format long
eta
xex
format
switch M
    case 1, etaM1 = eta;
    case 2, etaM2 = eta;
    case 3, etaM3 = eta;
    case 4, etaM4 = eta;
    case 5, etaM5 = eta;
end
% Plots
switch M
    case 1
        plot(t,etaM1,'k:',tt,xexg,'k-');
        title('PC-Loesungen mit AB(1)+AM(2) und exakte Loesung x(t)');
        xlabel('t');
        axis([0 1 0.5 1]);
        text(0.4,0.71,'x(t)');
        text(0.5,0.75,'PC, M=1');
        hold on
    case 2
        plot(t,etaM2,'b-');
        text(0.52,0.68,'PC, M=2');
    case 3
        plot(t,etaM3,'r-');
        text(0.70,0.68,'PC, M=3,');

```

```

case 4
  plot(t,etaM4,'g-');
  text(0.70,0.68,'PC, M= 4,');
case 5
  plot(t,etaM5,'m-');
  text(0.70,0.68,'PC, M= 5');
end
end
print pcv1.ps -dpsc
hold off

disp(' '); disp('Tabelle')
fprintf('t(i) xex(i) etaM1(i) etaM2(i) etaM3(i) etaM4(i) etaM5(i)\n');
fprintf('-----\n');
for i=1:n+1
  fprintf('%4.2f %8.6f %8.6f %8.6f %8.6f %8.6f %8.6f\n',...
    t(i),xex(i),etaM1(i),etaM2(i),etaM3(i),etaM4(i),etaM5(i));
end;
fprintf('-----\n');
fprintf('err bei t=1 %8.6f %9.6f %9.6f %9.6f %9.6f %9.6f\n',...
etaM1(n+1)-xex1,etaM2(n+1)-xex1,etaM3(n+1)-xex1,etaM4(n+1)-xex1,etaM5(i)-xex1);
diary off

```

Ausschnitte aus dem Ergebnisprotokoll bei $h = 0.2$

PC-Verfahren mit Adams-Bashforth(1) + Adams-Moulton(2)

```

M=1
err1 = 5.7208e-003
eta =
  1.000000000000000
  0.840000000000000
  0.744800000000000
  0.702736000000000
  0.704243520000000
  0.741479686400000
xex =
  1.000000000000000
  0.83746150615596
  0.74064009207128
  0.69762327218805
  0.69865792823444
  0.73575888234288
M=2
err1 = -3.2776e-003
eta =
  1.000000000000000
  0.836000000000000
  0.738248000000000
  0.694686864000000
  0.69545385475200
  0.73248125318714
M=3
err1 = -2.3817e-003
eta =
  1.000000000000000
  0.836400000000000
  0.738902480000000
  0.69549000913600
  0.69632992547508
  0.73337714502371

```

Tabelle						
t(i)	xex(i)	etaM1(i)	etaM2(i)	etaM3(i)	etaM4(i)	etaM5(i)
0.00	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
0.20	0.837462	0.840000	0.836000	0.836400	0.836360	0.836364
0.40	0.740640	0.744800	0.738248	0.738902	0.738837	0.738844
0.60	0.697623	0.702736	0.694687	0.695490	0.695410	0.695418
0.80	0.698658	0.704244	0.695454	0.696330	0.696242	0.696251
1.00	0.735759	0.741480	0.732481	0.733377	0.733288	0.733296
err bei t=1		0.005721	-0.003278	-0.002382	-0.002471	-0.002462

Man sieht, dass 2–3 Korrektorschritte ausreichend sind.

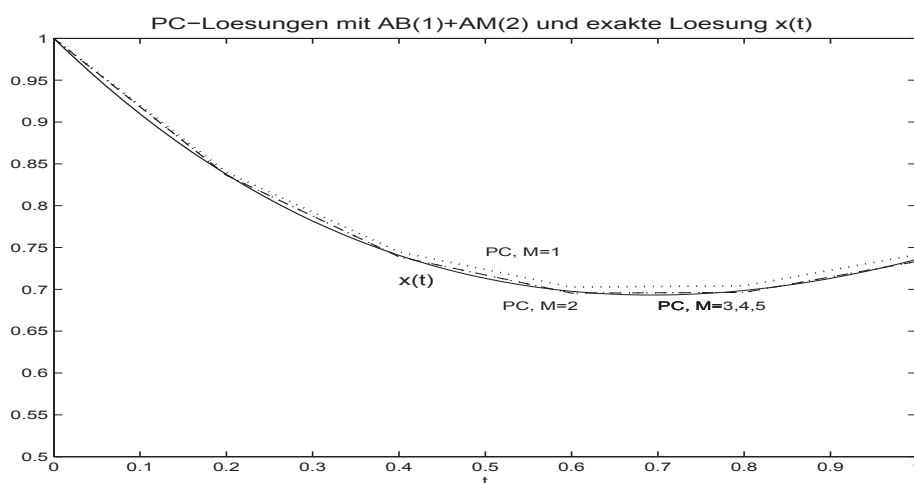


Abb. 6.4 Datei *pcv1.ps*,

Lösung zum AWP $x' = 2e^{-t} + t - 1$, $x(0) = 1$, $t \in [0, 1]$, sowie

Näherungslösungen aus dem PCV mit M Korrektorschritten, $M = 1(1)5$

Ausschnitte aus dem Ergebnisprotokoll bei $h = 0.025$

Tabelle						
t(i)	xex(i)	etaM1(i)	etaM2(i)	etaM3(i)	etaM4(i)	etaM5(i)
0.00	1.000000	1.000000	1.00000000	1.00000000	1.00000000	1.00000000
0.10	0.909675	0.909694	0.90966505	0.90966541	0.90966541	0.90966541
0.20	0.837462	0.837496	0.83744380	0.83744446	0.83744445	0.83744445
0.30	0.781636	0.781684	0.78161241	0.78161330	0.78161329	0.78161329
0.40	0.740640	0.740697	0.74061110	0.74061217	0.74061216	0.74061216
0.50	0.713061	0.713126	0.71302853	0.71302974	0.71302973	0.71302973
0.60	0.697623	0.697693	0.69758767	0.69758899	0.69758897	0.69758897
0.70	0.693171	0.693244	0.69313302	0.69313441	0.69313440	0.69313440
0.80	0.698658	0.698734	0.69861906	0.69862050	0.69862048	0.69862048
0.90	0.713139	0.713217	0.71309975	0.71310122	0.71310120	0.71310120
1.00	0.735759	0.735837	0.73571910	0.73572058	0.73572056	0.73572056
err bei t=1		0.000078	-0.00003978	-0.00003831	-0.00003832	-0.00003832

Es sollen nun in Anlehnung an die Betrachtungen in den Abschnitten 5.1 und 5.2 die Konsistenzbedingungen für PCV auf der Basis von r -Schritt-Verfahren präzisiert werden.

MSV, explizit

$$\eta_{i+r} + \sum_{k=0}^{r-1} a_k \eta_{i+k} = h \sum_{k=0}^{r-1} b_k f_{i+k}, \quad i \geq 0, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben,}$$

MSV, implizit

$$\eta_{i+r} + \sum_{k=0}^{r-1} \alpha_k \eta_{i+k} = h \sum_{k=0}^r \beta_k f_{i+k}, \quad a_r = 1,$$

PCV

$$\begin{aligned} \eta_{i+r}^{(P)} &= - \sum_{k=0}^{r-1} a_k \eta_{i+k} + h \sum_{k=0}^{r-1} b_k f_{i+k}, \\ \eta_{i+r} &= - \sum_{k=0}^{r-1} \alpha_k \eta_{i+k} + h \beta_r f(t_{i+r}, \eta_{i+r}^{(P)}) + h \sum_{k=0}^{r-1} \beta_k f_{i+k}. \end{aligned}$$

Konsistenzordnung

Sei $z(t) = \Phi^{t,t_i} \eta_i$ sowie $z_j = z(t_j)$.

Man definiert

$$z_{i+r}^{(P)} = - \sum_{k=0}^{r-1} a_k z_{i+k} + h \sum_{k=0}^{r-1} b_k z'_{i+k}$$

und

$$f(t_{i+r}, z_{i+r}^{(P)}) = \left(z_{i+r}^{(P)} \right)' = - \sum_{k=0}^{r-1} a_k z'_{i+k} + h \sum_{k=0}^{r-1} b_k z''_{i+k}.$$

Eingesetzt folgt als lokaler Diskretisierungsfehler (vergl. Definition 6.2)

$$\begin{aligned} \tau(t_i, \eta_i, h) &= \\ &= \frac{1}{h} (z_{i+r} + \alpha_{r-1} z_{i+r-1} + \dots + \alpha_0 z_i) - \left(\beta_r \left(z_{i+r}^{(P)} \right)' + \beta_{r-1} f_{i+r-1} + \dots + \beta_0 f_i \right) \\ &= \frac{1}{h} \left(z_{i+r} + \sum_{k=0}^{r-1} \alpha_k z_{i+k} \right) - \left[\beta_r \left(- \sum_{k=0}^{r-1} a_k z'_{i+k} + h \sum_{k=0}^{r-1} b_k z''_{i+k} \right) + \sum_{k=0}^{r-1} \beta_k z'_{i+k} \right] \\ &= \frac{1}{h} \left(z_{i+r} + \sum_{k=0}^{r-1} \alpha_k z_{i+k} \right) - \left(\sum_{k=0}^{r-1} (\beta_k - \beta_r a_k) z'_{i+k} + h \beta_r \sum_{k=0}^{r-1} b_k z''_{i+k} \right), \end{aligned}$$

der für die entsprechenden Parameter a_k , b_k , α_k , β_k auszuwerten ist.

Weitere PCV findet der Leser in der Übungsserie 6, Aufgabe 10.

6.2 Die Ordnung linearer Mehrschrittverfahren

Ähnlich wie in der Definition 5.1 für ESV definieren wir die Ordnung von MSV mit Hilfe des lokalen Diskretisierungsfehlers, welcher sich ergibt, wenn wir die exakte Lösung der gDGL in die Gleichung für das MSV einsetzen. Im Folgenden werde wieder für den gegebenen Punkt (t_i, η_i) mit

$$z(t) = \Phi^{t, t_i} \eta_i$$

die exakte Lösung der gDGL durch diesen Punkt (AB) bezeichnet.

Definition 6.2 (a) *Der lokale Diskretisierungsfehler für das MSV (6.8) ist definiert durch*

$$\begin{aligned} \tau(t_i, \eta_i, h) &= \frac{1}{h} [z(t_{i+r}) + a_{r-1}z(t_{i+r-1}) + \dots + a_0z(t_i)] \\ &\quad - [b_r f(t_{i+r}, z(t_{i+r})) + \dots + b_0 f(t_i, z(t_i))]. \end{aligned} \quad (6.26)$$

(b) *Das MSV ist ein Verfahren der Ordnung $p > 0$, wenn*

$$\tau(t_i, \eta_i, h) = \mathcal{O}(h^p) \quad \text{bzw.} \quad |\tau(t_i, \eta_i, h)| \leq Ch^p, \quad C \geq 0. \quad (6.27)$$

p bezeichnet erneut die Konsistenzordnung.

Die Berechnung der Ordnung erfolgt wieder mit Hilfe von Taylor-Entwicklungen um (t_i, η_i) . Aus

$$z(t_{i+k}) = z(t_i) + kh z'(t_i) + \frac{(kh)^2}{2!} z''(t_i) + \frac{(kh)^3}{3!} z'''(t_i) + \dots$$

und

$$f(t_{i+k}, z(t_{i+k})) = z'(t_{i+k}) = z'(t_i) + kh z''(t_i) + \frac{(kh)^2}{2!} z'''(t_i) + \frac{(kh)^3}{3!} z^{(4)}(t_i) + \dots$$

folgt die Entwicklung für den lokalen Diskretisierungsfehler

$$\tau(t_i, \eta_i, h) = \frac{1}{h} c_0 z(t_i) + c_1 z'(t_i) + c_2 h z''(t_i) + \dots = \sum_{k=0}^{\infty} c_k h^{k-1} z^{(k)}(t_i) \quad (6.28)$$

mit

$$c_0 = a_0 + a_1 + \dots + a_r, \quad (6.29)$$

$$c_1 = (a_1 + 2a_2 + \dots + r a_r) - (b_0 + b_1 + \dots + b_r) \quad (6.30)$$

sowie für $k \geq 2$

$$c_k = \frac{1}{k!} (a_1 + 2^k a_2 + \dots + r^k a_r) - \frac{1}{(k-1)!} (b_1 + 2^{k-1} b_2 + \dots + r^{k-1} b_r). \quad (6.31)$$

Hierbei ist $a_r = 1$.

Wir fassen zusammen.

Satz 6.1 *Das lineare MSV (6.8) hat genau dann (mindestens) die Konsistenzordnung p ($p \geq 1$), wenn mit den Definitionen (6.29), (6.30) und (6.31) für c_k gilt, dass*

$$c_0 = c_1 = \dots = c_p = 0. \tag{6.32}$$

Beim Entwerfen von MSV ist nun eines der Ziele, die Koeffizienten a_k, b_k so zu bestimmen, dass möglichst viele der Größen c_k verschwinden.

Beispiel 6.5

(a) Ein explizites 3-Schritt-Verfahren der Form

$$\eta_{i+1} + a_1\eta_{i-1} = h [b_2f(t_i, \eta_i) + b_1f(t_{i-1}, \eta_{i-1}) + b_0f(t_{i-2}, \eta_{i-2})] \tag{6.33}$$

mit maximaler Ordnung soll ermittelt werden. Man beachte die Indexverschiebung bei der Notation der η_j und, dass $a_3 = 1$ und $a_0 = a_2 = b_3 = 0$ sind. Die maximale Ordnung 3 erhält man, wenn a_1, b_0, b_1 und b_2 so bestimmt werden, dass

$$\begin{aligned} 0 &= c_0 = a_1 + 1, \\ 0 &= c_1 = a_1 + 3 - b_0 - b_1 - b_2, \\ 0 &= 2c_2 = a_1 + 9 - 2b_1 - 4b_2, \\ 0 &= 6c_3 = a_1 + 27 - 3b_1 - 12b_2 \end{aligned}$$

gelten. Die Lösung

$$a_1 = -1, \quad b_2 = \frac{7}{3}, \quad b_1 = -\frac{2}{3}, \quad b_0 = \frac{1}{3}$$

führt auf das Verfahren dritter Ordnung

$$\eta_{i+1} = \eta_{i-1} + \frac{h}{3}[7f(t_i, \eta_i) - 2f(t_{i-1}, \eta_{i-1}) + f(t_{i-2}, \eta_{i-2})]. \tag{6.34}$$

Das MSV gehört zu den expliziten Adams-Nyström-Formeln oder kurz Nyström-Formeln n -ter Ordnung. Dafür eignet sich besonders die spezielle Darstellung (6.9)

$$\eta_{i+1} = \eta_{i-1} + h \sum_{j=0}^r \beta_j f_{i-j}, \quad i = s = \max(r, 1), s + 1, \dots, r, \quad \eta_0, \eta_1, \dots, \eta_s \text{ geg.} \tag{6.35}$$

r	Ordnung $n = s + 1$ Stufen	K	j			
			0	1	2	3
0	2	1	2			
2	3	3	7	-2	1	
3	4	3	8	-5	4	-1

Tab. 6.5
Koeffizienten der n -stufigen
Adams-Nyström(n)-Verfahren (6.35)
(AN(n)) der Ordnung $n = s + 1$,
 $s = \max(r, 1)$, $r = 0, 2, 3$

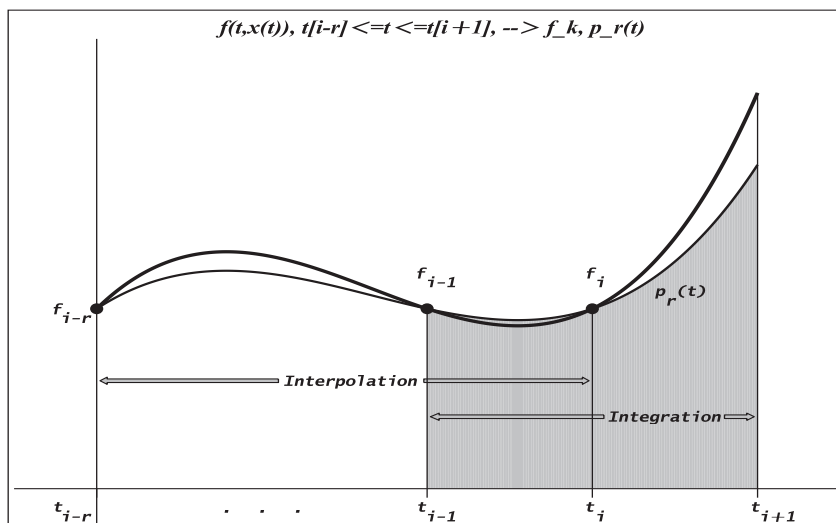


Abb. 6.5 Datei *msv_schritt1_04.ps*,

Interpolation und Integration beim MSV von Adams-Nyström

(b) Zur Bestimmung einer impliziten 3-Schritt-Methode der Form

$$\eta_{i+1} + a_1 \eta_{i-1} = h [b_3 f(t_{i+1}, \eta_{i+1}) + b_2 f(t_i, \eta_i) + b_1 f(t_{i-1}, \eta_{i-1}) + b_0 f(t_{i-2}, \eta_{i-2})] \quad (6.36)$$

mit maximaler Ordnung ist das Gleichungssystem

$$\begin{aligned} 0 &= c_0 = a_1 + 1, \\ 0 &= c_1 = a_1 + 3 - b_0 - b_1 - b_2 - b_3, \\ 0 &= 2c_2 = a_1 + 9 - 2b_1 - 4b_2 - 6b_3, \\ 0 &= 6c_3 = a_1 + 27 - 3b_1 - 12b_2 - 27b_3, \\ 0 &= 24c_4 = a_1 + 81 - 4b_1 - 32b_2 - 108b_3 \end{aligned}$$

zu lösen. Das Ergebnis lautet

$$a_1 = -1, \quad b_3 = \frac{1}{3}, \quad b_2 = \frac{4}{3}, \quad b_1 = \frac{1}{3}, \quad b_0 = 0.$$

Das hieraus resultierende Verfahren hat die Ordnung 4 und ist – entgegen dem ursprünglichen Ansatz – wegen $b_0 = 0$ ein implizites 2-Schritt-Verfahren.

Es gehört zur Gruppe der impliziten s -stufigen Milne-Simpson-Formeln, auch als Adams-Störmer-Formeln bezeichnet. In seiner Notation

$$\frac{1}{2h}(\eta_{i+1} - \eta_{i-1}) = \frac{1}{6}f_{i+1} + \frac{4}{6}f_i + \frac{1}{6}f_{i-1} \quad (6.37)$$

erkennt man leicht die Ähnlichkeit zur Quadraturformel von Simpson.

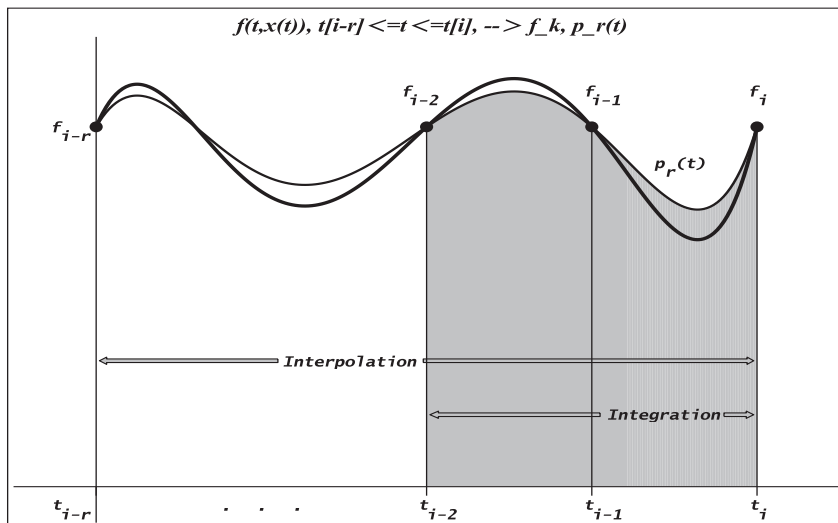


Abb. 6.6 Datei *msv_schritt1_05.ps*,
 Interpolation und Integration beim MSV von Milne-Simpson

Einige der impliziten s -stufigen Milne-Simpson-Formeln sollen tabellarisch zusammengefasst werden. Dafür nehmen wir die spezielle Darstellung (6.9)

$$\eta_i = \eta_{i-2} + h \sum_{j=0}^r \beta_j f_{i-j}, \quad i = s = \max(r, 2), s + 1, \dots, r, \quad \eta_0, \eta_1, \dots, \eta_{s-1} \text{ geg.} \quad (6.38)$$

r	Stufen s	Ordnung n	j			
			0	1	2	3
			β_j			
0	2	1	2			
1	2	1	$b \neq 0$	$2 - b$		
2	2	1	$b \neq 0$	$2 - b - \hat{b}$	\hat{b}	
2	2	2	$b \neq 0$	$2 - 2b$	b	
2	2	4	$b = \frac{1}{3}$	$\frac{4}{3}$	$\frac{1}{3}$	
3	3	3	$b \neq 0$	$-3b + \frac{7}{3}$	$3b - \frac{2}{3}$	$\frac{1}{3} - b$
3	$3 \searrow 2$	4	$b = \frac{1}{3}$	$\frac{4}{3}$	$\frac{1}{3}$	0

Tab. 6.6 Koeffizienten ausgewählter s -stufiger Milne-Simpson(n)-Verfahren (6.38) (MS(n)) der Ordnung $n = 1, 2, 3, 4$, evtl. mit Parameter

Damit sieht man die zentrale Rolle des 2-stufigen MS(4).

In der Übungsserie 6, Aufgabe 1, sind in einer Tabelle weitere implizite Milne-Simpson-Verfahren höherer Ordnung notiert.

(c) In der Vorgehensweise von (b) erhält man die Erkenntnis, dass aus dem Ansatz

$$a_2\eta_{i+1} + a_1\eta_i + a_0\eta_{i-1} = h [b_2f(t_{i+1}, \eta_{i+1}) + b_1f(t_i, \eta_i) + b_0f(t_{i-1}, \eta_{i-1})], \quad a_2 = 1, \quad (6.39)$$

ein einziges 2-Schritt-Verfahren der Ordnung 4 folgt. Es ist darüber hinaus implizit.

Das zu lösende Gleichungssystem lautet

$$\begin{aligned} 0 &= c_0 = a_0 + a_1 + a_2 = a_0 + a_1 + 1, \\ 0 &= c_1 = a_1 + 2a_2 - (b_0 + b_1 + b_2) = a_1 + 2 - b_0 - b_1 - b_2, \\ 0 &= 2c_2 = a_1 + 2^2a_2 - 2(b_1 + 2^1b_2) = a_1 + 4 - 2b_1 - 4b_2, \\ 0 &= 6c_3 = a_1 + 2^3a_2 - 3(b_1 + 2^2b_2) = a_1 + 8 - 3b_1 - 12b_2, \\ 0 &= 24c_4 = a_1 + 2^4a_2 - 4(b_1 + 2^3b_2) = a_1 + 16 - 4b_1 - 32b_2. \end{aligned}$$

Die Parameter sind

$$a_1 = 0, \quad a_0 = -1, \quad b_2 = \frac{1}{3}, \quad b_1 = \frac{4}{3}, \quad b_0 = \frac{1}{3}.$$

Als MSV ergibt sich die schon notierte Formel (6.37).

Man prüft nach, dass für die nächste Ordnungsbedingung

$$5!c_5 = a_1 + 2^5a_2 - 5(b_1 + 2^4b_2) = 32 - 4\frac{4}{3}b_1 - 80\frac{1}{3} = -\frac{4}{3}, \quad c_5 = -\frac{1}{90} \neq 0$$

gilt und damit diese nicht erfüllt ist.

(d) Das implizite 3-Schritt-Verfahren maximaler Ordnung mit dem Ansatz

$$\eta_{i+1} + a_2\eta_i + a_1\eta_{i-1} + a_0\eta_{i-2} = hb_3f_{i+1} \quad (6.40)$$

erhält man durch das Gleichungssystem

$$\begin{aligned} 0 &= c_0 = a_0 + a_1 + a_2 + 1, \\ 0 &= c_1 = a_1 + 2a_2 + 3 - b_3, \\ 0 &= 2c_2 = a_1 + 4a_2 + 9 - 6b_3, \\ 0 &= 6c_3 = a_1 + 8a_2 + 27 - 27b_3. \end{aligned}$$

Seine Lösung ist

$$a_2 = -\frac{18}{11}, \quad a_1 = \frac{9}{11}, \quad a_0 = -\frac{2}{11}, \quad b_3 = \frac{6}{11},$$

und das MSV (bei Multiplikation aller Koeffizienten mit 11) ist gegeben durch

$$11\eta_{i+1} - 18\eta_i + 9\eta_{i-1} - 2\eta_{i-2} = 6hf(t_{i+1}, \eta_{i+1}). \quad (6.41)$$

Das Verfahren gehört in die Klasse der BDF-Methoden, da die linke Seite eine Rückwärtsapproximation der ersten Ableitung an der Stelle t_{i+1} darstellt (vergl. Abschnitt 3.3.4).

Ersetzen wir η_{i+k} durch $z(t_{i+k}) = \Phi^{t_{i+k}, t_{i+1}} \eta_{i+1}$, so folgt

$$\begin{aligned} \frac{11}{6}z(t_{i+1}) - 3z(t_i) + \frac{3}{2}z(t_{i-1}) - \frac{1}{3}z(t_{i-2}) &= \\ &= \left(\frac{11}{6} - 3 + \frac{3}{2} - \frac{1}{3}\right) z(t_{i+1}) + \left(3 - 2 \cdot \frac{3}{2} + 3 \cdot \frac{1}{3}\right) h z'(t_{i+1}) + \mathcal{O}(h^2) \\ &= h z'(t_{i+1}) + \mathcal{O}(h^2). \end{aligned}$$

Die BDF-Methoden haben wir schon kennengelernt als besondere implizite Form (6.10)

$$\sum_{j=0}^r a_j \eta_{i+j} = h b_r f_{i+r}, \quad a_r, b_r \neq 0, \quad i \geq 0, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben.}$$

Mit einer Indexverschiebung bei der Notation der η_j notieren wir die modifizierten Koeffizienten (ganze Zahlen) der impliziten n -stufigen BDF(n)-Verfahren n -ter Ordnung

$$\alpha_r \eta_{i+1} + \alpha_{r-1} \eta_i + \dots + \alpha_0 \eta_{i-r+1} = h \beta_r f_{i+1}, \quad i = r-1, r, \dots, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ geg.} \quad (6.42)$$

r	Ordnung $n = r$ Stufen	β_r	j							
			r	$r-1$	$r-2$	$r-3$	$r-4$	$r-5$	$r-6$	α_j
1	1	1	1	-1						
2	2	2	3	-4	1					
3	3	6	11	-18	9	-2				
4	4	12	25	-48	36	-16	3			
5	5	60	137	-300	300	-200	75	-12		
6	6	60	147	-360	450	-400	225	-72	10	

Tab. 6.7 Koeffizienten der n -stufigen BDF(n)-Verfahren, $n = 1, 2, \dots, 6$

Satz 6.2 Ordnung von MSV

Das n -Schritt-Adams-Bashforth-Verfahren hat die Ordnung n .

Das n -Schritt-Adams-Moulton-Verfahren hat die Ordnung $n+1$ (außer dem Fall $r = 0$, $s = n = 1$).

Das zugehörige PCV, zusammengesetzt aus dem n -Schritt-Adams-Bashforth-Prädiktor und dem n -Schritt-Adams-Moulton-Korrektor, hat dann die Ordnung $n+1$.

Übung 6.3 (a) Zeigen Sie, dass das 3-Schritt-Adams-Bashforth-Verfahren die Ordnung 3 sowie das 3-Schritt-Adams-Moulton-Verfahren die Ordnung 4 haben.

(b) Zeigen Sie durch Taylor-Reihenentwicklung, dass das PCV aus Beispiel 6.4 (b) die Ordnung 4 hat. Vergleichen Sie den Rechenaufwand für dieses Verfahren mit dem für das Adams-Moulton-Verfahren derselben Ordnung.

Die Ergebnisse der Beispiele 6.4 und 6.5 lassen sich verallgemeinern.

6.3 Homogene lineare Differenzgleichungen

Lineare Differenzgleichungen

$$\sum_{k=0}^r a_k \eta_{i+k} = c_i, \quad (6.43)$$

wie sie bei der Formulierung von MSV entstehen (vergl. Definition 6.1), entwickeln eine "Eigendynamik". Dies ist bereits der Fall für $c_i = 0$ entsprechend der gDGl $x' = 0$.

Diese Differenzgleichungen sind der Inhalt der folgenden Untersuchungen.

Beispiel 6.6 (a) Die Formel

$$\eta_{i+1} + 4\eta_i - 5\eta_{i-1} = h(4f_i + 2f_{i-1}) \quad (6.44)$$

beschreibt ein explizites 2-Schritt-Verfahren. Wir bestimmen mittels (6.26) die Ordnung.

$$\begin{aligned} \tau(t_{i-1}, \eta_{i-1}, h) &= \frac{1}{h}[z(t_{i+1}) + 4z(t_i) - 5z(t_{i-1})] - [4f(t_i, z(t_i)) + 2f(t_{i-1}, z(t_{i-1}))] \\ &= \frac{1}{h}[z_{i-1} + 2hz'_{i-1} + \frac{(2h)^2}{2!}z''_{i-1} + \frac{(2h)^3}{3!}z'''_{i-1} + \frac{(2h)^4}{4!}z''''_{i-1} + \mathcal{O}(h^5)] \\ &\quad + 4z_{i-1} + 4hz'_{i-1} + \frac{4h^2}{2!}z''_{i-1} + \frac{4h^3}{3!}z'''_{i-1} + \frac{4h^4}{4!}z''''_{i-1} + \mathcal{O}(h^5) - 5z_{i-1}] \\ &\quad - 4(z'_{i-1} + hz''_{i-1} + \frac{h^2}{2!}z'''_{i-1} + \frac{h^3}{3!}z''''_{i-1} + \mathcal{O}(h^4)) - 2z'_{i-1} \\ &= \frac{1}{h}[6hz'_{i-1} + 4h^2z''_{i-1} + 2h^3z'''_{i-1} + \frac{5h^4}{6}z''''_{i-1} + \mathcal{O}(h^5)] \\ &\quad - 6z'_{i-1} - 4hz''_{i-1} + 2h^2z'''_{i-1} - \frac{4h^4}{6}z''''_{i-1} + \mathcal{O}(h^4) \\ &= \frac{h^3}{6}z''''_{i-1} + \mathcal{O}(h^4) \\ &= \mathcal{O}(h^3). \end{aligned}$$

Wir wenden das MSV auf das einfache AWP $x'(t) = 0$, $x(t_0) = 1$, mit der exakten Lösung $x(t) \equiv 1$ an.

Die AB sei genau erfüllt, also ist $\eta_0 = 1$. Wir nehmen an, dass η_1 durch ein ESV mit einem gewissen Rundungsfehler berechnet wurde, also $\eta_1 = 1 + \delta h$. Für $\delta = 10^{-6}$ und $h = 0.05$ erhalten wir auf diese Weise mit der rekursiven Berechnungsvorschrift

$$\eta_{i+1} = -4\eta_i + 5\eta_{i-1}$$

die folgenden anfänglichen Werte η_i , $i = 0, 1, \dots$

eta(0)=	1		eta(11)=	1.406901049
eta(1)=	1.000000050		eta(12)=	-1.034505196
eta(2)=	0.999999800		eta(13)=	11.172526032
eta(3)=	1.000001050		eta(14)=	-49.862630108
eta(4)=	0.999994800		eta(15)=	255.313150589
eta(5)=	1.000026050		eta(16)=	-1270.565752893
eta(6)=	0.999869800		eta(17)=	6358.828764516
eta(7)=	1.000651050		eta(18)=	-31788.143822528
eta(8)=	0.996744800		eta(19)=	158946.719112688
eta(9)=	1.016276050		eta(20)=	-794727.595563392
eta(10)=	0.918619800			

Damit hat (6.44) offenbar ein oszillierendes, stark divergierendes Verhalten. Diese Instabilität wollen wir untersuchen. Zur Bestimmung der allgemeinen Lösung der Gleichung

$$\eta_{i+1} + 4\eta_i - 5\eta_{i-1} = 0 \quad (6.45)$$

mit den Startwerten $\eta_0 = c_0$ und $\eta_1 = c_1$ berechnen wir zunächst sein charakteristisches oder erzeugendes Polynom. Durch Einsetzen des Ansatzes $\eta_i = \xi^i$, $\xi \neq 0$, in die Differenzgleichung (6.45) erhält man

$$\rho(\xi) = \xi^2 + 4\xi - 5 = 0. \quad (6.46)$$

Seine beiden Nullstellen sind $\xi_1 = 1$ und $\xi_2 = -5$.

Man kann sich nun leicht überlegen, dass die allgemeine Lösung von (6.45) gegeben ist durch

$$\eta_i = \gamma_1 \xi_1^i + \gamma_2 \xi_2^i = \gamma_1 + \gamma_2 (-5)^i. \quad (6.47)$$

Die Größen $\gamma_{1,2}$ sind eindeutig bestimmt durch die Startwerte c_0 und c_1 . Es gilt unter Berücksichtigung der konkreten AB $\eta_0 = 1$, $\eta_1 = 1 + \delta h$ die Beziehung

$$\eta_i = \left(c_0 - \frac{c_0 - c_1}{6}\right) 1^i + \frac{c_0 - c_1}{6} (-5)^i = 1 + \frac{\delta h}{6} - \frac{\delta h}{6} (-5)^i. \quad (6.48)$$

Wenn man das MSV auf die gDGI $x' = 0$, $x(0) = 1$, anwendet und den Anfangswert $\eta_1 = 1$ nicht stört, dann ist $\eta_i \equiv 1$ und diese stimmen mit der Lösung $x(t) = 1$ überein. Jedoch werden sie bei leichter Störung $\eta_1 = 1 + \delta h$, $\delta \neq 0$, durch den Faktor $(-5)^i$ dramatisch verstärkt.

Offenbar wird das instabile Verhalten des MSV dadurch verursacht, dass $\rho(\xi)$ eine Nullstelle besitzt, deren Betrag größer als Eins ist.

(b) Betrachten wir die parameterabhängige homogene Differenzgleichung

$$\eta_{i+1} - 2c\eta_i + c^2\eta_{i-1} = 0, \quad c \in \mathbb{R}, \quad (6.49)$$

die bei $c = 1$ und den AB $\eta_0 = \eta_1 = 1$ die Folge der Werte $\eta_i = 1$ liefert.

Mit dem Ansatz $\eta_i = \xi^i$, $\xi \neq 0$, gelangt man zum charakteristischen Polynom

$$\rho(\xi) = \xi^2 - 2c\xi + c^2 = (\xi - c)^2 \quad (6.50)$$

und seiner doppelten Nullstelle $\xi_{1,2} = c$.

Die allgemeine Lösung von (6.50) ist gegeben durch

$$\eta_i = \gamma_1 \xi_1^i + \gamma_2 i \xi_2^i = \gamma_1 c^i + \gamma_2 i c^i. \quad (6.51)$$

Für Parameterwerte $|c| > 1$ liegt ein starkes Wachstum der η_i -Folge vor, für $|c| < 1$ tendieren die Werte $|\eta_i|$ gegen Null. Aber auch im Fall $|c| = 1$ besteht i. Allg. wegen des

Terms $i c^i$ die Gefahr eines oszillierenden, nicht konvergenten Verhaltens.

Nur wenn $|\xi_{1,2}| = 1$ und $\xi_1 \neq \xi_2$ sind, wie das für die Differenzgleichung $\eta_{i+1} - \eta_{i-1} = 0$ z.B. der Fall ist, wird die allgemeine Lösung $\eta_i = \gamma_1 \xi_1^i + \gamma_2 \xi_2^i$ konvergent sein.

Die Argumentationen im Beispiel 6.6 für die Lösung der homogenen Differenzgleichung lassen sich als Satz verallgemeinern.

Satz 6.3 *Hat das Polynom r -ten Grades*

$$\rho(\xi) = a_r \xi^r + a_{r-1} \xi^{r-1} + \dots + a_1 \xi + a_0 \quad (6.52)$$

die r paarweise verschiedenen Nullstellen $\xi_1, \xi_2, \dots, \xi_r$, so lässt sich jede Lösungsfolge $(\eta_i)_{i \in \mathbb{N}}$ der homogenen Differenzgleichung (auch homogene Rekursion genannt)

$$\sum_{s=0}^r a_s \eta_{i+s} = 0 \quad (6.53)$$

in der Form

$$\eta_i = \sum_{s=1}^r \gamma_s \xi_s^i \quad (6.54)$$

darstellen.

Die Koeffizienten γ_s sind eindeutig bestimmt durch die Startwerte $\eta_0, \eta_1, \dots, \eta_{r-1}$.

Bei doppelten oder i. Allg. mehrfachen Nullstellen ist die Lösungsdarstellung komplexer.

Satz 6.4 *Hat das Polynom r -ten Grades $\rho(\xi)$ (6.52) die m -fache reelle Nullstelle ξ , d.h.*

$$\rho(\xi) = \rho'(\xi) = \dots = \rho^{(m)}(\xi) = 0, \quad m \leq r,$$

dann gelten die folgenden Aussagen.

$$(1) \quad \eta_{i,1} = \xi^i, \quad \eta_{i,2} = D\xi^i = i \xi^{i-1}, \quad \eta_{i,3} = D^2\xi^i = i(i-1) \xi^{i-2}, \dots,$$

$$\eta_{i,m} = D^{m-1}\xi^i = i(i-1) \cdot \dots \cdot (i-m+2) \xi^{i-m+1},$$

sind spezielle Lösungen der homogenen Differenzgleichung (6.53).

(2) Auch die "skalierten" Werte

$$\tilde{\eta}_{i,1} = \eta_{i,1}, \quad \tilde{\eta}_{i,j} = \eta_{i,j} \xi^{j-1} = i(i-1) \cdot \dots \cdot (i-j+2) \xi^i, \quad j = 2, 3, \dots, m,$$

sind spezielle Lösungen von (6.53).

(3) Sind alle Nullstellen von $\rho(\xi)$ reell, dann ist die allgemeine Lösung von (6.53) eine Linearkombination der insgesamt r speziellen Lösungen nach (1).

Beweis.

(1) Betrachten wir die spezielle Lösung $\eta_{i,j} = D^{j-1}\xi^i$, $1 \leq j \leq m$.

Mit der Produktregel der Differentiation nach Leibniz

$$D^l(f \cdot g) = \sum_{s=0}^l \binom{l}{s} D^s f D^{l-s} g$$

gilt

$$\begin{aligned} a_0 \eta_{i,j} + a_1 \eta_{i+1,j} + \dots + a_r \eta_{i+r,j} &= D^{j-1}(a_0 \xi^i + a_1 \xi^{i+1} + \dots + a_r \xi^{i+r}) \\ &= D^{j-1}(\rho(\xi) \xi^i) \\ &= \underbrace{\rho(\xi)}_{=0} D^{j-1} \xi^i + \dots + \binom{j-1}{s} \underbrace{D^s \rho(\xi)}_{=0} D^{j-1-s} \xi^i + \dots + \underbrace{D^{j-1} \rho(\xi)}_{=0} \xi^i \\ &= 0. \end{aligned}$$

(2) Alle skalierten Größen $i(i-1) \cdot \dots \cdot (i-j+2) \xi^{i-j+1+k}$, $k \geq 0$, insbesondere bei $k = j-1$ genügen der Differenzgleichung (6.53).

(3) Es gibt insgesamt r spezielle Lösungen entsprechend dem Grad von $\rho(\xi)$.

Seien ξ_l , $l = 1, 2, \dots, k$, die reellen Nullstellen mit der entsprechenden Vielfachheit $m_l \geq 1$ und $\sum_{l=1}^k m_l = r$.

Jede Linearkombination in der Form

$$\eta_i = \sum_{l=1}^k \left[\gamma_{l,1} \xi_l^i + \sum_{s=2}^{m_l} \gamma_{l,s} i(i-1) \cdot \dots \cdot (i-s+2) \xi_l^{i-s+1} \right]$$

ist ebenfalls Lösung der homogenen Differenzgleichung.

Zu zeigen bleibt, dass solch eine Linearkombination, also die Koeffizienten $\gamma_{l,s}$, durch die Startwerte $\eta_0, \eta_1, \dots, \eta_{r-1}$ eindeutig festgelegt sind (siehe [104]). \square

Damit haben wir eine Fallunterscheidung.

- Falls ξ eine Nullstelle von $\rho(\xi)$ ist mit $|\xi| > 1$, wächst $\eta_i = \xi^i$ exponentiell.
- Falls ξ eine Nullstelle von $\rho(\xi)$ ist mit $|\xi| < 1$, fällt $\eta_i = \xi^i$ exponentiell.
- Falls ξ eine Nullstelle von $\rho(\xi)$ ist mit $|\xi| = 1$, sind zwei Fälle zu unterscheiden.
 - Einfache Nullstelle mit $\eta_i = \xi^i$, $|\eta_i| = 1$,
 - m -fache Nullstelle ($m \geq 2$) mit $\eta_{i,m} = i(i-1) \cdot \dots \cdot (i-m+2) \xi^{i-m+1}$,
 $|\eta_i| = i(i-1) \cdot \dots \cdot (i-m+2)$ wächst polynomial.

Die homogene Differenzgleichung (6.53) wird also vollständig durch die Nullstellen des charakteristischen Polynoms $\rho(\xi)$ (6.52) beschrieben.

Wichtig für unsere Zwecke ist es, dass die Lösungen der homogenen Gleichung (6.53) langsamer als linear bzgl. des Index anwachsen, dass also die Wachstumsbeschränkung

$$\lim_{i \rightarrow \infty} \frac{\eta_i}{i} = 0 \tag{6.55}$$

gilt. Hierzu haben wir das folgende wichtige Resultat.

Satz 6.5 Die Wachstumsbeschränkung (6.55) gilt genau dann für beliebige Startwerte $\eta_0, \eta_1, \dots, \eta_{r-1}$, wenn das folgende Stabilitäts- oder Wurzelkriterium erfüllt ist.

Stabilitätskriterium: Das Polynom $\rho(\xi)$ (6.52) besitzt nur Nullstellen ξ_s mit Betrag $|\xi_s| \leq 1$. Ist ξ_s eine Nullstelle mit Betrag $|\xi_s| = 1$, so ist ξ_s einfache Nullstelle.

Ein MSV mit dieser Eigenschaft heißt nullstabil (zero-stable).

Das MSV heißt stark stabil, wenn für alle Nullstellen ξ gilt $|\xi| < 1$, außer für $\xi = 1$, ansonsten schwach stabil.

Aufgrund der Test-DGL $x' = 0$ spricht man manchmal auch beim Stabilitätskriterium von der Nullstabilität oder, zu Ehren von G. Dahlquist⁸, von der D-Stabilität.

Die Ordnung eines A-stabilen MSV ist höchstens 2. Das ist die sogenannte zweite Ordnungsbarriere von Dahlquist.

Bei den folgenden Konsistenzuntersuchungen zu MSV wird als Grundforderung die Bedingung $\rho(1) = 0$ gefunden, sodass $\xi = 1$ immer eine Nullstelle von $\rho(\xi)$ ist.

Bei den beiden Verfahrensklassen von Adams-Bashforth und Adams-Moulton ist $a_r = 1$, $a_{r-1} = -1$, $a_{r-1} = \dots = a_0 = 0$. Damit sind $\xi = 1$ eine einfache Nullstelle und $\xi = 0$ eine $(r - 1)$ -fache Nullstelle.

Beispiel 6.7

(a) Das MSV des Beispiels 6.6 (a) erfüllt nicht das Stabilitätskriterium des Satzes 6.5, da mit $\xi_2 = -5$ eine Nullstelle mit Betrag größer als 1 vorliegt.

(b) Für das Beispiel 6.5 (a) ist $\rho(\xi)$ gegeben durch

$$\rho(\xi) = \xi^2 - 1 = (\xi + 1)(\xi - 1).$$

Die Nullstellen sind -1 und 1 . Das Stabilitätskriterium ist damit erfüllt.

(c) Für das BDF-Verfahren im Beispiel 6.5 (c) ist

$$\rho(\xi) = 11\xi^3 - 18\xi^2 + 9\xi - 2.$$

Die Nullstellen sind $\xi_1 = 1$ sowie $\xi_{2,3} = (7 \pm \sqrt{39}i)/22 = 0.318181 \pm 0.283863i$.

Das Stabilitätskriterium ist erfüllt. Übrigens gilt dies für alle BDF-Verfahren.

⁸Germund Dahlquist (1925-2005), schwedischer Mathematiker, Arbeitsgebiete waren numerische Mathematik, gDGL, Stabilitätsuntersuchungen

6.4 Konsistenz und Konvergenz von Mehrschrittverfahren

Gegeben sei das r -stufige lineare MSV

$$\sum_{k=0}^r a_k \eta_{i+k} = h \sum_{k=0}^r b_k f_{i+k}, \quad f_{i+k} = f(t_{i+k}, \eta_{i+k}), \quad a_r = 1, \quad r \geq 1. \quad (6.56)$$

Definition 6.3 *Das erste und das zweite charakteristische (erzeugende) Polynom des MSV (6.56) sind definiert durch*

$$\rho(\xi) = \sum_{k=0}^r a_k \xi^k, \quad \sigma(\xi) = \sum_{k=0}^r b_k \xi^k. \quad (6.57)$$

Das Polynom $\rho(\xi)$, das auch in der Formel (6.52) und im Stabilitätskriterium (Satz 6.5) erscheint, nimmt also eine zentrale Stellung ein. Es ist das wichtigere von beiden, da wegen $h \approx 0$ bzw. $h \rightarrow 0$ das Polynom $\sigma(\xi)$ sozusagen entfallen kann.

Definition 6.4 (a) *Das MSV (6.56) heißt konsistent, falls es mindestens die Ordnung 1 hat, falls also nach Satz 6.1*

$$\begin{aligned} 0 &= c_0 = a_0 + a_1 + \dots + a_r, \\ 0 &= c_1 = a_1 + 2a_2 + \dots + ra_r - (b_0 + b_1 + \dots + b_r) \end{aligned} \quad (6.58)$$

gelten. Man überlegt sich leicht, dass diese Konsistenzbedingung äquivalent ist zur

Konsistenzbedingung: $\rho(1) = 0, \quad \rho'(1) - \sigma(1) = 0.$

(6.59)

Ein weitere dazu äquivalente Aussage ist, dass das MSV (6.56) exakt ist für die AWP

$$\begin{aligned} x' &= 0, \quad x(0) = c, \\ x' &= 1, \quad x(0) = c, \end{aligned} \quad (6.60)$$

also mit den zugehörigen Lösungstrajektorien $x(t) = c$ bzw. $x(t) = t + c$ an den Stellen $t_i = ih$ bei exakten AB übereinstimmt.

(b) Ähnlich wie bei ESV (vergl. Abschnitt 5.3) wird die Konvergenz von MSV definiert. Hierzu sei $x(t)$ die exakte Lösung des AWP im Intervall $[t_0, T]$ und η_i sei die Lösung des MSV zur Schrittweite h . Das MSV heißt konvergent, wenn für beliebige $t > t_0$ und $N = 1, 2, 3, \dots$ zu den Schrittweiten $h = (t - t_0)/N$ gilt, dass

$$\lim_{N \rightarrow \infty} \eta_N = x(t). \quad (6.61)$$

Das heißt, dass der globale Diskretisierungsfehler oder der Fehler der Näherungslösung

$$e(t, h) = \eta_N - x(t)$$

mit fallender Schrittweite h bzw. wachsender Anzahl N der Teilschritte für beliebiges $t \in [t_0, T]$ gegen Null tendiert.

Die Konsistenzbedingung als Genauigkeitsordnung 1 ist in (6.59) mit den charakteristischen Polynomen (6.57) ausgedrückt worden. Zugleich haben wir für den lokalen Diskretisierungsfehler $\tau(t_i, \eta_i, h)$ (6.26) mittels Taylor-Entwicklung die allgemeinen Bedingungen für die Konsistenzordnung $p \geq 1$ erhalten.

Das Fehlermaß $\tau(t_i, \eta_i, h)$ ergibt sich also durch Einsetzen der Lösung $z(t) = \Phi^{t, t_i} \eta_i$ in die Differenzgleichung (6.8).

Im Sonderfall exakter Daten

$$\eta_i = z(t_i), \eta_{i+1} = z(t_{i+1}), \dots, \eta_{i+r-1} = z(t_{i+r-1})$$

hat man für die numerische Lösung unter der Annahme $b_r = 0$ (explizites MSV)

$$\eta_{i+r} + \sum_{k=0}^{r-1} a_k \eta_{i+k} = h \sum_{k=0}^{r-1} b_k f_{i+k}$$

und für die exakte Lösung

$$z(t_{i+r}) + \sum_{k=0}^{r-1} a_k z(t_{i+k}) = h \sum_{k=0}^{r-1} b_k f(t_{i+k}, z(t_{i+k})) + h \tau(t_i, \eta_i, h),$$

sodass sich daraus die Beziehung

$$\tau(t_i, \eta_i, h) = \frac{1}{h} (z(t_{i+r}) - \eta_{i+r}) \quad (6.62)$$

ergibt. Die Darstellung (6.62) ist analog zur Definition 5.1 des lokalen Diskretisierungsfehlers bei ESV.

Im Abschnitt 6.2 sind die Bedingungsgleichungen für die Koeffizienten a_k, b_k durch einen Abgleich hergeleitet und im Satz 6.1 die Konsistenzordnung p formuliert worden.

Der lokale Diskretisierungsfehler hat die kompakte Form

$$\tau(t_i, \eta_i, h) = \frac{1}{h} z(t_i) \sum_{l=0}^r a_l + \sum_{k=1}^p \left(\frac{h^{k-1}}{k!} z(t_i)^{(k)} \sum_{l=0}^r (l^k a_l - k l^{k-1} b_l) \right) + \mathcal{O}(h^p). \quad (6.63)$$

Ein MSV ist demnach konsistent, falls

$$\sum_{l=0}^r a_l = 0 \quad \text{und} \quad \sum_{l=0}^r (l a_l - b_l) = \sum_{l=1}^r l a_l - \sum_{l=0}^r b_l = 0 \quad (6.64)$$

gelten. Für die Ordnung $p \geq 1$ müssen auch die höheren Terme in der Entwicklung (6.63) herausfallen, d.h.

$$\begin{aligned} \sum_{l=0}^r l^k a_l &= k \sum_{l=0}^r l^{k-1} b_l, \quad k = 1, 2, \dots, p, \quad \text{bzw.} \\ \sum_{l=1}^r l^k a_l &= k \sum_{l=1}^r l^{k-1} b_l, \quad k = 2, 3, \dots, p. \end{aligned} \quad (6.65)$$

Alternativ zu den Bedingungsgleichungen (6.64) und (6.65) benutzt man die charakteristischen Polynome $\rho(\xi)$, $\sigma(\xi)$ aus (6.57).

Die Äquivalenz von (6.58) und (6.59) wurde schon genannt. Die weiteren Bedingungen erhält man über den Ansatz mit $\xi = e^h$ und

$$\begin{aligned}\rho(e^h) &= \sum_{k=0}^r a_k e^{kh} = \sum_{k=0}^r a_k \left(1 + kh + \frac{1}{2}(kh)^2 + \dots\right), \\ \sigma(e^h) &= \sum_{k=0}^r b_k e^{kh} = \sum_{k=0}^r b_k \left(1 + kh + \frac{1}{2}(kh)^2 + \dots\right).\end{aligned}\tag{6.66}$$

Damit gelangt man zu

$$\begin{aligned}\rho(e^h) - h\sigma(e^h) &= \sum_{k=0}^r a_k + h \sum_{k=0}^r (k a_k - b_k) + \frac{h^2}{2!} \sum_{k=0}^r (k^2 a_k - 2k b_k) + \\ &+ \dots + \frac{h^p}{p!} \sum_{k=0}^r (k^p a_k - p k^{p-1} b_k) + \mathcal{O}(h^{p+1}).\end{aligned}\tag{6.67}$$

Das heißt, die Entwicklung (6.63) lässt sich auch über die Exponentialfunktion kompakt darstellen. Setzt man noch $h = \ln(\mu)$, dann ist $h = 0$ gdw. $\mu = 1$ und es gilt

$$\rho(e^h) - h\sigma(e^h) = \mathcal{O}(h^{p+1}) \Leftrightarrow \rho(\mu) - \ln(\mu)\sigma(\mu) \text{ hat } \mu = 1 \text{ als } (p+1)\text{-fache NST.}$$

Damit lässt sich der Satz 6.1 verallgemeinern.

Satz 6.6 *Das lineare MSV (6.8) hat die Konsistenzordnung p ($p \geq 1$), falls eine der drei äquivalenten Bedingungen erfüllt ist.*

$$(1) \quad \sum_{l=0}^r a_l = 0 \quad \text{und} \quad \sum_{l=0}^r l^k a_l = k \sum_{l=0}^r l^{k-1} b_l, \quad k = 1, 2, \dots, p.$$

$$(2) \quad \rho(e^h) - h\sigma(e^h) = \mathcal{O}(h^{p+1}) \quad \text{für } h \rightarrow 0.$$

$$(3) \quad \rho(\mu) - \ln(\mu)\sigma(\mu) \text{ hat } \mu = 1 \text{ als } (p+1)\text{-fache Nullstelle.}$$

Was ist nun die maximal erreichbare Konsistenzordnung?

Insgesamt hat man $2r + 2$ Parameter $a_0, \dots, a_r, b_0, \dots, b_r$. Nach der Normierung $a_r = 1$ stehen $2r + 1$ Parameter zur Verfügung. Somit existiert ein implizites MSV der Ordnung $p = 2r$ und ein explizites mit $p = 2r - 1$. Die maximal erreichbare Ordnung ist also deutlich höher als bei den bisher vorgestellten MSV mittels Interpolationszugang. Allerdings sind solche Verfahren nicht mehr stabil.

Die Konstruktion des impliziten 2-Schritt-Verfahrens

$$a_2 \eta_{i+1} + a_1 \eta_i + a_0 \eta_{i-1} = h(b_2 f_{i+1} + b_1 f_i + b_0 f_{i-1})$$

mit $r = 2$ mittels der Konsistenzbedingungen als Gleichungssystem ist schon im Beispiel 6.5 (c) gezeigt worden.

Das Ergebnis ist die 2-stufige Milne-Simpson-Formel der Ordnung 4

$$\eta_{i+1} - \eta_{i-1} = \frac{h}{3} (f_{i+1} + 4f_i + f_{i-1}).$$

Beispiel 6.8 Konsistenzordnung von MSV(a) Explizites MSV $\eta_{i+1} + 4\eta_i - 5\eta_{i-1} = h(4f_i + 2f_{i-1})$ Konsistenz: $a_0 + a_1 + a_2 = 0, \quad a_1 + 2a_2 - (b_0 + b_1 + b_2) = 0,$ Ordnung $p = 2$: $a_1 + 4a_2 - 2(b_1 + 2b_2) = 0,$ Ordnung $p = 3$: $a_1 + 8a_2 - 3(b_1 + 4b_2) = 0,$ Ordnung $p = 4$: $a_1 + 16a_2 - 4(b_1 + 8b_2) = 4 \neq 0.$ (b) Implizites MSV AM(2) $\eta_{i+1} - \eta_i = \frac{h}{2}(f_{i+1} + f_i)$ Konsistenz: $a_0 + a_1 = 0, \quad a_1 - (b_0 + b_1) = 0,$ Ordnung $p = 2$: $a_1 - 2b_1 = 0,$ Ordnung $p = 3$: $a_1 - 3b_1 = -\frac{1}{2} \neq 0.$

Bei der Konvergenz von MSV geht es um das Verhalten der Näherungslösung für $h \rightarrow 0$ in einem endlichen Intervall und nicht um das Langzeitverhalten von η_i bei festem h und $i \rightarrow \infty$. Bei letzterem kann auch die Näherungslösung eines stabilen und konsistenten MSV immer mehr von der exakten Lösung abweichen.

Das folgende zentrale Ergebnis, nach welchem Konsistenz und Stabilität von MSV zur Konvergenz führen, geben wir zunächst ohne Beweis an.

Wir notieren den Konvergenzsatz in einer "älteren Ausgabe" und dann in Kurzform.

Satz 6.7 P. Henrici⁹ 1962

Betrachtet wird zum AWP $x' = f(t, x)$, $x(t_0) = x_0$, das MSV (6.56).

Für die Anlaufwerte η_i , $i = 0, 1, \dots, r-1$, des MSV gelte

$$\lim_{h \rightarrow 0} \eta_i = x(t_0), \quad \eta_i \approx x(t_i), \quad t_i = t_0 + ih.$$

Notwendige Bedingung für die Konvergenz des MSV, d.h.

$$\lim_{h \rightarrow 0} \eta_i = x(t), \quad t = t_0 + ih \in [t_0, T],$$

ist die Erfüllung des Stabilitätskriteriums im Satz 6.5.

Das Stabilitätskriterium ist auch hinreichend für die Konvergenz, falls zusätzlich das MSV mindestens die Konsistenzordnung 1 (vergl. Definition 6.4) besitzt.

Satz 6.8 Ein MSV der Form (6.56) ist genau dann konvergent für beliebige AWP, wenn es konsistent ist und das Stabilitätskriterium des Satzes 6.5 erfüllt.

Konsistenz mit der Ordnung $p \geq 1$ und Stabilitätskriterium liefern die Konvergenzordnung p .

Für den Beweis braucht man einige Vorbetrachtungen aus der Algebra über Matrizen, Begleitmatrix, Jordansche Normalform und Eigenwerte [75].

⁹Peter Henrici (1923-1987), schweizer Mathematiker, Theorie und Numerik gDGl, Approximation, komplexe Analysis

Die Matrix $A(n, n)$ hat das allgemeine charakteristische Polynom

$$p_n(\lambda) = \det(A - \lambda I) = (-1)^n \lambda^n + c_1 \lambda^{n-1} + \dots + c_{n-1} \lambda + c_n. \quad (6.68)$$

Das gleiche charakteristische Polynom erhalten wir mit der einfachen Matrix

$$\hat{A}(n, n) = \begin{pmatrix} 0 & 1 & \cdot & \cdots & \cdot & 0 \\ 0 & 0 & 1 & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdots & 0 & 1 \\ (-1)^{n+1}c_n & (-1)^{n+1}c_{n-1} & (-1)^{n+1}c_{n-2} & \cdots & (-1)^{n+1}c_2 & (-1)^{n+1}c_1 \end{pmatrix}. \quad (6.69)$$

Die Matrix \hat{A} wird auch als Begleitmatrix oder erzeugende Matrix bezeichnet.

Den Nachweis kann man führen, indem man die Determinante $\det(\hat{A} - \lambda I)$ nach der ersten Spalte oder auch letzten Zeile entwickelt.

Weiterhin brauchen wir den Begriff der Jordanschen Normalform.

Definition 6.5 Jordansche Normalform

Eine Blockdiagonalmatrix $J = \text{diag}(J_1, J_2, \dots, J_p)$ mit den quadratischen Jordan-Zellen (Jordan-Kästchen)

$$J_j(k, k)(\lambda) = \begin{pmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda \end{pmatrix}, \quad j = 1, 2, \dots, p, \quad (6.70)$$

der Dimension $k \times k$ heißt Jordansche Normalform. Im Fall $k = 1$ vereinfacht sich die Jordan-Zelle zu $J_j(1, 1)(\lambda) = (\lambda)$.

Lemma 6.9 Jordansche Normalform

Jede reelle Matrix $A(n, n)$ mit reellen Eigenwerten ist ähnlich zu einer Matrix in Jordanscher Normalform. Zu jedem Eigenwert λ_i mit seiner algebraischen und geometrischen Vielfachheit n_i bzw. m_i ($1 \leq m_i \leq n_i \leq n$) gehören m_i Jordan-Zellen, wobei die Summe ihrer Dimensionen gleich n_i ist. Wenn wir q verschiedene Eigenwerte λ_i voraussetzen, können wir die Jordansche Normalform mit der regulären Matrix T als

$$J = T^{-1}AT = \begin{pmatrix} J_1(k_1^{(1)}, k_1^{(1)})(\lambda'_1) & & & & \\ & \ddots & & & \\ & & J_{m_1}(k_{m_1}^{(1)}, k_{m_1}^{(1)})(\lambda'_1) & & \\ & & & \ddots & \\ & & & & J_1(k_1^{(q)}, k_1^{(q)})(\lambda'_q) \\ & & & & & \ddots \\ & & & & & & J_{m_q}(k_{m_q}^{(q)}, k_{m_q}^{(q)})(\lambda'_q) \end{pmatrix} \quad (6.71)$$

notieren, wobei

$$\lambda'_1 = \lambda_1 = \lambda_2 = \dots = \lambda_{n_1},$$

$$\lambda'_2 = \lambda_{n_1+1} = \lambda_{n_1+2} = \dots = \lambda_{n_1+n_2},$$

$$\lambda'_3 = \lambda_{n_1+n_2+1} = \lambda_{n_1+n_2+2} = \dots = \lambda_{n_1+n_2+n_3},$$

...

$$\lambda'_q = \lambda_{n_1+n_2+\dots+n_{q-1}+1} = \lambda_{n_1+n_2+\dots+n_{q-1}+2} = \dots = \lambda_{n_1+n_2+\dots+n_q}, \quad n = \sum_{i=1}^q n_i,$$

$$J_j(k_j^{(i)}, k_j^{(i)})(\lambda'_i) = \begin{pmatrix} \lambda'_i & 1 & & & \\ & \lambda'_i & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda'_i \end{pmatrix}, \quad j = 1, 2, \dots, m_i, \quad n_i = \sum_{j=1}^{m_i} k_j^{(i)},$$

$$i = 1, 2, \dots, q.$$

Beweis. [12], [32]

Der Satz gilt auch in Fall komplexer Eigenwerte, wobei dann T komplexe Elemente enthält.

Folgerung 6.10

Wenn in $J = \text{diag}(J_1, J_2, \dots, J_p)$ der Index $p = n$ ist, dann hat J die Form einer Diagonalmatrix, die Jordan-Zellen sind eindimensional, die Matrix A ist diagonalisierbar, ihre Elementarteiler $\lambda_i - \lambda$ sind ausschließlich linear und es gibt zu jedem Eigenwert λ_i ein System von $m_i = n_i$ linear unabhängigen Eigenvektoren. Damit ist das Gesamtsystem der n Eigenvektoren linear unabhängig.

Beispiel 6.9

$$(1) \quad A = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}, \quad \text{EW: } \lambda_1 = 1, \lambda_2 = 2,$$

EV: $x_1 = (1, -1)^T$, $x_2 = (0, 1)^T$ linear unabhängig,

$$X = (x_1, x_2) = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}, \quad X^{-1} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad X^{-1}AX = \text{diag}(\lambda_1, \lambda_2) = \text{diag}(J_1, J_2).$$

(2) Die Matrix $A(9, 9)$ möge folgende EW mit ihren Vielfachheiten haben.

$$\lambda'_1 = 1 = \lambda_1 = \dots = \lambda_5, \quad m_1 = 2, \quad n_1 = 5,$$

$$\lambda'_2 = -1 = \lambda_6 = \dots = \lambda_9, \quad m_1 = 3, \quad n_2 = 4.$$

Ihre Jordan-Normalform mit den $p = 5$ Jordan-Zellen ist

$$J = \begin{pmatrix} \boxed{\begin{matrix} 1 & 1 \\ & 1 & 1 \\ & & 1 \end{matrix}} & & & & \\ & \boxed{\begin{matrix} 1 & 1 \\ & 1 \end{matrix}} & & & \\ & & \boxed{\begin{matrix} -1 & 1 \\ & -1 \end{matrix}} & & \\ & & & \boxed{-1} & \\ & & & & \boxed{-1} \end{pmatrix}.$$

Die Elementarteiler sind $(1 - \lambda)^3$, $(1 - \lambda)^2$, $(-1 - \lambda)^2$, $(-1 - \lambda)$, $(-1 - \lambda)$.

Das charakteristische Polynom hat die Gestalt $p_9(\lambda) = (-1)^9(\lambda - 1)^5(\lambda + 1)^4$.

(3) Die Matrix $A(9, 9)$ möge folgende EW mit ihren Vielfachheiten haben.

$$\lambda'_1 = 1 = \lambda_1, \quad m_1 = 1, \quad n_1 = 1,$$

$$\lambda'_2 = -1 = \lambda_2, \quad m_2 = 1, \quad n_2 = 1,$$

$$\lambda'_3 = 0.9 = \lambda_3 = \lambda_4, \quad m_3 = 1, \quad n_3 = 2,$$

$$\lambda'_4 = 0.8 = \lambda_5 = \dots = \lambda_8, \quad m_4 = 2, \quad n_4 = 4,$$

$$\lambda'_5 = 0.7 = \lambda_9, \quad m_5 = 1, \quad n_5 = 1.$$

Ihre Jordansche Normalform mit den $p = 5$ Jordan-Zellen ist

$$J = \begin{pmatrix} \boxed{1} & & & & \\ & \boxed{-1} & & & \\ & & \boxed{\begin{matrix} 0.9 & 1 \\ & 0.9 \end{matrix}} & & \\ & & & \boxed{\begin{matrix} 0.8 & 1 \\ & 0.8 & 1 \\ & & 0.8 \end{matrix}} & \\ & & & & \boxed{0.8} \\ & & & & & \boxed{0.7} \end{pmatrix}.$$

Die Elementarteiler sind $(1 - \lambda)$, $(-1 - \lambda)$, $(0.9 - \lambda)^2$, $(0.8 - \lambda)^3$, $(0.8 - \lambda)$, $(0.7 - \lambda)$.

Das charakteristische Polynom ist $p_9(\lambda) = (-1)^9(\lambda - 1)(\lambda + 1)(\lambda - 0.9)^2(\lambda - 0.8)^4(\lambda - 0.7)$.

Beweis. Satz 6.8

Wir zeigen beide Richtungen des Beweises.

Der Teil, dass aus der Konvergenz des MSV sowohl seine Konsistenz als auch das Stabilitätskriterium folgt, ist etwas einfacher.

Zu dessen Nachweis notieren wir zwei Versionen, in der zweiten ist eine Fallunterscheidung bzgl. der Nullstellen des charakteristischen Polynoms $\rho(\xi)$ sinnvoll. Von diesen Fällen betrachten wir ausführlich nur die ersten drei.

(1) Konvergenz des MSV \Rightarrow Konsistenz

Die Konvergenz haben wir in der Definition 6.4 (b) erläutert und sie bedeutet

$$\lim_{h \rightarrow 0} \eta_N = x(t), \quad h = (t - t_0)/N, \quad t \in [t_0, T].$$

Wir nehmen als gDGI $x' = f(t, x)$ die speziellen AWP

$$\begin{aligned} x' &= 0, & x(0) &= 1, \\ x' &= 1, & x(0) &= 0. \end{aligned}$$

Wenn die Lösung des MSV bei genauen AB mit $h \rightarrow 0$ gegen die exakte Lösung $x(t) \equiv 1$ bzw. $x(t) = t$ konvergiert, dann heißt das $\lim_{h \rightarrow 0} \eta_i = 1$ bzw. $\eta_i = ih$. Daraus folgen die beiden Konsistenzbedingungen in (6.58).

Im Detail. Im ersten Fall ist $\eta_0 = \dots = \eta_{r-1} = 1$ und $a_0 + a_1 + \dots + a_{r-1} + a_r \eta_r = 0$ für alle h , sodass aus der Konvergenz $\eta_r \rightarrow 1$ die Forderung $\rho(1) = \sum_{k=0}^r a_k = 0$ folgt.

Im zweiten Fall mit den Startdaten $\eta_i = ih$, $i = 0, 1, \dots, r-1$, lautet das MSV

$$\begin{aligned} h \sum_{k=0}^{r-1} k a_k + a_r \eta_r &= h \sum_{k=0}^r b_k \cdot 1, \\ h \sum_{k=1}^r k a_k + a_r (\eta_r - hr) &= h \sum_{k=0}^r b_k, \\ h \rho'(1) + a_r (\eta_r - hr) &= h \sigma(1). \end{aligned}$$

Aus der Konvergenz $\eta_r \rightarrow hr$ ergibt sich $\rho'(1) = \sigma(1)$.

(2) Konvergenz des MSV \Rightarrow Stabilitätskriterium

(2.1) Kurzversion, indirekter Beweis

Angenommen das MSV ist nicht stabil.

Das charakteristische Polynom $\rho(\xi)$ hat dann eine Nullstelle ξ_1 mit $|\xi_1| > 1$ oder eine mehrfache Nullstelle ξ_1 mit $|\xi_1| = 1$.

Wir betrachten das AWP $x' = 0$, $x(0) = 0$, mit seiner exakten Lösung $x(t) \equiv 0$ sowie die Wurzeln des charakteristischen Polynoms $\rho(\xi)$ des MSV.

Wir stören nun die Startdaten η_i , $i = 0, 1, \dots, r-1$, sodass aber $\eta_i = \mathcal{O}(h^q)$, $q > 0$, und $\lim_{h \rightarrow 0} \eta_i = x(0) = 0$.

(a) $\rho(\xi_1) = 0$ mit $|\xi_1| > 1$ sind.

Die anderen Nullstellen seien ξ_2, ξ_3, \dots . Die allgemeine Lösung der Differenzgleichung ist

$$\eta_i = c_1 \xi_1^i + c_2 \xi_2^i + c_3 \xi_3^i + \dots + c_r \xi_r^i.$$

Anstelle der exakten AB $\eta_i = 0$, $i = 0, 1, \dots, r-1$, verwendet man die gestörte AB in der Gestalt $\eta_i = \sqrt{h} \xi_1^i$, $i = 0, 1, \dots, r-1$. Daraus ergeben sich die Koeffizienten c_k der Näherungslösung, nämlich $c_1 = \sqrt{h}$ und $c_2 = c_3 = \dots = c_r = 0$, sowie

$$\eta_i = c_1 \xi_1^i = \sqrt{h} \xi_1^i.$$

Im Lösungsintervall $[0, t] \subset [0, T]$ und bei der Schrittweite $h = \frac{t}{N}$ ergibt sich

$$\eta_N = \sqrt{h} \xi_1^N = \sqrt{h} \xi_1^{t/h}.$$

Mit $|\xi_1| > 1$ und $\lim_{h \rightarrow 0} \sqrt{h} |\xi_1|^{t/h} = \infty$ folgt die Divergenz, im Widerspruch zur Voraussetzung.

(b) $\rho(\xi_1) = 0$ mit $|\xi_1| = 1$ mehrfach, der Einfachheit halber zweifach.

Die anderen Nullstellen seien ξ_3, ξ_4, \dots . Die allgemeine Lösung der Differenzgleichung ist

$$\eta_i = c_1 \xi_1^i + c_2 i \xi_1^i + c_3 \xi_3^i + \dots + c_r \xi_r^i.$$

Anstelle der exakten AB $\eta_i = 0$, $i = 0, 1, \dots, r-1$, verwendet man die gestörte AB in der Gestalt $\eta_0 = 0$, $\eta_i = \sqrt{h} i \xi_1^i$, $i = 1, 2, \dots, r-1$. Daraus ergeben sich erneut die Koeffizienten c_k der Näherungslösung, nämlich $c_2 = \sqrt{h}$ und $c_1 = c_3 = \dots = c_r = 0$, sowie

$$\eta_i = c_2 i \xi_1^i = \sqrt{h} i \xi_1^i.$$

Im Lösungsintervall $[0, t]$ und bei der Schrittweite $h = \frac{t}{N}$ folgen

$$\eta_N = \sqrt{h} N \xi_1^N = \frac{t}{\sqrt{h}} \xi_1^N, \quad |\eta_N| = \frac{t}{\sqrt{h}},$$

sodass sofort $\lim_{h \rightarrow 0} |\eta_N| = \infty$ die Divergenz bedeutet.

(2.2) Langversion

Auch hier brauchen wir das AWP $x' = 0$, $x(0) = 0$, mit seiner exakten Lösung $x(t) \equiv 0$ sowie das charakteristische Polynom $\rho(\xi)$ und seine Ableitung, also

$$\rho(\xi) = \sum_{k=0}^r a_k \xi^k, \quad \rho'(\xi) = \sum_{k=1}^r k a_k \xi^{k-1}.$$

Seien $\xi_1, \xi_2, \dots, \xi_r$ die r komplexen Nullstellen von $\rho(\xi)$, d.h.

$$\begin{aligned} \rho(\xi_j) &= \sum_{k=0}^r a_k \xi_j^k = 0, \quad j = 1, 2, \dots, r, \quad a_r = 1, \\ \xi_j^r &= -\sum_{k=0}^{r-1} a_k \xi_j^k. \end{aligned}$$

1. Fall: Die Nullstellen ξ_j sind einfach und reell.

Die reellen Zahlen

$$v_l = \sum_{j=1}^r h \xi_j^l, \quad l \geq r,$$

erfüllen die Gleichung des MSV zum AWP $x' = 0$, $x(0) = 0$, was einfach durch Einsetzen nachzurechnen ist.

$$\begin{aligned} \sum_{k=0}^r a_k \eta_{i+k} - h \sum_{k=0}^r b_k f_{i+k} &= \sum_{k=0}^r a_k \eta_{i+k}, \quad f = 0, \quad a_r = 1, \quad i \geq 0, \\ &= \sum_{k=0}^r \left(a_k \sum_{j=1}^r h \xi_j^{i+k} \right) = \sum_{j=1}^r \left(h \sum_{k=0}^r a_k \xi_j^{i+k} \right) = \sum_{j=1}^r \left(h \xi_j^i \underbrace{\sum_{k=0}^r a_k \xi_j^k}_{=\rho(\xi_j)=0} \right) = 0. \end{aligned}$$

Damit gilt $\eta_i = v_i$, $i = r, r+1, \dots$.

Des Weiteren sind für $i = 0, 1, \dots, r-1$ die AB gemäß

$$\lim_{h \rightarrow 0} v_i = \lim_{h \rightarrow 0} \sum_{j=1}^r h \xi_j^i = 0 = x(0)$$

erfüllt.

Die Lösung des MSV in der neuen Darstellung

$$\eta_i = \sum_{j=1}^r h \xi_j^i, \quad i = r, r+1, \dots,$$

kann bei $h \rightarrow 0$ nur gegen die exakte Nulllösung tendieren, falls alle $|\xi_j| \leq 1$ sind, womit das Stabilitätskriterium gilt.

Würde nämlich irgendein ξ_s betragsmäßig größer als Eins sein, dann ist

$$\lim_{h \rightarrow 0} h |\xi_s|^i = \lim_{h \rightarrow 0} h(1+\delta)^i = \infty,$$

denn letzteres verhält sich wie $h(1+\delta)^{1/h} \rightarrow \infty$. Die Konvergenz gegen Null wäre damit nicht möglich.

2. Fall: Nullstellen ξ_j sind einfach und reell oder komplex.

Komplexe Nullstellen eines reellen Polynoms treten als Paare auf, nämlich als komplexer und dazu konjugiert komplexer Wert. Sei z.B. $\xi_1 = r e^{i\varphi}$ und $\xi_2 = \overline{\xi_1}$. Bei weiteren Paaren ist die Vorgehensweise analog.

Damit die allgemeine Lösung des MSV

$$\eta_i = \sum_{s=1}^r \gamma_s \xi_s^i$$

reell ist, muss man die durch die reellen Startwerte $\eta_0, \eta_1, \dots, \eta_{r-1}$ eindeutig bestimmten Koeffizienten γ_s betrachten. Man erhält

$$\begin{aligned} \eta_0 &= \sum_{s=1}^r \gamma_s \xi_s^0 = \sum_{s=1}^r \gamma_s, \\ \eta_1 &= \sum_{s=1}^r \gamma_s \xi_s^1 = \gamma_1 \xi_1 + \gamma_2 \xi_2 + \sum_{s=3}^r \gamma_s \xi_s \\ &= \gamma_1 r e^{i\varphi} + \gamma_2 r e^{-i\varphi} + \sum_{s=3}^r \gamma_s \xi_s \\ &= (\gamma_1 + \gamma_2) r \cos(\varphi) + i(\gamma_1 - \gamma_2) r \sin(\varphi) + \sum_{s=3}^r \gamma_s \xi_s, \end{aligned}$$

sodass $\gamma_1 = \gamma_2$ folgt.

Es gilt natürlich $\xi_1 + \xi_2 = 2\Re\xi_1 \in \mathbb{R}$, wie auch allgemein die Beziehung $\xi_1^l + \xi_2^l = \xi_1^l + \overline{\xi_1^l} = \xi_1^l + \xi_1^{\overline{l}} \in \mathbb{R}$ für $l = 0, 1, 2, \dots$.

Damit kann man die reellen Zahlen

$$v_l = \sum_{j=1}^r h \xi_j^l, \quad l \geq r,$$

definieren und weiter analog zum ersten Fall vorgehen.

3. Fall: Nullstellen ξ_j sind reell und es können mehrfache auftreten.

Wir betrachten nur die Situation, dass ξ_1 eine zweifache reelle Nullstelle ist, also $\xi_2 = \xi_1$. Bei anderen Nullstellen mit Vielfachheiten ist die Vorgehensweise entsprechend.

Es genügt zu zeigen, dass $|\xi_1| < 1$ sein muss.

Zunächst bedeutet die Zweifachheit der Nullstelle ξ_1 von $\rho(\xi)$ die Gültigkeit der Beziehungen

$$\rho(\xi_1) = \sum_{k=0}^r a_k \xi_1^k = 0, \quad \rho'(\xi_1) = \sum_{k=1}^r k a_k \xi_1^{k-1} = 0.$$

Wie in den bisherigen Fällen definieren wir geeignete Zahlen v_l , die Lösung des MSV sind und gegen die exakte Nulllösung des AWP konvergieren. Sei

$$v_l = h \xi_1^l + hl \xi_1^l + \sum_{j=3}^r h \xi_j^l, \quad l \geq r.$$

Diese Zahlen erfüllen die Gleichung des MSV zum AWP $x' = 0$, $x(0) = 0$, was durch Einsetzen nachzurechnen ist.

$$\begin{aligned} & \sum_{k=0}^r a_k \eta_{i+k} - h \sum_{k=0}^r b_k f_{i+k} = \sum_{k=0}^r a_k \eta_{i+k}, \quad f = 0, \quad a_r = 1, \quad i \geq 0, \\ & = \sum_{k=0}^r a_k \left(h \xi_1^{i+k} + h(i+k) \xi_1^{i+k} + \sum_{j=3}^r h \xi_j^{i+k} \right) \\ & = h \left[\left(\xi_1^{i+r} + (i+r) \xi_1^{i+r} + \sum_{j=3}^r \xi_j^{i+r} \right) + a_{r-1} \left(\xi_1^{i+r-1} + (i+r-1) \xi_1^{i+r-1} + \sum_{j=3}^r \xi_j^{i+r-1} \right) \right. \\ & \quad \left. + a_{r-2} \left(\xi_1^{i+r-2} + (i+r-2) \xi_1^{i+r-2} + \sum_{j=3}^r \xi_j^{i+r-2} \right) + \dots + a_0 \left(\xi_1^i + i \xi_1^i + \sum_{j=3}^r \xi_j^i \right) \right] \\ & = h \left[\xi_1^i \sum_{k=0}^r a_k \xi_1^k + \xi_1^i \sum_{k=0}^r (i+k) a_k \xi_1^k + \xi_3^i \sum_{k=0}^r a_k \xi_3^k + \dots + \xi_r^i \sum_{k=0}^r a_k \xi_r^k \right], \quad \rho(\xi_s) = 0 \\ & = h \xi_1^i \sum_{k=0}^r (i+k) a_k \xi_1^k = h \xi_1^i \left(i \sum_{k=0}^r a_k \xi_1^k + \xi_1 \sum_{k=1}^r k a_k \xi_1^{k-1} \right) \\ & = h \xi_1^i [i \rho(\xi_1) + \xi_1 \rho'(\xi_1)] \\ & = 0. \end{aligned}$$

Damit gilt $\eta_i = v_i$, $i = r, r+1, \dots$.

Weiterhin sind für $i = 0, 1, \dots, r-1$ die AB gemäß

$$\lim_{h \rightarrow 0} v_i = \lim_{h \rightarrow 0} h \left(\xi_1^i + i \xi_1^i + \sum_{j=3}^r \xi_j^i \right) = 0 = x(0)$$

erfüllt. Die Lösung des MSV in der neuen Darstellung

$$\eta_i = h \xi_1^i + h i \xi_1^i + \sum_{j=3}^r h \xi_j^i, \quad i = r, r+1, \dots,$$

kann bei $h \rightarrow 0$ nur gegen die exakte Nulllösung tendieren, falls $|\xi_j| \leq 1$, $j = 3, 4, \dots, r$, und wegen $hi \approx t - t_0 > 0$ falls $|\xi_1| < 1$ ist, womit das Stabilitätskriterium gilt.

4. Fall: Nullstellen ξ_j sind reell oder komplex und es können mehrfache auftreten.

Zum Nachweis vereinigt man alle bisherigen Aspekte.

(3) Stabilität und Konsistenz des MSV sind auch hinreichend für die Konvergenz.

(3.1) Zunächst muss der formale Rahmen für den Konvergenzbeweis in $[t_0, T]$ bereitgestellt werden. Dazu braucht man einige algebraische und funktionalanalytische Grundlagen, die zum Teil schon erläutert wurden.

Weiterhin kann damit auch der allgemeine Fall eines SysgDGI der Dimension m einbezogen werden.

Wir schreiben zunächst das MSV (6.56) als

$$\eta_{i+r} = - \sum_{k=0}^{r-1} a_k \eta_{i+k} + h \sum_{k=0}^{r-1} b_k f(t_i + kh, \eta_{i+k}) + h b_r f(t_i + rh, \eta_{i+r}).$$

Die letzten beiden Terme auf der rechten Seite ohne den Faktor h fassen wir zur implizit definierten Größe $\psi = \psi(t_i, \eta_i, \dots, \eta_{i+r-1}, h)$ mit

$$\psi = \sum_{k=0}^{r-1} b_k f(t_{i+k}, \eta_{i+k}) + b_r f\left(t_i + rh, - \sum_{k=0}^{r-1} a_k \eta_{i+k} + h \psi\right). \quad (6.72)$$

zusammen. Das MSV lautet nun kurz

$$\eta_{i+r} = - \sum_{k=0}^{r-1} a_k \eta_{i+k} + h \psi. \quad (6.73)$$

Wenn \mathbf{f} von der Dimension m ist, sind die Größen η_i m -dimensionale Zeilenvektoren, sodass die (r, m) -Matrix

$$\mathbf{Y}_i = (\eta_{i+r-1}, \eta_{i+r-2}, \dots, \eta_i)^T$$

die Daten aus der "Vergangenheit" zur Berechnung von η_{i+r} bzw. die sogenannte "Schleppe" des MSV enthält. Wir bleiben der Einfachheit halber bei $m = 1$.

Die Idee für die Konvergenzanalyse besteht nun darin, das MSV auf ein ESV bzgl. \mathbf{Y}_i zurückzuführen.

Ziel wäre eine Formulierung des MSV in der Gestalt $\mathbf{Y}_{i+1} = \mathbf{Y}_i + h\Psi(t_i, \mathbf{Y}_i, h)$ mit einer Verfahrensfunktion Ψ wie in Definition 5.1. Dies ist zwar nicht ganz erreichbar, aber es gilt

$$\mathbf{Y}_{i+1} = \begin{pmatrix} \eta_{i+r} \\ \eta_{i+r-1} \\ \vdots \\ \eta_{i+1} \end{pmatrix} = \begin{pmatrix} -\sum_{k=0}^{r-1} a_k \eta_{i+k} + h \psi(t_i, \mathbf{Y}_i, h) \\ \eta_{i+r-1} \\ \vdots \\ \eta_{i+1} \end{pmatrix}.$$

Die zugehörige Matrix-Vektor-Notation lautet

$$\mathbf{Y}_{i+1} = A\mathbf{Y}_i + h\Psi(t_i, \mathbf{Y}_i, h) \quad (6.74)$$

mit

$$A = \begin{pmatrix} -a_{r-1} & -a_{r-2} & \cdots & -a_1 & -a_0 \\ 1 & 0 & \cdots & \cdot & 0 \\ & 1 & & \cdot & 0 \\ & & \ddots & \vdots & \vdots \\ & & & 1 & 0 \end{pmatrix}, \quad \Psi(t_i, \mathbf{Y}_i, h) = \begin{pmatrix} \psi(t_i, \mathbf{Y}_i, h) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Die Rekursion (6.74) charakterisiert also das MSV.

Hätten wir $A = I$, dann läge ein ESV in der üblichen Notation vor und wir wären praktisch fertig.

Die Matrix $A(r, r)$ enthält mit den a_j die Koeffizienten, welche die Stabilität des Verfahrens bestimmen.

Es gilt nun Folgendes. Falls das Stabilitätskriterium (Satz 6.5) vom MSV erfüllt wird, dann gibt es eine Vektornorm $\|\cdot\|$ auf \mathbb{R}^k , sodass

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\| \leq 1. \quad (6.75)$$

Man rechnet aus, dass die Eigenwerte der Matrix A genau die Wurzeln des charakteristischen Polynoms $\rho(\xi)$ sind (vergl. Polynom (6.68) und Begleitmatrix (6.69)). Damit erfüllen alle Eigenwerte von A die Bedingung $|\lambda_j| \leq 1$.

Andererseits kann man die Jordan-Normalform von A erzeugen (siehe Lemma 5.4) und die Matrix $J = T^{-1}AT$ hat die Form (6.71). Die einfachen Eigenwerte haben eindimensionale Jordan-Zellen, die mehrfachen besitzen Jordan-Zellen der Dimension ≥ 1 , bei mehrdimensionalen mit Einsen auf der oberen Nebendiagonale. Wir stellen diese Jordansche Normalform in der verkürzten Version

$$J = T^{-1}AT = \text{diag} \left(\lambda_1, \dots, \lambda_l, \begin{pmatrix} \lambda_{l+1} & \delta_{l+1} & & \\ & \ddots & \delta_{r-1} & \\ & & & \lambda_r \end{pmatrix} \right) \quad (6.76)$$

dar, wobei $\lambda_1, \dots, \lambda_l$ die einfachen sowie $\lambda_{l+1}, \dots, \lambda_r$ die mehrfachen Eigenwerte sind, zu welchen je nach Vielfachheit die Größen $\delta_{l+1} = 0$ oder $\delta_{l+1} = 1$ werden.

Da die mehrfachen Eigenwerte im Innern des Einheitskreises liegen, gibt es nun ein $\varepsilon > 0$ mit $|\lambda_j| < 1 - \varepsilon$, $j = l + 1, \dots, r$. Jetzt definiert man die Diagonalmatrix

$$D_\varepsilon = \text{diag}(1, \varepsilon, \varepsilon^2, \dots, \varepsilon^{r-1}).$$

Mit einer weiteren Ähnlichkeitstransformation ergibt sich

$$B = D_\varepsilon^{-1} J D_\varepsilon = \text{diag} \left(\lambda_1, \dots, \lambda_l, \begin{pmatrix} \lambda_{l+1} & \varepsilon \delta_{l+1} & & \\ & \ddots & \varepsilon \delta_{r-1} & \\ & & & \lambda_r \end{pmatrix} \right)$$

und mit der Zeilensummennorm

$$\|B\|_\infty = \max_{i=1(1)r} \sum_{j=1}^r |b_{ij}|$$

folgt

$$\|D_\varepsilon^{-1} J D_\varepsilon\|_\infty \leq 1. \quad (6.77)$$

Nun definieren wir die Vektornorm

$$\|x\| = \|D_\varepsilon^{-1} T^{-1} x\|_\infty,$$

die als Maximumnorm kompatibel mit der Zeilensummennorm ist.

Damit ergibt sich schließlich

$$\begin{aligned} \|Ax\| &= \|D_\varepsilon^{-1} T^{-1} A x\|_\infty = \|D_\varepsilon^{-1} T^{-1} A (T D_\varepsilon D_\varepsilon^{-1} T^{-1}) x\|_\infty \\ &= \|D_\varepsilon^{-1} (T^{-1} A T) D_\varepsilon D_\varepsilon^{-1} T^{-1} x\|_\infty = \|D_\varepsilon^{-1} J D_\varepsilon D_\varepsilon^{-1} T^{-1} x\|_\infty \\ &\leq \|D_\varepsilon^{-1} J D_\varepsilon\| \|D_\varepsilon^{-1} T^{-1} x\|_\infty \leq \|D_\varepsilon^{-1} T^{-1} x\|_\infty = \|x\| \end{aligned} \quad (6.78)$$

und die gewünschte Normungleichung $\|A\| \leq 1$.

Will man Abschätzungen mit der Vorschrift (6.74) als ESV vornehmen, werden die genannten Normen und $\|A\| \leq 1$ benötigt.

Für zwei diskrete Approximationen \mathbf{Y}_i und \mathbf{Z}_i ist

$$\mathbf{Y}_{i+1} = A \mathbf{Y}_i + h \Psi(t_i, \mathbf{Y}_i, h), \quad \mathbf{Z}_{i+1} = A \mathbf{Z}_i + h \Psi(t_i, \mathbf{Z}_i, h)$$

und somit

$$\|\mathbf{Y}_{i+1} - \mathbf{Z}_{i+1}\| \leq \|A\| \|\mathbf{Y}_i - \mathbf{Z}_i\| + h \|\Psi(t_i, \mathbf{Y}_i, h) - \Psi(t_i, \mathbf{Z}_i, h)\|. \quad (6.79)$$

Falls f Lipschitz-stetig ist, so auch ψ aus (6.72) und folglich Ψ bezüglich \mathbf{Y} . Mit $\|A\| \leq 1$ und der Lipschitz-Konstanten L folgt die Abschätzung

$$\|\mathbf{Y}_{i+1} - \mathbf{Z}_{i+1}\| \leq (1 + hL) \|\mathbf{Y}_i - \mathbf{Z}_i\|. \quad (6.80)$$

Die weitere Vorgehensweise zum Nachweis der Konvergenz ähnelt der für das ESV gemäß Satz 5.5. Allerdings wird hier nicht eine Folge exakter Lösungswerte $x(t_i)$ durch die Näherungsfolge η_i und damit der globale Diskretisierungsfehler e_i abgeschätzt, sondern wir haben eine Folge von Näherungsvektoren \mathbf{Y}_i , deren Komponenten mit einem konsistenten MSV erzeugt werden. Zum Schluss brauchen wir eine Genauigkeitsaussage in der Norm zum Fehlervektor

$$\mathbf{X}_n - \mathbf{Y}_n = (x(t_{n+r-1}), x(t_{n+r-2}), \dots, x(t_n))^T - (\eta_{n+r-1}, \eta_{n+r-2}, \dots, \eta_n)^T,$$

wobei $t_{n+r-1} = t_0 + (n+r-1)h \leq T$. Das erreichen wir durch spezielle Zwischenschritte als Folge numerischer Approximationen.

Wir erzeugen nicht nur zwei Folgen von Näherungen \mathbf{Y}_i und \mathbf{Z}_i , sondern eine Liste von solchen Folgen.

Dazu betrachten wir die exakte Lösung $x(t)$ in der Gestalt von Vektoren

$$\mathbf{X}_i = (x(t_{i+r-1}), x(t_{i+r-2}), \dots, x(t_i))^T = (x_{i+r-1}, x_{i+r-2}, \dots, x_i)^T, \quad i = 0(1)n-1,$$

die als Startwerte verwendet werden.

Und zwar, nimmt das MSV (6.74) als Startwert $\mathbf{Y}_i^{(i)} = \mathbf{X}_i$, dann erzeugt es die Folge der Vektoren $\mathbf{Y}_{i+1}^{(i)}, \mathbf{Y}_{i+2}^{(i)}, \dots, \mathbf{Y}_n^{(i)}$ gemäß

$$\mathbf{Y}_{i+1}^{(i)} = \begin{pmatrix} \eta_{i+r} \\ x_{i+r-1} \\ x_{i+r-2} \\ \vdots \\ x_{i+1} \end{pmatrix}, \quad \mathbf{Y}_{i+2}^{(i)} = \begin{pmatrix} \eta_{i+r+1} \\ \eta_{i+r} \\ x_{i+r-1} \\ \vdots \\ x_{i+2} \end{pmatrix}, \quad \dots, \quad \mathbf{Y}_{i+r}^{(i)} = \begin{pmatrix} \eta_{i+2r-1} \\ \eta_{i+2r-2} \\ \eta_{i+2r-3} \\ \vdots \\ \eta_{i+r} \end{pmatrix}, \quad \dots, \quad \mathbf{Y}_n^{(i)} = \begin{pmatrix} \eta_{n+r-1} \\ \eta_{n+r-2} \\ \eta_{n+r-3} \\ \vdots \\ \eta_n \end{pmatrix}.$$

Mit dem ersten Schritt ist

$$\mathbf{X}_{i+1} - \mathbf{Y}_{i+1}^{(i)} = \begin{pmatrix} x_{i+r} \\ x_{i+r-1} \\ x_{i+r-2} \\ \vdots \\ x_{i+1} \end{pmatrix} - \begin{pmatrix} \eta_{i+r} \\ x_{i+r-1} \\ x_{i+r-2} \\ \vdots \\ x_{i+1} \end{pmatrix} = h \tau(h)$$

mit dem lokalen Diskretisierungsfehler $\|\tau(h)\| = \mathcal{O}(h^p)$ für η_{i+r} aus dem MSV.

Wir starten nun mit $\mathbf{Y}_{i+1}^{(i+1)} = \mathbf{X}_{i+1}$ ein weiteres MSV, das dann die Folge $\mathbf{Y}_{i+2}^{(i+1)}, \mathbf{Y}_{i+3}^{(i+1)}, \dots, \mathbf{Y}_n^{(i+1)}$ liefert, also

$$\mathbf{Y}_{i+2}^{(i+1)} = \begin{pmatrix} \eta_{i+r+1} \\ x_{i+r} \\ x_{i+r-1} \\ \vdots \\ x_{i+2} \end{pmatrix}, \quad \mathbf{Y}_{i+3}^{(i+1)} = \begin{pmatrix} \eta_{i+r+2} \\ \eta_{i+r+1} \\ x_{i+r} \\ \vdots \\ x_{i+3} \end{pmatrix}, \quad \dots, \quad \mathbf{Y}_{i+r+1}^{(i+1)} = \begin{pmatrix} \eta_{i+2r} \\ \eta_{i+2r-1} \\ \eta_{i+2r-2} \\ \vdots \\ \eta_{i+r+1} \end{pmatrix}, \quad \dots, \quad \mathbf{Y}_n^{(i+1)} = \begin{pmatrix} \eta_{n+r-1} \\ \eta_{n+r-2} \\ \eta_{n+r-3} \\ \vdots \\ \eta_n \end{pmatrix}.$$

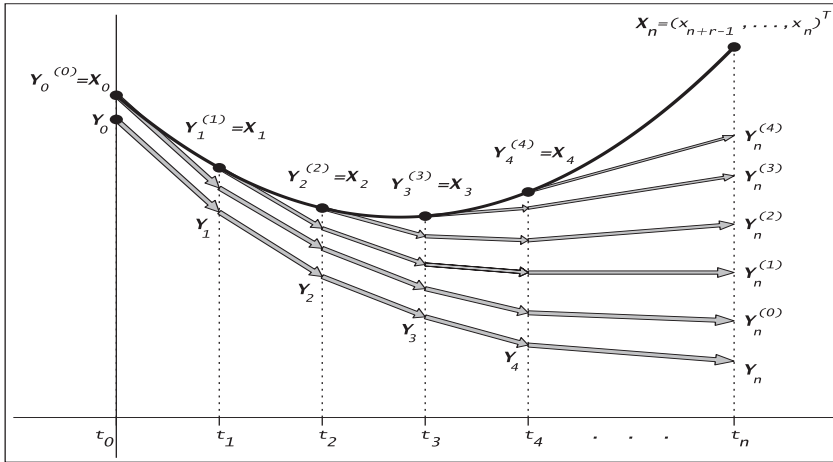


Abb. 6.7 Datei *msv_schritt1_01.ps*,

Vorgehen beim Konvergenzbeweis des MSV mit Zwischenschritten

Wir wenden nun die Abschätzung (6.80) auf die Differenz der beiden Folgen am Endpunkt an.

$$\begin{aligned}
 \|\mathbf{Y}_n^{(i+1)} - \mathbf{Y}_n^{(i)}\| &\leq (1 + hL) \|\mathbf{Y}_{n-1}^{(i+1)} - \mathbf{Y}_{n-1}^{(i)}\| \\
 &\leq (1 + hL)^{n-i-1} \|\mathbf{Y}_{i+1}^{(i+1)} - \mathbf{Y}_{i+1}^{(i)}\| \\
 &\leq (1 + hL)^{n-i-1} ch^{p+1}.
 \end{aligned}
 \tag{6.81}$$

Die Fortpflanzung der Differenz $\mathbf{X}_0 - \mathbf{Y}_0 = \mathbf{Y}_0^{(0)} - \mathbf{Y}_0$ in den Startdaten ergibt sich mit (6.80) zu

$$\|\mathbf{Y}_n^{(0)} - \mathbf{Y}_n\| \leq (1 + hL)^n \|\mathbf{Y}_0^{(0)} - \mathbf{Y}_0\|.
 \tag{6.82}$$

Jetzt muss nur noch gesplittet und aufsummiert werden.

$$\begin{aligned}
 \|\mathbf{X}_n - \mathbf{Y}_n\| &\leq \|\mathbf{X}_n - \mathbf{Y}_n^{(n-1)} + \mathbf{Y}_n^{(n-1)} - \mathbf{Y}_n^{(n-2)} + \dots + \mathbf{Y}_n^{(1)} - \mathbf{Y}_n^{(0)} + \mathbf{Y}_n^{(0)} - \mathbf{Y}_n\| \\
 &\leq (1 + hL)^n \|\mathbf{Y}_0^{(0)} - \mathbf{Y}_0\| + ch^{p+1} \sum_{i=0}^{n-1} (1 + hL)^i \\
 &\leq e^{nhL} \|\mathbf{Y}_0^{(0)} - \mathbf{Y}_0\| + ch^{p+1} \sum_{i=0}^{n-1} e^{ihL} \\
 &\leq e^{nhL} \|\mathbf{Y}_0^{(0)} - \mathbf{Y}_0\| + ch^{p+1} \int_{s=0}^n e^{ihL} ds \\
 &\leq e^{nhL} \|\mathbf{Y}_0^{(0)} - \mathbf{Y}_0\| + \frac{ch^p}{L} (e^{nhL} - 1) \\
 &\leq \left(\|\mathbf{Y}_0^{(0)} - \mathbf{Y}_0\| + \frac{ch^p}{L} \right) e^{nhL}.
 \end{aligned}
 \tag{6.83}$$

Für Startdaten $\|\mathbf{Y}_0^{(0)} - \mathbf{Y}_0\| = \mathcal{O}(h^p)$ folgt die Konvergenz mit der Ordnung p . □

Die Konvergenztheorie ist auf den Fall konstanter Schrittweiten beschränkt. Die Erweiterung auf variable Schrittweiten ist möglich, verlangt aber erheblichen zusätzlichen Aufwand.

Wir kommen nun zu einer weiteren Fehleranalyse für explizite lineare MSV (6.56).

Wie früher bezeichne $x(t)$ die exakte Lösung des AWP

$$x' = f(t, x), \quad x(t_0) = x_0.$$

In Anlehnung an die Definition 6.2 mit $b_r = 0$ erhält man den lokalen Diskretisierungsfehler des expliziten MSV beim Einsetzen der Lösung $z(t) = \Phi^{t, t_i} \eta_i$ bzw. exakten Lösung $x(t) = x(t; t_0, x_0) = \Phi^{t, t_0} x_0$ in zwei Varianten. Dabei wurde eine kleine Indexverschiebung vorgenommen.

$$\begin{aligned} \tau(t_i, \eta_i, h) &= \frac{1}{h} [z(t_{i+r}) + a_{r-1} z(t_{i+r-1}) + \dots + a_0 z(t_i)] \\ &\quad - [b_{r-1} f(t_{i+r-1}, z(t_{i+r-1})) + \dots + b_0 f(t_i, z(t_i))] \\ &= \frac{1}{h} [z(t_{i+1}) + a_{r-1} z(t_i) + \dots + a_0 z(t_{i-r+1})] \\ &\quad - [b_{r-1} f(t_i, z(t_i)) + \dots + b_0 f(t_{i-r+1}, z(t_{i-r+1}))] \\ &= \frac{1}{h} \left(z_{i+1} + \sum_{k=0}^{r-1} a_k z_{i-r+1+k} \right) - \sum_{k=0}^{r-1} b_k f_{i-r+1+k} \\ &= \frac{1}{h} \left[z_{i+1} - \sum_{k=0}^{r-1} (-a_k z_{i-r+1+k} + h b_k f_{i-r+1+k}) \right], \\ \tilde{\tau}_{i+1} &= \frac{1}{h} \left[x(t_{i+1}) - \sum_{k=0}^{r-1} (-a_k x(t_{i-r+1+k}) + h b_k f(t_{i-r+1+k}, x_{i-r+1+k})) \right]. \end{aligned} \quad (6.84)$$

Satz 6.11 Für das explizite MSV (6.56) mit $b_r = 0$ gelte

$$a_k \leq 0, \quad k = 0, 1, \dots, r-1, \quad a_r = 1. \quad (6.85)$$

Es sei

$$e_i = \eta_i - x(t_i)$$

der Fehler an der Stelle $t_i = t_0 + ih$.

Definieren wir die Güte der Startwerte durch

$$G = \max_{0 \leq j \leq r-1} |\eta_j - x(t_j)|,$$

den maximalen lokalen Diskretisierungsfehler durch

$$D = \max_{j \geq r} |\tilde{\tau}_j|$$

und die Größe

$$B = \sum_{k=0}^{r-1} |b_k|$$

sowie die Lipschitz-Stetigkeit der rechten Seite $f(t, x)$ mit der Konstanten $L \geq 0$ als

$$|f(t, x_1) - f(t, x_2)| \leq L|x_1 - x_2|,$$

so gilt die Abschätzung

$$|e_i| \leq \left(G + \frac{D}{LB}\right) e^{LB(t_i - t_0)}. \quad (6.86)$$

Beweis.

Zuerst wollen wir erläutern, dass sich hinter den Beziehungen (6.85) das Stabilitätskriterium verbirgt.

Dazu notieren wir das Polynom $\rho(\xi)$ gemäß

$$\rho(\xi) = \sum_{k=0}^r a_k \xi^k = \xi^r - \sum_{k=0}^{r-1} \tilde{a}_k \xi^k, \quad \tilde{a}_k = -a_k \geq 0, \quad k = 0, 1, \dots, r-1.$$

Es gilt natürlich $1 = \sum_{k=0}^{r-1} \tilde{a}_k$ und $\rho(1) = 0$. Aber man kann generell zeigen, dass alle seine Nullstellen betragsmäßig ≤ 1 sind und solche vom Betrag Eins nicht doppelt auftreten. Dass $\xi = 1$ keine doppelte Nullstelle ist, macht man sich geometrisch plausibel.

Die Polynome ξ^r und $\sum_{k=0}^{r-1} \tilde{a}_k \xi^k$ sind für $\xi \geq 0$ nicht negativ, wachsend, konvex und haben in $\xi = 1$ einen "echten" Schnittpunkt.

Rechnerisch führt man den Nachweis mittels $\rho'(1) \neq 0$.

$$\begin{aligned} \rho'(\xi) &= r \xi^{r-1} - \sum_{k=1}^{r-1} k \tilde{a}_k \xi^{k-1}, \\ \rho'(1) &= r - \sum_{k=1}^{r-1} k \tilde{a}_k = \left(1 - \sum_{k=1}^{r-1} \tilde{a}_k\right) + \left(1 - \sum_{k=2}^{r-1} \tilde{a}_k\right) + \dots + (1 - \tilde{a}_{r-1}) + 1 \\ &= \tilde{a}_0 + (\tilde{a}_0 + \tilde{a}_1) + \dots + \sum_{k=0}^{r-2} \tilde{a}_k + 1 \geq 1. \end{aligned}$$

Zuerst zeigen wir, dass alle Nullstellen $|\xi_s| \leq 1$, $s = 1, 2, \dots, r$, sind, indem wir den Polynomwert außerhalb des Einheitskreises abschätzen.

Für $|\xi| > 1$ erhält man

$$\begin{aligned} |\rho(\xi)| &= \left| \xi^r - \sum_{k=0}^{r-1} \tilde{a}_k \xi^k \right| \geq |\xi^r| - \left| \sum_{k=0}^{r-1} \tilde{a}_k \xi^k \right| \geq |\xi^r| - \sum_{k=0}^{r-1} |\tilde{a}_k \xi^k| \geq |\xi^r| - \sum_{k=0}^{r-1} \tilde{a}_k |\xi^{r-1}| \\ &= |\xi^r| - |\xi^{r-1}| \sum_{k=0}^{r-1} \tilde{a}_k = |\xi^r| - |\xi^{r-1}| = |\xi^{r-1}| (|\xi| - 1) > 0. \end{aligned}$$

Damit liegen die Nullstellen ξ_s nicht außerhalb des Einheitskreises.

Es bleibt nachzuweisen, dass Nullstellen auf dem Einheitskreis nicht doppelt auftreten. Wenn man eine solche i. Allg. komplexe Nullstelle $\xi_s = e^{i\varphi}$, $|\xi_s| = 1$, hätte, dann ist auch $\bar{\xi}_s = e^{-i\varphi}$ eine Nullstelle.

Für $\xi = 1$ haben wir mit $\rho'(1) \neq 0$ den Nachweis schon gemacht und die allgemeine Vorgehensweise ist ähnlich.

Es genügt also, für $|\xi_s| = 1$ wiederum $\rho'(\xi_s) \neq 0$ zu zeigen.

$$\begin{aligned}\rho'(\xi_s) &= r \xi_s^{r-1} - \sum_{k=1}^{r-1} k \tilde{a}_k \xi_s^{k-1} = \xi_s^{-1} \left[r \xi_s^r - \sum_{k=1}^{r-1} k \tilde{a}_k \xi_s^k \right] \\ |\rho'(\xi_s)| &= \left| r \xi_s^r - \sum_{k=1}^{r-1} k \tilde{a}_k \xi_s^k \right| \geq |r \xi_s^r| - \left| \sum_{k=1}^{r-1} k \tilde{a}_k \xi_s^k \right| \geq r |\xi_s|^r - \sum_{k=1}^{r-1} k \tilde{a}_k |\xi_s|^k \\ &= r - \sum_{k=1}^{r-1} k \tilde{a}_k \geq 1\end{aligned}$$

analog zur Situation wie bei $\rho'(1)$.

Nach den Bemerkungen zur Voraussetzung (6.85) wenden wir uns nun dem Beweis und der Abschätzung (6.86) zu.

Aus den Formeln (6.56) ($j = i - r + 1$)

$$\begin{aligned}\sum_{k=0}^r a_k \eta_{i+k} &= h \sum_{k=0}^r b_k f_{i+k}, \quad a_r = 1, \quad b_r = 0, \\ \eta_{i+r} &= \sum_{k=0}^{r-1} (-a_k \eta_{i+k} + h b_k f_{i+k}), \\ \eta_{i+1} &= \sum_{k=0}^{r-1} (-a_k \eta_{j+k} + h b_k f_{j+k}) \quad (\text{Indexverschiebung})\end{aligned}$$

und (6.84)

$$x(t_{i+1}) = \sum_{k=0}^{r-1} [-a_k x(t_{j+k}) + h b_k f(t_{j+k}, x(t_{j+k}))] + h \tilde{\tau}_{i+1}$$

folgt durch Subtraktion, Anwendung der Dreiecksungleichung und Ausnutzung der Lipschitz-Stetigkeit von f für $i \geq r - 1$ die Beziehung

$$|e_{i+1}| = |\eta_{i+1} - x(t_{i+1})| \leq \sum_{k=0}^{r-1} (|a_k| + hL |b_k|) |e_{j+k}| + h |\tilde{\tau}_{i+1}|. \quad (6.87)$$

Für $k = 0, 1, \dots, r - 1$ definieren wir

$$C_k = |a_k| + hL |b_k| \geq 0.$$

Außerdem sei

$$C = \sum_{k=0}^{r-1} C_k.$$

Aus der Konsistenzbedingung (6.58) folgt $\sum_{k=0}^{r-1} a_k = -a_r = -1$ und damit

$$C = \sum_{k=0}^{r-1} |a_k| + hL \sum_{k=0}^{r-1} |b_k| = 1 + hLB \geq 1. \quad (6.88)$$

Eingesetzt in (6.87) ergeben sich hieraus

$$\begin{aligned} |e_r| &\leq \sum_{k=0}^{r-1} C_k |e_k| + hD \leq G \sum_{k=0}^{r-1} C_k + hD = GC + hD, \\ |e_{r+1}| &\leq \sum_{k=0}^{r-1} C_k |e_{k+1}| + hD \leq G \sum_{k=0}^{r-2} C_k + C_{r-1} |e_r| + hD \\ &\leq G \left(\sum_{k=0}^{r-2} C_k + C C_{r-1} \right) + hD (C_{r-1} + 1) \\ &= G \left(\sum_{k=0}^{r-1} C_k + C_{r-1} (C - 1) \right) + hD (C_{r-1} + 1) \\ &\leq G(C + C(C - 1)) + hD(C + 1) \\ &= GC^2 + hD(C + 1), \\ |e_{r+2}| &\leq \dots \leq GC^3 + hD(C^2 + C + 1) \end{aligned}$$

und allgemein durch Induktion

$$|e_{r+s}| \leq GC^{s+1} + hD \sum_{k=0}^s C^k, \quad (6.89)$$

bzw. mit $r \geq 1$ und $C \geq 1$ die gröbere Abschätzung

$$|e_{r+s}| \leq GC^{s+r} + hD \sum_{k=0}^{s+r-1} C^k. \quad (6.90)$$

Mit $i = r + s$ folgt wegen (6.88)

$$\begin{aligned} |e_i| &\leq GC^i + hD \sum_{k=0}^{i-1} C^k = GC^i + hD \frac{C^i - 1}{C - 1} \\ &= G(1 + hLB)^i + \frac{D}{LB} [(1 + hLB)^i - 1], \quad L > 0 \\ &\leq Ge^{ihLB} + \frac{D}{LB} (e^{ihLB} - 1) \\ &\leq \left(G + \frac{D}{LB} \right) e^{ihLB} = \left(G + \frac{D}{LB} \right) e^{LB(t_i - t_0)} \end{aligned}$$

die Behauptung des Satzes. □

Bemerkung 6.1

- (a) Die Adams-Bashforth-Formeln erfüllen die Voraussetzung (6.85) des Satzes 6.11.
 (b) Eine ähnliche Abschätzung wie die des Satzes 6.11 kann auch für allgemeinere und insbesondere für implizite MSV durchgeführt werden.
 (c) Einige Ungleichungen im Satz 6.11 stellen eine Vergrößerung der Abschätzung dar.
 (d) Um in jedem Fall die Konvergenz zu sichern, muss $D = \mathcal{O}(h^p)$, $p > 0$, und bzgl. der AB

$$\lim_{h \rightarrow 0} G = 0 \text{ bzw. } G \leq c_1 h^q, \quad q > 0, \quad c_1 \geq 0, \quad (6.91)$$

gelten.

(e) Falls $G \leq c_1 h^p$ ist, und insbesondere bei $G = 0$, überträgt sich die Konsistenzordnung p auf die Konvergenz und damit Genauigkeit der Lösung. Die Konvergenzordnung ist dann ebenfalls p .

(f) Sind die ersten r Werte $\eta_0, \eta_1, \dots, \eta_{r-1}$ exakt gegeben, d.h. ist $G = 0$, und hat das MSV die Konsistenzordnung p , so ist $D \leq c h^p$ und es folgt in Analogie zu Satz 5.5 für ESV die Abschätzung

$$|e_i| \leq \frac{c h^p}{LB} e^{LB(t_i - t_0)} = \mathcal{O}(h^p). \quad (6.92)$$

Sind die Startwerte nicht exakt bekannt, so müssen diese – um Verluste in der Ordnung zu vermeiden – mit einem Verfahren, z.B. ESV, der Ordnung p ermittelt werden.

(g) Tritt der Sonderfall $L = 0$ ein, so ist dann $C = 0$ sowie

$$\begin{aligned} |e_{r+s}| &\leq GC^{s+1} + hD \sum_{k=0}^s C^k = G + (s+1)hD \leq G + (s+r)hD, \\ |e_i| &\leq G + ihD = G + D(t_i - t_0). \end{aligned} \quad (6.93)$$

Das bedeutet für $G = \mathcal{O}(h^p)$ und $D = \mathcal{O}(h^p)$ die Konvergenzordnung p des MSV.

Das explizite zweistufige MSV $\eta_{i+1} + 4\eta_i - 5\eta_{i-1} = h(4f_i + 2f_{i-1})$ (vergl. Bsp. 6.6) erfüllt nicht das Stabilitätskriterium. Damit konvergiert es nicht und die gute Konsistenzordnung $\mathcal{O}(h^3)$ ist nicht relevant für die Konvergenz.

G. Dahlquist gab folgende Schranke für die maximal erreichbare Konvergenzordnung von k -stufigen MSV an.

Satz 6.12 Erste Dahlquist-Barriere

Ein konvergentes lineares k -stufiges MSV hat höchstens die Konvergenzordnung

$$p = \begin{cases} k, & \text{falls es explizit ist,} \\ k+1, & \text{falls es implizit und } k \text{ ungerade ist,} \\ k+2, & \text{falls es implizit und } k \text{ gerade ist.} \end{cases} \quad (6.94)$$

Beweis. [39]

Kapitel 7

Stabilität von Schrittverfahren

7.1 Absolute Stabilität

Während der Abschnitt 6.3 sich mit Instabilitäten befasste, welche durch die Eigendynamik der linken Seiten von MSV entstanden, wollen wir jetzt Instabilitäten untersuchen, welche mit den rechten Seiten der gDGL zusammenhängen. Die Untersuchungen beziehen sich dabei sowohl auf ESV als auch auf MSV.

Stabilitätsuntersuchungen zu Diskretisierungsverfahren setzen sinnvollerweise voraus, dass man das Verhalten der Lösung des AWP selber kennt bzw. beurteilen kann. Dazu braucht man nur auf einige bisher gerechnete AWP zu schauen.

Wenn man zum einfachen AWP $x' = x$, $x(0) = 1$, seine exponentiell wachsende Lösung $x(t) = e^t$ mit der Näherung aus einem Schrittverfahren vergleicht, so kann die Konvergenz bei $h \rightarrow 0$ nur in einem endlichen Intervall $[0, T]$ möglich sein, während bei Verfolgung der Näherungslösung ins unendliche stetig wachsende Abweichungen auftreten (siehe Beispiel 4.2 (a)).

Anders verhält sich die Situation beim AWP $x' = -x$, $x(0) = 1$, mit der Asymptotik seiner Lösung $x(t) = e^{-t}$ gegen Null. Das Schrittverfahren dazu sollte dann nicht nur konvergent sein, sondern auch die Asymptotik garantieren und wenn möglich auch "zwischen durch" noch gute Näherungswerte liefern. Für eine exakte Lösung, die monoton ist, ist eine oszillierende Näherungslösung kaum brauchbar. Wenn wie hier noch eine stabile Gleichgewichtslage $x(t) \equiv 0$ der gDGL $x' = -x$ mit der AB $x(0) = 0$ vorliegt, sollte das auch die Wahl eines günstigen Schrittverfahrens ermöglichen (siehe Beispiel 4.2 (b)).

Die Verfolgung von Lösungen, die asymptotisch stabil sind oder gegen stabile Gleichgewichtslagen tendieren, ist mit Schrittverfahren bei hinreichend kleiner Schrittweite h oder für lange Integrationsintervalle (dabei kann "unterwegs" h durchaus vergrößert werden) eher möglich als die Approximation von instabilen Gleichgewichtslagen.

Gerade Näherungsverfahren können bei Rechnungen in der GKA, die damit Rundungseffekten und verschiedenen Störungen ausgesetzt sind, Lösungen erzeugen, die sich schnell von der exakten Trajektorie entfernen.

Für die gDGL $x' = t \cos(x)$ in den Beispielen 5.8 (a) bzw. 5.10 (b) erkennen wir mit $x(t) \equiv \frac{\pi}{2}$ eine sehr ausgeprägte stabile Gleichgewichtslage.

Zur gDGL $x' = x(1 - x)$ mit ihrer exakten Lösung $x(t) = Ce^t/(1 + C(e^t - 1))$ haben wir das Richtungsfeld in Abb. 4.1 dargestellt. Es gibt zwei Gleichgewichtslagen $x(t) \equiv 0$ und $x(t) \equiv 1$. Sobald im AWP $x' = x(1 - x)$, $x(0) = 0$, die AB einer noch so kleinen Störung δ unterliegt, tendiert die Lösungstrajektorie entweder bei $\delta < 0$ gegen $-\infty$ oder bei $\delta > 0$ gegen 1.

Für das AWP $x' = 10(x - t^2/(1 + t^2)) + 2t/(1 + t^2)^2$, $x(0) = 0$, im Beispiel 5.10 (c) ist die Lösungstrajektorie $x(t) = t^2/(1 + t^2)$ eine instabile Gleichgewichtslage. Nur in einem endlichen Zeitintervall $[0, T]$ kann ein stabiles Schrittverfahren konvergieren. Für Rechnungen bei $T \gg 1$ wird es immer Probleme mit der Approximation geben.

Zusätzliche Aspekte sind zu berücksichtigen,

- wenn das Richtungsfeld sehr steile Anstiege enthält wie bei $x' = 1/(x + 10^{-p})$, $p \gg 1$ (vergl. Beispiel 5.8 (b)),
- wenn wie für die gDGL $x' = x^2$ nicht nur eine globale instabile Gleichgewichtslage $x(t) \equiv 0$ vorliegt, sondern mit der AB $x(0) = 1/c > 0$ die Lösung $x(t) = 1/(c - t)$ bei $t = c$ "explodiert" und gegen $+\infty$ strebt (vergl. Bemerkung 4.2 (b)),
- wenn das Lösungsintervall wie bei $x' = -1/\sqrt{x}$, $x(0) = 1$, eingeschränkt ist und die Lösung $x(t) = (1 - \frac{3}{2}t)^{2/3}$ bei $t = \frac{2}{3}$ kollabiert (vergl. Bemerkung 4.2 (d)),
- wenn die Lösung wie für $x' = \sin(1/x)$, $x(0) = 1$, an bestimmten Punkten nicht definiert ist, aber ein Schrittverfahren über diese Stellen hinweg integriert (vergl. Bemerkung 4.2 (e)).

Wir erkennen aus den Betrachtungen, dass AWP mit fallenden Lösungen für Stabilitätsuntersuchungen eher geeignet sind.

Deshalb begnügt man sich in der ersten Phase zunächst mit einfachen Test-DGL.

Im skalaren Fall ist das die parameterabhängige gDGL (AWP)

$$x' = \lambda x, \quad t \geq t_0, \quad x(t_0) = x_0 \in \mathbb{R}, \quad \lambda \in \mathbb{C}, \quad (7.1)$$

im vektoriellen Fall das SysgDGL

$$\mathbf{x}' = A \mathbf{x}, \quad t \geq t_0, \quad \mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbb{R}^n, \quad \text{Matrix } A \text{ reell, regulär, diagonalisierbar.} \quad (7.2)$$

Mit der Lösung $x(t) = x_0 e^{\lambda(t-t_0)}$ der Test-DGL (7.1) sieht man, dass der Realteil $\Re \lambda$ darüber entscheidet, ob für wachsendes t der Betrag $|x(t)|$ fällt oder wächst.

Definition 7.1

(a) Die Test-DGL (7.1) heißt stabil, falls $\Re \lambda \leq 0$.

(b) Die Test-DGL (7.1) heißt asymptotisch stabil, falls sie stabil ist und bei Störung der AB $\tilde{x}(t_0) = x_0 + \Delta x_0$ die Lösung $\tilde{x}(t)$ des gestörten AWP die Bedingung

$$\lim_{t \rightarrow \infty} (\tilde{x}(t) - x(t)) = 0 \quad (7.3)$$

erfüllt. Hier bedeutet das wegen $\tilde{x}(t) - x(t) = \Delta x_0 e^{\lambda(t-t_0)}$ die Beziehung $\Re \lambda < 0$. Deshalb wird λ aus der negativen komplexen Halbebene \mathbb{C}^- genommen.

Natürlich wird man von Diskretisierungsverfahren adäquates Wachstumsverhalten verlangen bzw. untersuchen, welche Bedingungen dafür zu erfüllen sind.

Beispiel 7.1 Betrachten wir das lineare Testproblem

$$x' = \lambda x, \quad x(0) = 1, \quad \lambda \in \mathbb{C}, \quad (7.4)$$

mit der exakten Lösung $x(t) = e^{\lambda t}$ und wenden hierauf das KRKV an. Wir erhalten

$$\begin{aligned} k_1 &= \lambda \eta_i, \\ k_2 &= \lambda \left(\eta_i + \frac{1}{2} h k_1 \right) = \left(\lambda + \frac{1}{2} h \lambda^2 \right) \eta_i, \\ k_3 &= \lambda \left(\eta_i + \frac{1}{2} h k_2 \right) = \left(\lambda + \frac{1}{2} h \lambda^2 + \frac{1}{4} h^2 \lambda^3 \right) \eta_i, \\ k_4 &= \lambda \left(\eta_i + h k_3 \right) = \left(\lambda + h \lambda^2 + \frac{1}{2} h^2 \lambda^3 + \frac{1}{4} h^3 \lambda^4 \right) \eta_i, \end{aligned}$$

und damit

$$\eta_{i+1} = \eta_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) = F(h\lambda) \eta_i \quad (7.5)$$

mit dem Polynom bzw. der Wachstums- oder Übergangsfunktion von $i \rightarrow (i+1)$

$$F(z) = 1 + z + \frac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4, \quad z = h\lambda \in \mathbb{C}. \quad (7.6)$$

Ist der Realteil $\Re \lambda < 0$, so klingt unabhängig von seinem Imaginärteil die exakte Lösung $x(t)$ für $t \rightarrow \infty$ gegen Null ab. Dasselbe qualitative Verhalten für die numerische Lösung η_i erhalten wir nur, wenn $h > 0$ so gewählt ist, dass

$$|F(z)| < 1. \quad (7.7)$$

Wegen $|F(z)| \rightarrow \infty$ für $|z| \rightarrow \infty$ (allgemeine Eigenschaft eines Polynoms) ist das aber nur möglich, wenn $|z|$ und damit die Schrittweite h hinreichend klein ist.

7.1.1 Stabilität und Stabilitätsgebiet von Einschrittverfahren

Das Beispiel 7.1 motiviert die folgende Definition.

Definition 7.2 (a) Führt ein ESV für das Problem (7.1) auf eine Rekursion der Form

$$\eta_{i+1} = F(z) \eta_i, \quad z = h\lambda, \quad (7.8)$$

so ist die Menge

$$\mathcal{E} = \{z \in \mathbb{C} : |F(z)| < 1\} \quad (7.9)$$

das zum ESV gehörende Gebiet der absoluten Stabilität.

Wegen $F(0) = 1$ liegt $z = 0$ immer auf der Grenze $\partial \mathcal{E}$ des Stabilitätsgebiets \mathcal{E} .

Die Stabilität von ESV korrespondiert also mit dem günstigen Verhalten der Wachstumsfunktion $F(z)$.

Der Grenzfall $\lambda = 0$ kann unter verschiedenen Gesichtspunkten gut oder schlecht sein. Zur exakten Lösung $x(t) \equiv 1$ ist die Näherung $\eta_i = 1$, $i = 0, 1, \dots$, natürlich in Ordnung. Ist die AB jedoch gestört, also $\eta_0 = 1 + \delta$, dann gilt auch $\eta_i = 1 + \delta$, $i = 1, 2, \dots$, und wegen $\lim_{h \rightarrow 0} |\eta_i - x(t_i)| = \delta \neq 0$ hat man keine Konvergenz.

(b) Liegt die linke komplexe Halbebene

$$\mathbb{C}^- = \{z \in \mathbb{C} : \Re z < 0\} \quad (7.10)$$

ganz in \mathcal{E} , so heißt das Verfahren A-stabil.

(c) Liegt die linke komplexe Halbebene \mathbb{C}^- bzw. ein genügend großer unbeschränkter Teil von \mathbb{C}^- im Stabilitätsgebiet \mathcal{E} , dann heißt das Verfahren unbeschränkt absolut stabil. Zuweilen wird diese letzte etwas schwächere Bedingung auch als A-Stabilität verwendet.

(d) Ist zudem $\lim_{\Re \lambda \rightarrow -\infty} |F(z)| < 1$, so heißt das Verfahren stark stabil.

(e) A-stabile Verfahren mit der zusätzlichen Eigenschaft $\lim_{\Re \lambda \rightarrow -\infty} F(z) = 0$ werden als L-stabil bezeichnet.

Entsprechend dem Beispiel 7.1 sind Schrittweiten $h > 0$ für das Testproblem so zu wählen, dass $h\lambda$ für gegebenes λ mit $\Re \lambda < 0$ im Gebiet der absoluten Stabilität liegt. Für A-stabile Verfahren gibt es keine Schrittweitenbeschränkung.

Weniger aufwändig als die Betrachtung von Gebieten in der komplexen Ebene ist natürlich die im Reellen mit $\lambda \in \mathbb{R}$ und $\lambda < 0$.

Beispiel 7.2

(a) Für das PZV ist

$$\eta_{i+1} = \eta_i + hf(\eta_i) = (1 + h\lambda) \eta_i, \quad i = 0, 1, \dots, \eta_0 \text{ gegeben,} \quad (7.11)$$

also die Wachstumsfunktion

$$F(z) = 1 + z. \quad (7.12)$$

Das Gebiet \mathcal{E} der absoluten Stabilität ist in der komplexen Ebene (Gaußsche Zahlenebene) wegen $1 = |1 + z| = |z - (-1)|$ das Innere des Kreises um den Mittelpunkt $z_0 = -1$ und mit dem Radius $r = 1$.

Zu reellem $\lambda < 0$ ist die Herleitung einer Bedingung für h sehr einfach, denn es gilt

$$\begin{aligned} |1 + h\lambda| &< 1, \\ -1 &< 1 + h\lambda < 1, \\ -h\lambda &< 2, \\ h &< \frac{2}{-\lambda} = \frac{2}{|\lambda|} \end{aligned}$$

(vergl. Beispiel 4.2).

Zu komplexem $\lambda = s + i\omega$ mit $\Re\lambda = s < 0$ kann man zwei Abschätzungen vornehmen.

$$|h\lambda| - 1 \leq ||h\lambda| - 1| \leq |1 + h\lambda| < 1,$$

$$|h\lambda| < 2, \quad h < \frac{2}{|\lambda|},$$

$$1 - |h\lambda| \leq |1 - |h\lambda|| \leq |1 + h\lambda| < 1, \quad \text{falls } |h\lambda| < 1, \quad \text{d.h. } h < \frac{1}{|\lambda|}.$$

Die erste Ungleichung $h < \frac{2}{|\lambda|}$ ist relativ grob und stellt damit nur eine notwendige Bedingung an h dar. Der zweite Zugang liefert eine Ungleichung, die i. Allg. eine zu strenge Forderung an h enthält.

Günstiger ist es, das Ganze im Komplexen zu betrachten. Dann erhält man die Beziehungen

$$\begin{aligned} |1 + h\lambda| &= |1 + h(s + i\omega)| = |1 + hs + ih\omega| < 1, \\ (1 + hs)^2 + (h\omega)^2 &= 1 + 2hs + h^2s^2 + h^2\omega^2 < 1, \\ 2s + h(s^2 + \omega^2) &< 0, \\ h|\lambda|^2 &< -2 \Re\lambda, \\ h < \frac{2 |\Re\lambda|}{|\lambda|^2} &= -\frac{2s}{s^2 + \omega^2}. \end{aligned} \quad (7.13)$$

Die reelle Version ist als Sonderfall enthalten.

Betrachten wir die Schranke als Schrittweitenfunktion $h = h(\lambda) = h(s, \omega) = -\frac{2s}{s^2 + \omega^2}$ etwas genauer, indem wir eine der unabhängigen Variablen konstant lassen.

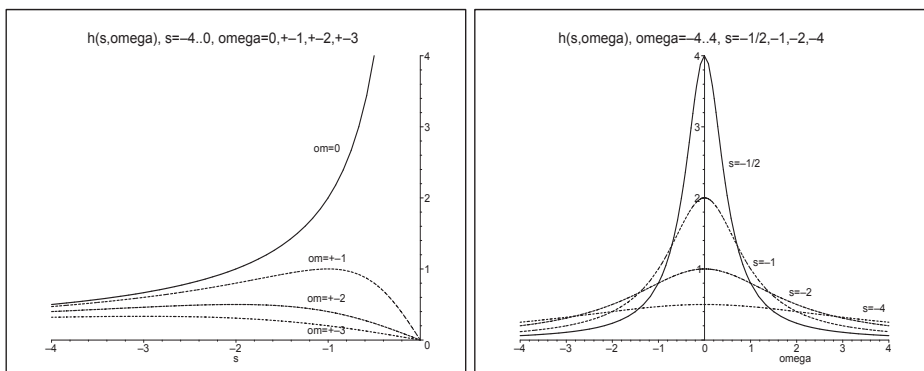


Abb. 7.1 Dateien *schr01.ps*, *schr02.ps*, Schrittweitenfunktion $h(s, \omega)$

Wir erkennen folgende Eigenschaften der Funktion $h(s, \omega)$.

- Wenn $s \ll -1$, dann ist die Kurve $h(s, 0)$ als obere Schranke für die maximal mögliche Schrittweite zuständig. ω^2 ist vernachlässigbar und $h < -\frac{2s}{s^2} = -\frac{2}{s} \approx \frac{2}{|\lambda|}$.

- Ist $s \in [-2, 0)$, dann hat ω wachsenden Einfluss auf die Schranke h . Diese bleibt klein, falls $\omega \rightarrow \pm\infty$ tendiert. Sie kann aber auch sehr groß werden bei $\omega = 0$ bzw. ω nahe Null (wenig Imaginärteil).
- Falls $s \ll 0$ ist, dann gibt es eine starke Einschränkung für h bei allen ω .
- Falls $\omega \approx 0$ gilt, dann ist mit $s \approx 0$ eine große Schrittweite h möglich.

Die Berechnung der begrenzenden Kontur $\partial\mathcal{E} = \{z : |F(z)| = 1\}$ des Stabilitätsgebiets \mathcal{E} kann hier einfach durch den Ansatz $F(z) = e^{i\varphi}$, $\varphi \in \mathbb{R}$, erfolgen. Damit erhält man

$$z = -1 + e^{i\varphi} = -1 + \cos(\varphi) + i \sin(\varphi), \quad \varphi \in [0, 2\pi),$$

was genau den Kreis mit dem Radius 1 um den Mittelpunkt $z_0 = -1$ beschreibt. Im Inneren dieses Kreises bzw. des Intervalls $[-2, 0]$ ist $|F(z)| < 1$.

Zur Schrittweitenwahl $h(\lambda)$, $\lambda \in \mathbb{C}^-$, betrachten wir das Testproblem

$$x' = \lambda x, \quad x(0) = 1, \quad \lambda = -1 + i,$$

mit der exakten komplexen Lösung

$$\begin{aligned} x(t) &= e^{\lambda t} = e^{-t}(\cos(t) + i \sin(t)) = \Re x(t) + i \Im x(t) = x_1(t) + i x_2(t), \\ x_1(t) &= e^{-t} \cos(t), \quad x_1(0) = 1, \quad x_2(t) = e^{-t} \sin(t), \quad x_2(0) = 0. \end{aligned}$$

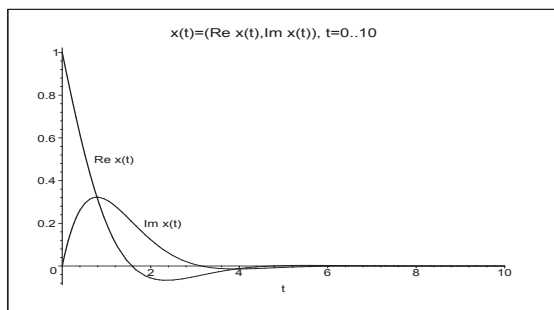


Abb. 7.2

Datei *komp1.ps*,

Lösung als Real- und Imaginärteil

$$\begin{aligned} \Re x(t) &= x_1(t) = e^{-t} \cos(t), \quad x_1(0) = 1, \\ \Im x(t) &= x_2(t) = e^{-t} \sin(t), \quad x_2(0) = 0, \\ t &\in [0, 10] \end{aligned}$$

Berechnungen in Maple

```
> assume(t,real):
lambda:=-1+I; dgl:=diff(x(t),t)-lambda*x(t); dsolve(dgl,x(t));
> loes:=dsolve({dgl,x(0)=1},x(t));
x1:=unapply(Re(rhs(loes)),t);
x2:=unapply(Im(rhs(loes)),t);
```

$$\lambda := -1 + I$$

$$dgl := \left(\frac{d}{dt} x(t) \right) + (1 - I)x(t)$$

$$x(t) = C1 e^{(-1+I)t}$$

$$loes := x(t) = e^{(-1+I)t},$$

$$x1 := t \rightarrow e^{(-t)} \cos(t), \quad x2 := t \rightarrow e^{(-t)} \sin(t)$$

Wollen wir Richtungsfelder zur gDGL wie in Abschnitt 4.1 erzeugen, so scheitern wir am komplexen Parameter.

```
> v:=[1,lambda*y];
   fieldplot(v,t=0..2,x=-1..2);
     # Error, (in fieldplot) no non-zero vectors found
   dfieldplot({dgl=0},[x],t=0..2,x=-1..2);
     # Koordinatensystem ohne Grafik
   DEplot({dgl},[x(t)],t=0..2,{[0,1]},linecolor=black,axes=frame);
     # Error, (in DEtools/DEplot/drawlines) complex argument to max/min
```

Es bleibt also nur die Umwandlung der komplexen gDGL

$$(x_1(t) + i x_2(t))' = (-1 + i)(x_1(t) + i x_2(t)) \quad \text{mit} \quad x_1(t) = \Re x(t), \quad x_2(t) = \Im x(t),$$

in ein System von zwei gDGL 1. Ordnung

$$x_1' + x_1 + x_2 = 0, \quad x_1(0) = 1,$$

$$x_2' - x_1 + x_2 = 0, \quad x_2(0) = 0.$$

```
> dgl:=diff(x1(t),t)+x1(t)+x2(t)=0,diff(x2(t),t)-x1(t)+x2(t)=0;
   vars:=x1(t),x2(t):
   dsolve({dgl},{vars});
```

$$dgl := \left(\frac{d}{dt} x_1(t) \right) + x_1(t) + x_2(t) = 0, \quad \left(\frac{d}{dt} x_2(t) \right) - x_1(t) + x_2(t) = 0$$

$$\{x_1(t) = e^{(-t)}(-C_1 \sin(t) + C_2 \cos(t)), \quad x_2(t) = e^{(-t)}(-C_1 \cos(t) + C_2 \sin(t))\}$$

```
> inits:=x1(0)=1,x2(0)=0;
   loes2:=dsolve({dgl,inits},{vars});
   x1t:=rhs(op(1,loes2));
   x2t:=rhs(op(2,loes2));
```

$$inits := x_1(0) = 1, \quad x_2(0) = 0$$

$$loes2 := \{x_1(t) = e^{(-t)} \cos(t), \quad x_2(t) = e^{(-t)} \sin(t)\}$$

$$x1t := e^{(-t)} \cos(t)$$

$$x2t := e^{(-t)} \sin(t)$$

Aber man kann die Lösung $x(t)$ als Phasenkurve $(x_1(t), x_2(t))$, $t \in [0, T]$, zeichnen bzw. als Phasenportrait, d.h. die Phasenkurve gemeinsam mit dem Richtungsfeld $(x_1'(t), x_2'(t))$ in dem vom Integrationsintervall $t \in [0, T]$ definierten und aufgespannten rechteckigen Bereich $[\min x_1(t), \max x_1(t)] \times [\min x_2(t), \max x_2(t)]$.

Wenn das SysgDGL nicht autonom ist, erhält man nur die Kurve. Im autonomen Fall Kurve und Richtungsfeld.

```

> p1:=plot([x1t,x2t,t=0..10],thickness=3,labels=['Re x(t)', 'Im x(t)'],
           color=black,title='Kurve (x1(t),x2(t))=(Re x(t),Im x(t)), t=0..10'):
display(p1);

> p1:=DEplot({dgl1}, [vars], t=0..10, [[x1(0)=1,x2(0)=0]], color=red,
            thickness=1,stepsize=0.01,
            title='Phasenportrait: x1(t)=Re x(t),x2(t)=Im x(t), t=0..10'):
p11:=plot([x1t,x2t,t=0..10],color=black,thickness=3):
display(p11,p1);

```

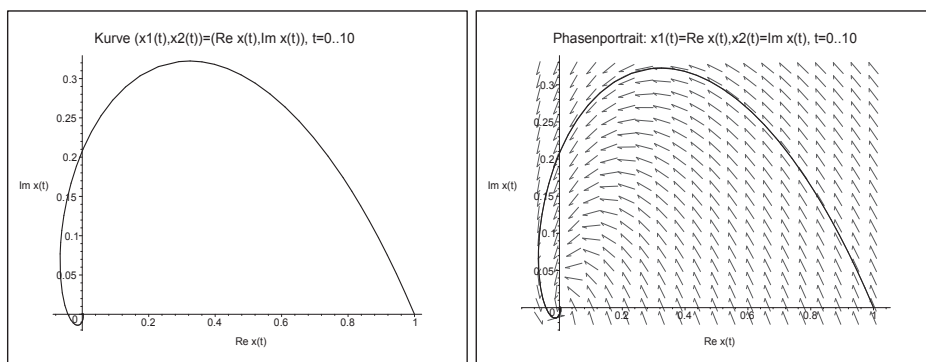


Abb. 7.3 Dateien *komp2.ps*, *komp3.ps*,

Phasenkurve $(x_1(t), x_2(t)) = (\Re x(t), \Im x(t)) = (e^{-t} \cos(t), e^{-t} \sin(t))$, $t \in [0, T]$,
 $(x_1(0), x_2(0)) = (1, 0)$, sowie gemeinsam mit Richtungsfeld $(x'_1(t), x'_2(t))$

Nun wenden wir das PZV (7.11) im Komplexen für $\eta_0 = 1 + i0$ und verschiedene Schrittweiten h an. Zu den komplexen Näherungswerten η_i haben wir ihre Real- und Imaginärteile, die wir sowohl als Punkte als auch durch einen die Punkte verbindenden Polygonzug zeichnen und mit der exakten Lösungskurve vergleichen.

Zwecks Stabilität ist für $\lambda = -1 + i$, $|\lambda| = \sqrt{2}$, die Schrittweitenbedingung (7.13) einzuhalten, d.h. $h < 2|\Re \lambda|/|\lambda|^2 = 1$. Die anderen Abschätzungen wie $h < 2/|\lambda| = \sqrt{2}$ sind diesbezüglich zu grob, oder $h < 1/|\lambda| = \frac{\sqrt{2}}{2}$ zu scharf.

Will man jedoch nicht oszillierendes Lösungsverhalten und vernünftige Genauigkeit erzielen, muss man mit h deutlich unter der Stabilitätsgrenze 1 bleiben.

```

> veta:=array(1..2,0..1000, []):
> h:=1/10; N:=100; T:=N*h;
eta0:=1+I*0;
veta[1,0]:=Re(eta0); veta[2,0]:=Im(eta0);
etai:=eta0;
for i from 1 to N do
  etai1:=(1+h*lambda)*etai;
  veta[1,i]:=Re(etai1); veta[2,i]:=Im(etai1);
  etai:=etai1;
end do;

```

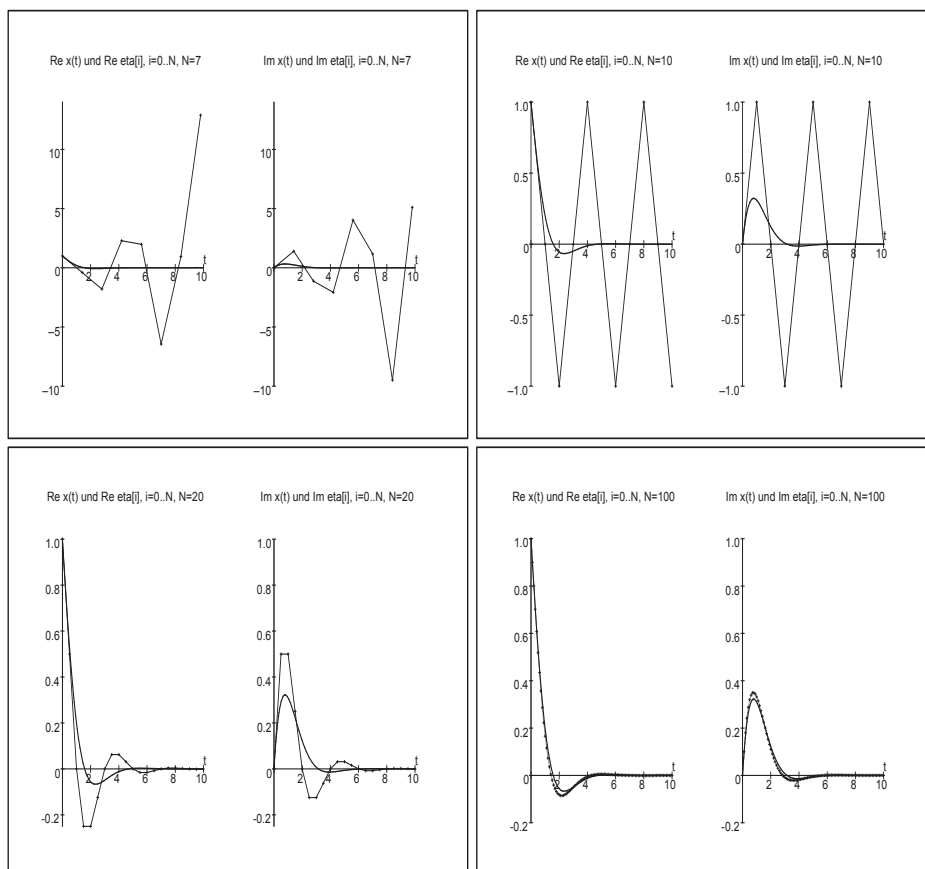



Abb. 7.4 Dateien *kpzv1.ps*, *kpzv2.ps*, *kpzv3.ps*, *kpzv4.ps*,
 Näherungswerte $\Re\eta_i$, $\Im\eta_i$, $i = 0, 1, \dots, N$, des PZV bei $h = 1.4, 1, \frac{1}{2}, \frac{1}{10}$,
 sowie exakte Lösungskurven $x_1(t)$, $x_2(t)$, $t \in [0, T]$, $T = 10$

```

> pp:=array(1..2, []):
> p1:=plot(x1(t),t=0..T,color=blue,thickness=2,
           title='Re x(t) und Re eta[i], i=0..N, N=100'):
  p2:=pointplot([seq([i*h,veta[1,i]],i=0..N)],style=LINE,color=red):
  p3:=pointplot([seq([i*h,veta[1,i]],i=0..N)],color=black):
  p4:=plot([[0,-0.2],[0,1]],color=black):
  pp[1]:=display(p1,p2,p3,p4):
> p1:=plot(x2(t),t=0..T,color=blue,thickness=2,
           title='Im x(t) und Im eta[i], i=0..N, N=100'):
  p2:=pointplot([seq([i*h,veta[2,i]],i=0..N)],style=LINE,color=red):
  p3:=pointplot([seq([i*h,veta[2,i]],i=0..N)],color=black):
  p4:=plot([[0,-0.2],[0,1]],color=black):
  pp[2]:=display(p1,p2,p3,p4):

> display(pp);

```

```
> datei:='C:/D/neundorf/nwptexte/num_iii_gdgl/neundorf06/maple/kpzv4.ps':
interface(plotdevice=ps,plotoutput=datei,
           plotoptions='color,portrait,width=640,height=480');
plots[display](pp);
interface(plotdevice=win);
```

(b) Für das IEV

$$\begin{aligned}\eta_{i+1} &= \eta_i + hf(t_{i+1}, \eta_{i+1}) = \eta_i + h\lambda \eta_{i+1}, \\ \eta_{i+1} &= \frac{1}{1 - h\lambda} \eta_i\end{aligned}\quad (7.14)$$

ist die Wachstumsfunktion

$$F(z) = \frac{1}{1 - z}.\quad (7.15)$$

Für $\Re\lambda < 0$ gilt $\Re z < 0$ und weiter mit $z = u + iv$, $u < 0$,

$$|F(z)| = \frac{1}{|1 - z|} = \frac{1}{|1 - u - iv|} = \frac{1}{\sqrt{(1 - u)^2 + v^2}} \leq \frac{1}{\sqrt{(1 - u)^2}} = \frac{1}{1 + |\Re z|} < 1.$$

Das Verfahren ist also A-stabil.

Wenn man in einem solchen Fall einen Stabilitätsbereich angeben will, so ist seine Begrenzung nicht im Gebiet \mathbb{C}^- zu suchen.

Es gilt $|F(z)| = 1$ für den Kreis um den Mittelpunkt $z_0 = 1$ und mit dem Radius 1, d.h. $z = 1 + e^{i\varphi}$, und außerhalb dieses Kreises sowie insbesondere in \mathbb{C}^- ist $|F(z)| < 1$.

(c) Wir kombinieren die Verfahren (a) und (b) zu einem PCV und definieren

$$\begin{aligned}\eta_{i+1}^{(P)} &= \eta_i + hf(t_i, \eta_i), \\ \eta_{i+1} &= \eta_i + hf(t_{i+1}, \eta_{i+1}^{(P)}).\end{aligned}$$

Das Einsetzen von $f(t, x) = \lambda x$ ergibt

$$\eta_{i+1} = (1 + h\lambda + (h\lambda)^2) \eta_i\quad (7.16)$$

und damit

$$F(z) = 1 + z + z^2.\quad (7.17)$$

Wegen $|F(z)| \rightarrow \infty$ für $|z| \rightarrow \infty$ ist das Verfahren nicht A-stabil. Es "erbt" einen Teil seiner Eigenschaften eben auch vom Prädiktor. Das PCV hat ein Stabilitätsgebiet, das sich von denen seiner beiden Anteile unterscheidet und wider Erwarten zu einer strengeren Forderung bzgl. der Schrittweite h führt.

Einige Varianten der Ermittlung und Darstellung des Stabilitätsgebiets mit Maple

```
> s:=solve(1+z+z^2-exp(I*phi)=0,z);
```

$$s := -\frac{1}{2} + \frac{1}{2} \sqrt{-3 + 4e^{(I\phi)}}, -\frac{1}{2} - \frac{1}{2} \sqrt{-3 + 4e^{(I\phi)}}$$

```
> s1:=s[1];
s2:=s[2];
```

$$s1 := -\frac{1}{2} + \frac{1}{2} \sqrt{-3 + 4e^{(I\phi)}}$$

$$s2 := -\frac{1}{2} - \frac{1}{2} \sqrt{-3 + 4e^{(I\phi)}}$$

```
> # Test
Re(s1)
evalc(Re(s1));
```

$$-\frac{1}{2} + \frac{1}{2} \Re(\sqrt{-3 + 4e^{(I\phi)}})$$

$$-\frac{1}{2} + \frac{1}{4} \sqrt{2 \sqrt{(-3 + 4 \cos(\phi))^2 + 16 \sin(\phi)^2} - 6 + 8 \cos(\phi)}$$

```
> x1:=phi->evalc(Re(s1)); # nicht Operator -> verwenden
x1:=unapply(evalc(Re(s1)),phi);
x2:=unapply(evalc(Im(s1)),phi);
```

$$x1 := \phi \rightarrow \text{evalc}(\Re(s1))$$

$$x1 := \phi \rightarrow -\frac{1}{2} + \frac{1}{4} \sqrt{2 \sqrt{(-3 + 4 \cos(\phi))^2 + 16 \sin(\phi)^2} - 6 + 8 \cos(\phi)}$$

$$x2 := \phi \rightarrow \frac{1}{4} \text{csgn}(4 \sin(\phi) + 3I - 4I \cos(\phi)) \sqrt{2 \sqrt{(-3 + 4 \cos(\phi))^2 + 16 \sin(\phi)^2} - 6 + 8 \cos(\phi)}$$

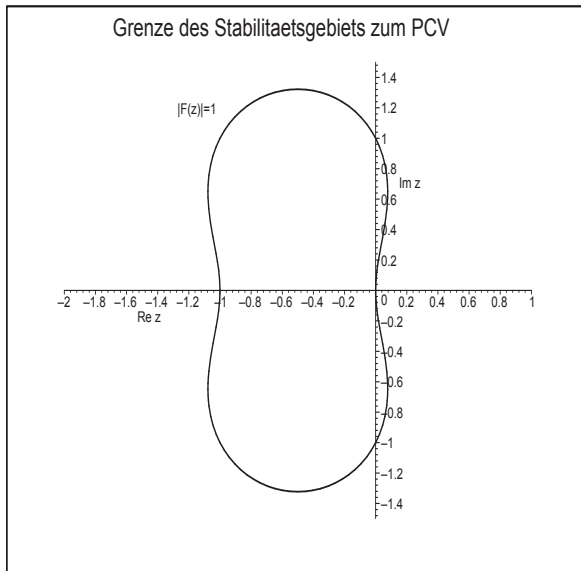
```
> y1:=unapply(evalc(Re(s2)),phi);
y2:=unapply(evalc(Im(s2)),phi);
```

$$y1 := \phi \rightarrow -\frac{1}{2} - \frac{1}{4} \sqrt{2 \sqrt{(-3 + 4 \cos(\phi))^2 + 16 \sin(\phi)^2} - 6 + 8 \cos(\phi)}$$

$$y2 := \phi \rightarrow -\frac{1}{4} \text{csgn}(4 \sin(\phi) + 3I - 4I \cos(\phi)) \sqrt{2 \sqrt{(-3 + 4 \cos(\phi))^2 + 16 \sin(\phi)^2} - 6 + 8 \cos(\phi)}$$

```
> p1:=plot([x1(t),x2(t),t=-Pi+0.001..Pi-0.001],view=[-2..1,-1.5..1.5],
thickness=2,labels=['Re z','Im z'],
title='Grenze des Stabilitaetsgebiets zum PCV'):
p2:=plot([y1(t),y2(t),t=-Pi+0.001..Pi-0.001],thickness=2):
p3:=textplot([-1.2,1.2,'|F(z)|=1']):
display(p1,p2,p3);
```

```
> datei1:='D:/neundorf/nwptexte/num_gdgl/neun06/maple/stab_01.ps':
interface(plotdevice=ps,
plotoutput=datei1,
plotoptions='color,portrait,width=640,height=480');
plots[display](p1,p2,p3);
interface(plotdevice=win);
```

**Abb. 7.5**

Datei *stab_01.ps*,
 Begrenzungskurve $|F(z)| = 1$
 des Stabilitätsgebiets \mathcal{E}
 zum PCV:
 PZV-Prädiktor und
 IEV-Korrektor

```
> # Weitere Varianten der Konturbestimmung |F(z)|=1 in 3D, 2D
f:=unapply(abs((x+I*y)^2+x+I*y+1),x,y);
```

$$f := (x, y) \rightarrow |(x + Iy)^2 + x + Iy + 1|$$

```
> plot3d(f(x,y),x=-2..1,y=-1.5..1.5,scaling=constrained,orientation=[-90,0],
axes=normal,style=contour,contours=[1],thickness=2);
```

```
> contourplot(f(x,y),x=-2..1,y=-1.5..1.5,contours=[1],
view=[-2..1,-1.5..1.5],thickness=2);
```

Zu reellem $\lambda < 0$ ist die Herleitung einer Bedingung für die Schrittweite h sehr einfach, denn aus $|1 + h\lambda + (h\lambda)^2| < 1$ folgt $h < \frac{1}{-\lambda} = \frac{1}{|\lambda|}$. Weder die schwächere Bedingung $h < \frac{2}{|\lambda|}$ des PZV noch die A-Stabilität des IEV bleiben erhalten.

(d) Für die (2,2)-RKV, zu denen u.a. das MPVZ und HV gehören, ist

$$F(z) = 1 + z + \frac{1}{2}z^2. \quad (7.18)$$

Zu reellem $\lambda < 0$ folgt die Bedingung für die Schrittweite h aus $|1 + h\lambda + \frac{1}{2}(h\lambda)^2| < 1$. Wegen

$$-2 < 2F(h\lambda) = 2 + 2h\lambda + (h\lambda)^2 = 1 + (1 + h\lambda)^2 < 2$$

ergeben sich als Grenzen des Stabilitätsintervalls $h\lambda = 0$ und $h\lambda = -2$ und somit wie beim PZV die Schrittweitenbedingung $h < \frac{2}{|\lambda|}$.

Die Berechnung der begrenzenden Kontur $\partial\mathcal{E} = \{z : |F(z)| = 1\}$ des Stabilitätsgebiets \mathcal{E} kann wieder einfach durch den Ansatz $F(z) = e^{i\varphi}$, $\varphi \in \mathbb{R}$, erfolgen. Damit ist

$$z = z(\varphi) = -1 \pm \sqrt{2e^{i\varphi} - 1}, \quad \varphi \in [-\pi, \pi]. \quad (7.19)$$

Die beiden Teilkonturen ergeben zusammen die Gebietsgrenze $\partial\mathcal{E}$, die einer Ellipse ähnlich ist (siehe Abb. 7.8).

Bemerkung 7.1

(a) Für explizite ESV bzw. für PCV ist $F(z)$ i. Allg. ein Polynom. Da Polynome für $|z| \rightarrow \infty$ unbeschränkt wachsen, sind solche Verfahren nicht A-stabil. Damit ist z.B. das KRKV nicht A-stabil. Für implizite Verfahren ist $F(z)$ i. Allg. eine gebrochene rationale Funktion, wo das Stabilitätsgebiet dann meistens sich vergrößern kann.

(b) Für $F(z)$ als Polynom bzw. gebrochene rationale Funktion ist das Stabilitätsgebiet symmetrisch zur reellen Achse, denn es gilt im Komplexen

$$z \in \partial\mathcal{E} \Leftrightarrow |F(z)| = 1 \Leftrightarrow |\overline{F(z)}| = 1 \Leftrightarrow |F(\bar{z})| = 1 \Leftrightarrow \bar{z} \in \partial\mathcal{E}.$$

(c) Zu reellem $\lambda < 0$ vergleichen wir die bisherigen Polynome (Wachstumsfunktionen) $F(z)$ von PZV, IEV, PCV und KRKV.

Wir kontrollieren das maximal zulässige reelle z -Intervall $(z_{\min}, 0)$, aus dem die Schrittweitenbedingung $h < \frac{z_{\min}}{\lambda}$ folgt.

Darstellung mit Maple

```
> # Stabilitätsgebiete (Intervalle) |F(z)|<1 im reellen Fall
# bei z=h*lambda, lambda<0
> f1:=z->1+z; # PZV
f2:=z->1/(1-z); # IEV
f3:=z->1+z+z^2; # PCV
f4:=z->1+z+z^2/2+z^3/6+z^4/24; # KRKV

> q1:=plot([f1(z),f2(z),f3(z),f4(z)],z=-3..0.5,
          scaling=constrained,color=[blue,green,red,magenta],
          view=[-3..0.5,-1.5..2],thickness=2,
          title='Stabilitätsintervalle (zmin,0) zu |F(z)|<1'):
q2:=plot([1,-1],z=-3..0.5,color=black):
q3:=plot([[[-2,-1],[-2,1]],color=blue,linestyle=2):
q4:=plot([[[-1,0],[-1,1]],color=red,linestyle=2):
q5:=plot([[[-2.785,0],[-2.785,1]],color=magenta,linestyle=2):
q6:=plot({[[[-1,0.007],[0,0.007]], [[-2,-0.01],[0,-0.01]],
          [[-2.785,0],[0,0]]},color=black,thickness=2):
q7:=textplot([[[-1.9,1.9,'1+z+z^2'], [-1.5,0.6,'1/(1-z)'],
              [-2.6,-1.3,'1+z'], [-2.5,1.5,'1+z+z^2+z^3/6+z^4/24']]]):
display(q1,q2,q3,q4,q5,q6,q7);
```

```

> datei2:='D:/neundorf/nwptexte/num_gdgl/neun06/maple/stab_02.ps':
interface(plotdevice=ps,
plotoutput=datei2,
plotoptions='color,portrait,width=640,height=480');
plots[display](q1,q2,q3,q4,q5,q6,q7);
interface(plotdevice=win);

```

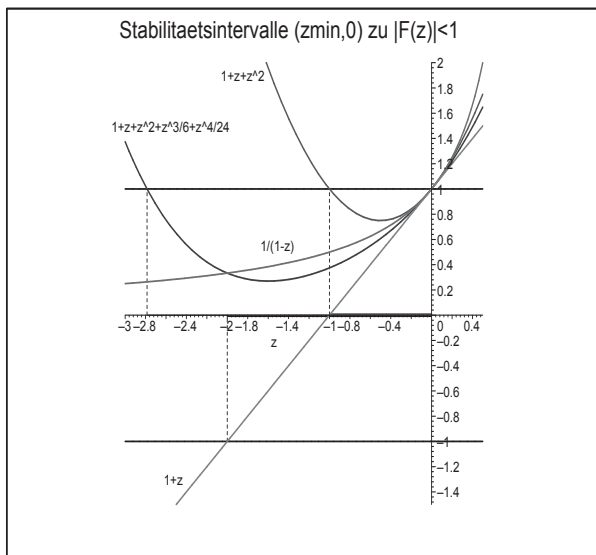


Abb. 7.6

Datei *stab_02.ps*,
 Polynome $F(z)$ zu
 PZV, IEV, PCV, KRKV und
 Stabilitätsintervalle $(z_{min}, 0)$,
 PZV: $z \in (-2, 0)$,
 IEV: $z \in (-\infty, 0)$,
 PCV: $z \in (-1, 0)$,
 KRKV: $z \in (-2.785, 0)$

(d) Analog kann man zu reellem $\lambda < 0$ die Polynome $F(z)$ einiger RKV vergleichen. Wir nehmen dazu die Verfahren PZV, MPZV, HV, (3,2)-RKV (5), (3,3)-RKV (7), (3,3)-KV (8) und KRKV aus Abschnitt 5.2.1. Die Polynome $F(z)$ sind entsprechend dieser Reihenfolge

$$F(z) = 1 + z,$$

$$F(z) = 1 + z + \frac{1}{2}z^2, \quad (2\times)$$

$$F(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{9}z^3,$$

$$F(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3, \quad (2\times)$$

$$F(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4, \quad z = h\lambda < 0.$$

MPZV und HV haben das gleiche Polynom sowie (3,3)-RKV (7) und (3,3)-KV (8).

Wir kontrollieren das zulässige z -Intervall $(z_{min}, 0)$, aus dem die Schrittweitenbedingung $h < \frac{z_{min}}{\lambda}$ folgt. Die Vorgehensweise mit Maple ist wie in Teil (c).

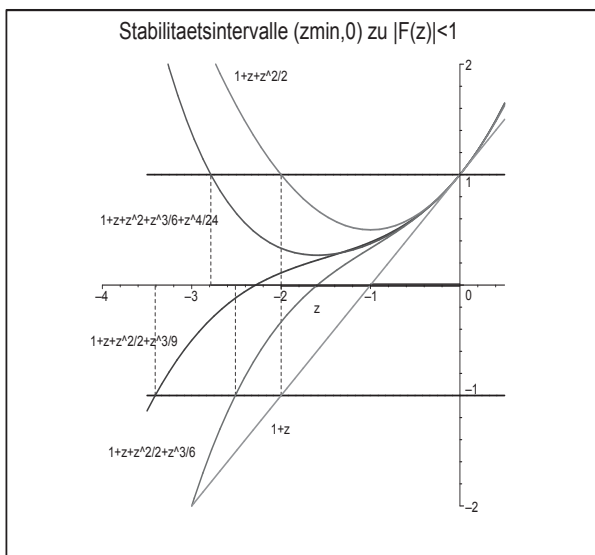


Abb. 7.7

Datei *stab_03.ps*,Polynome $F(z)$ zu
mehreren RKV undStabilitätsintervalle $(z_{min}, 0)$,PZV: $z \in (-2, 0)$,MPZV, HV: $z \in (-2, 0)$,(3,2)-RKV: $z \in (-3.408, 0)$,(3,3)-RKV: $z \in (-2.512, 0)$,KRKV: $z \in (-2.785, 0)$

Man beachte, dass bei RKV gleicher Genauigkeitsordnung dasjenige deutlich größeres h zulässt, das mit mehr Schritten (Anstiegen k_i) arbeitet.

Im nächsten Beispiel versuchen wir, die Idee der RKV auf implizite Verfahren zu verallgemeinern.

Beispiel 7.3 Ein zweistufiges implizites RKV sei definiert durch

$$k_1 = f\left(t_i + \frac{3-\sqrt{3}}{6}h, \eta_i + \frac{1}{4}hk_1 + \frac{3-2\sqrt{3}}{12}hk_2\right),$$

$$k_2 = f\left(t_i + \frac{3+\sqrt{3}}{6}h, \eta_i + \frac{3+2\sqrt{3}}{12}hk_1 + \frac{1}{4}hk_2\right)$$

sowie

$$\eta_{i+1} = \eta_i + \frac{h}{2}(k_1 + k_2). \quad (7.20)$$

Warum ist dieses Verfahren implizit? Worin besteht die Verallgemeinerung gegenüber den in Abschnitt 5.2 definierten RKV?

(1) Berechnung von $F(z)$

Mit $f(t, x) = \lambda x$ folgt das lineare Gleichungssystem für k_1 und k_2

$$k_1 = \lambda\left(\eta_i + \frac{1}{4}hk_1 + \frac{3-2\sqrt{3}}{12}hk_2\right), \quad k_2 = \lambda\left(\eta_i + \frac{3+2\sqrt{3}}{12}hk_1 + \frac{1}{4}hk_2\right)$$

mit der Lösung

$$k_1 = \frac{1 - \frac{\sqrt{3}}{6}h\lambda}{1 - \frac{1}{2}h\lambda + \frac{1}{12}(h\lambda)^2} \lambda\eta_i, \quad k_2 = \frac{1 + \frac{\sqrt{3}}{6}h\lambda}{1 - \frac{1}{2}h\lambda + \frac{1}{12}(h\lambda)^2} \lambda\eta_i.$$

Hieraus ergibt sich

$$F(z) = \frac{1 + \frac{z}{2} + \frac{z^2}{12}}{1 - \frac{z}{2} + \frac{z^2}{12}}. \quad (7.21)$$

(2) Untersuchung auf A-Stabilität

Das Gebiet der absoluten Stabilität wird begrenzt durch die Werte z mit $|F(z)| = 1$. Der Ansatz $F(z) = e^{i\varphi}$, $\varphi \in \mathbb{R}$, führt auf die quadratische Gleichung

$$\frac{z^2}{12}(1 - e^{i\varphi}) + \frac{z}{2}(1 + e^{i\varphi}) + (1 - e^{i\varphi}) = 0$$

und abgesehen vom Sonderfall

$$\varphi = 0, \quad e^{i\varphi} = 1, \quad z = 0, \quad F(z) = 1,$$

nach elementaren Umformungen gemäß

$$\frac{1+e^{i\varphi}}{1-e^{i\varphi}} = \frac{1+\cos(\varphi)+i\sin(\varphi)}{1-\cos(\varphi)-i\sin(\varphi)} = \frac{(1+\cos(\varphi)+i\sin(\varphi))(1-\cos(\varphi)+i\sin(\varphi))}{(1-\cos(\varphi))^2-(i\sin(\varphi))^2} = i \frac{\sin(\varphi)}{1-\cos(\varphi)}$$

auf die Gleichung

$$z^2 + i6az + 12 = 0 \quad \text{mit} \quad a = \frac{\sin(\varphi)}{1-\cos(\varphi)} \in \mathbb{R}.$$

Die beiden Lösungen

$$z = i(-3a \pm \sqrt{9a^2 + 12})$$

liegen auf der imaginären Achse. Damit trennt die imaginäre Achse genau den Bereich der absoluten Stabilität von den Werten z mit $|F(z)| > 1$.

Das Stabilitätsgebiet \mathcal{E} enthält die linke komplexe Halbebene \mathbb{C}^- . Es folgt somit die A-Stabilität des Verfahrens.

In der Regel ist es sehr schwierig, den Bereich \mathcal{E} der absoluten Stabilität explizit zu beschreiben. Aber mit Maple kann man dies auch im Komplexen bequem machen.

```
> # Stabilitätsgebiete |F(z)|<1 im komplexen Fall
# bei z=h*lambda, Re(lambda)<0
# Grenzen der Stabilitätsgebiete |F(z)|=1 --> Konturen
> f0:=unapply(abs(1-x-I*y),x,y); # IEV
f1:=unapply(abs(x+I*y+1),x,y); # PZV
f2:=unapply(abs((x+I*y)^2/2+x+I*y+1),x,y); # PCV
f3:=unapply(abs((x+I*y)^2/2+x+I*y+1),x,y); # MPZV, HV
f4:=unapply(abs((x+I*y)^3/6+(x+I*y)^2/2+x+I*y+1),x,y); # (3,3)-RKV
f5:=unapply(abs((x+I*y)^3/9+(x+I*y)^2/2+x+I*y+1),x,y); # (3,2)-RKV
f6:=unapply(abs((x+I*y)^4/24+(x+I*y)^3/6+(x+I*y)^2/2+x+I*y+1),x,y);
# KRVK
> p0:=contourplot(f0(x,y),x=-4..2,y=-3..3,contours=[1],color=black,
```



```

view=[-4..2,-3..3],thickness=2,scaling=constrained,
title='Stabilitaetsgebiete fuer RKV, PCV gemaess |F(z)|<1':

p1:=contourplot(f1(x,y),x=-4..2,y=-3..3,contours=[1],color=blue):
p2:=contourplot(f2(x,y),x=-4..2,y=-3..3,contours=[1],color=brown):
p3:=contourplot(f3(x,y),x=-4..2,y=-3..3,contours=[1],color=green):
p4:=contourplot(f4(x,y),x=-4..2,y=-3..3,contours=[1],color=magenta):
p5:=contourplot(f5(x,y),x=-4..2,y=-3..3,contours=[1],color=red):
p6:=contourplot(f6(x,y),x=-4..2,y=-3..3,contours=[1],color=cyan):
p7:=textplot([[[-1.6,0.5,'PZV'],[-1.0,1.5,'MPZV, HV'],
               [-3.0,0.5,'(3,2)-RKV'],[-1.0,1.9,'(3,3)-RKV'],
               [-2.2,1.4,'KRKV'],[1,1.15,'IEV'],[-0.75,0.5,'PCV']]]:
> display(p0,p1,p2,p3,p4,p5,p6,p7);

> datei4:='D:/neundorf/nwptexte/num_gdgl/neun06/maple/stab_04.ps':
interface(plotdevice=ps,
           plotoutput=datei4,
           plotoptions='color,portrait,width=640,height=480');
plots[display](p0,p1,p2,p3,p4,p5,p6,p7);
interface(plotdevice=win);

```

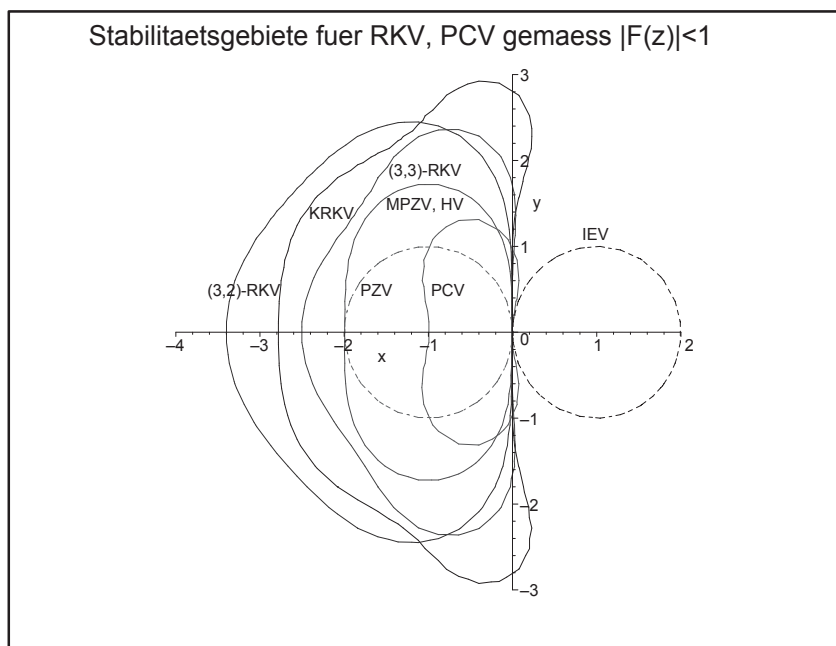


Abb. 7.8 Datei *stab_04.ps*,

Stabilitätsgebiete \mathcal{E} gemäß $|F(z)| < 1$ zu mehreren RKV:
 IEV, PZV, PCV, MPZV~HV, (3,3)-RKV, KRKV, (3,2)-RKV

Einfacher ist, wie man schon gesehen hat, die Untersuchung des Stabilitätsintervalls I_A , definiert als die Einschränkung des Bereichs auf reellwertige negative Werte, welche wir zur Unterscheidung mit ξ bezeichnen wollen, d.h.

$$I_A = \mathcal{E} \cap \mathbb{R}^- = \{\xi \in \mathbb{R} : \xi < 0 \text{ und } |F(\xi)| < 1\}. \quad (7.22)$$

Beispiel 7.4

(a) Für das explizite und das implizite Euler-Verfahren haben wir dies in den Beispielen 7.2 (a), (b) schon gemacht. Somit folgen die Stabilitätsintervalle aus der Untersuchung des Gebiets der absoluten Stabilität als $I_A = (-2, 0)$ bzw. als $I_A = (-\infty, 0)$.

(b) Für das PCV auf der Basis dieser Euler-Formeln gemäß Beispiel 7.2 (c) ist

$$F(\xi) = 1 + \xi + \xi^2 = \left(\xi + \frac{1}{2}\right)^2 + \frac{3}{4}.$$

Dies ist eine nach oben geöffnete Parabel mit Minimalwert $\frac{3}{4}$. Damit ergeben sich die Grenzen des Stabilitätsintervalls aus der Lösung der quadratischen Gleichung $F(\xi) = 1$, und es folgt $I_A = (-1, 0)$. Damit ist das Stabilitätsintervall kleiner als das des PZV.

(c) Für das KRKV ist

$$F(\xi) = 1 + \xi + \frac{1}{2}\xi^2 + \frac{1}{6}\xi^3 + \frac{1}{24}\xi^4.$$

Die Grenzen des Stabilitätsintervalls erhält man aus dem Ansatz $F(\xi) = 1$. Aus Maple-Berechnungen bzw. numerischen Rechnungen, z.B. mit dem Newton-Verfahren, folgt

$$\begin{aligned} I_A &= \left(-\frac{4}{3} - \frac{1}{3}(172 + 36\sqrt{29})^{1/3} + \frac{20}{3}(172 + 36\sqrt{29})^{-1/3}, 0\right) \\ &= \left(-\frac{4}{3} - \sqrt[3]{\frac{4\sqrt{29}}{3} + \frac{172}{27}} + \sqrt[3]{\frac{4\sqrt{29}}{3} - \frac{172}{27}}, 0\right) \\ &= (-2.785\,293, 0). \end{aligned}$$

(d) Für das (3,3)-RKV erhält man

$$F(\xi) = 1 + \xi + \frac{1}{2}\xi^2 + \frac{1}{6}\xi^3$$

und

$$I_A = \left(-1 - \sqrt[3]{\sqrt{17} + 4} + \sqrt[3]{\sqrt{17} - 4}, 0\right) = (-2.512\,745, 0).$$

(e) Betrachten wir als AWP die homogene autonome lineare gDGL 2. Ordnung

$$x'' + 12x' + 20x = 0, \quad x(0) = 4, \quad x'(0) = 0, \quad t \in [0, T],$$

mit der gegen Null tendierenden Lösung $x(t) = 5e^{-2t} - e^{-10t}$.

Mit $x' = y$ erhält man die Darstellung als homogenes lineares SysgDGL 1. Ordnung

$$\begin{aligned} x' &= y, & x(0) &= 4, \\ y' &= -20x - 12y, & y(0) &= 0, \\ \mathbf{x}' &= A \mathbf{x}, & A &= \begin{pmatrix} 0 & 1 \\ -20 & -12 \end{pmatrix}, & \mathbf{x}(0) &= (4, 0)^T. \end{aligned}$$

Die Matrix A besitzt die reellen Eigenwerte $\lambda_1 = -2$ und $\lambda_2 = -10$ sowie die zugehörigen Eigenvektoren $\mathbf{v}_1 = (1, -2)^T$ und $\mathbf{v}_2 = (1, -10)^T$ (vergl. Abschnitt 4.2, Bsp. 4.2 (b)).

Das PZV hat zur gewählten Schrittweite h mit $\eta_0 = \mathbf{x}(0)$ die Notation

$$\eta_i = (I + hA) \eta_{i-1} = (I + hA)^i \eta_0 = \tilde{A}^i \eta_0 = \begin{pmatrix} 1 & h \\ -20h & 1 - 12h \end{pmatrix}^i \begin{pmatrix} 4 \\ 0 \end{pmatrix}.$$

Die Matrix $\tilde{A} = I + hA$ bzw. ihre Eigenwerte $\tilde{\lambda} = 1 + h\lambda$ charakterisieren das Wachstum der Näherungslösung η_i . Das Wachstumsverhalten kann man mit der Funktion $F(z) = 1 + z$, $z = h\lambda(A)$, bzw. der Verstärkungsmatrix $E(A, h) = I + hA$ beschreiben. Zur Stabilität braucht man die Bedingungen $|\tilde{\lambda}_{1,2}| < 1$, die auf die Schrittweitenbeschränkung $h < \frac{2}{|\lambda_2|} = \frac{1}{5}$ führt.

Wird $\eta_i = \tilde{A} \eta_{i-1}$ als Differenzgleichung mit dem Ansatz $\eta_i = (\eta_{1i}, \eta_{2i})^T = (u\xi^i, v\xi^i)^T$ gelöst, gelangt man auf die Beziehung

$$\begin{aligned} u\xi &= u + hv, \\ v\xi &= -20hu + (1 - 12h)v, \end{aligned}$$

damit auf die quadratische Gleichung $\xi^2 - (2 - 12h)\xi + 1 - 12h + 20h^2 = 0$ mit den beiden Wurzeln $\xi_{1,2} = \tilde{\lambda}_{1,2} = \{1 - 2h, 1 - 10h\}$.

Es bleibt somit wegen der Forderung $|\xi_{1,2}| < 1$ natürlich bei der Ungleichung für die Schrittweite $h < \frac{1}{5}$.

In der allgemeinen Lösung der Differenzgleichung $\eta_{1i} = u_1\xi_1^i + u_2\xi_2^i$, $\eta_{2i} = v_1\xi_1^i + v_2\xi_2^i$ sind noch die AB zu berücksichtigen.

Somit ergeben sich $u_1 = 5$, $u_2 = -1$, $v_1 = -10$, $v_2 = 10$ und

$$\begin{aligned} \eta_{1i} &= 5(1 - 2h)^i - 1(1 - 10h)^i, \\ \eta_{2i} &= -10(1 - 2h)^i + 10(1 - 10h)^i. \end{aligned}$$

Das sind Näherungen der Ordnung h zur exakten Lösung und ihrer Ableitung.

Wir bemerken noch einmal, dass bei ESV die Konsistenz des Verfahrens und die Lipschitz-Stetigkeit der rechten Seite der gDGL für den Nachweis der Konvergenz ausreicht. Da ein Konvergenzsatz immer hinreichend kleine Schrittweiten h bzw. $h \rightarrow 0$ braucht, werden diese so klein sein, dass dadurch auch die Schrittweitenbeschränkung $h < h_{max}$ für ein stabiles Lösungsverhalten in der Regel eingehalten wird.

7.2 Stabilität von Mehrschrittverfahren

Übertragen wir nun die Überlegungen zur Stabilität von ESV auf die MSV (6.56).

Setzen wir das Testproblem $x' = \lambda x$ in das MSV ein, so erhalten wir die homogene lineare Differenzgleichung

$$(1 - h\lambda b_r)\eta_{j+r} + \dots + (a_1 - h\lambda b_1)\eta_{j+1} + (a_0 - h\lambda b_0)\eta_j = 0. \quad (7.23)$$

Zur Lösung bedienen wir uns, entsprechend den Überlegungen des Abschnitts 6.3 des mit (7.23) verknüpften Polynoms

$$\phi(\xi) = (1 - h\lambda b_r)\xi^r + \dots + (a_1 - h\lambda b_1)\xi + (a_0 - h\lambda b_0) = \rho(\xi) - h\lambda\sigma(\xi), \quad (7.24)$$

wobei $\xi \in \mathbb{C}$ sowie $\rho(\xi)$ und $\sigma(\xi)$ die in der Definition 6.3 notierten charakteristischen Polynome des MSV sind.

Für $h = 0$ haben wir das Polynom $\phi(\xi) = \rho(\xi)$ (6.52) sowie dazu die Gestalt der Lösung η_i (6.54) schon untersucht.

Hat $\phi(\xi)$ die r paarweise verschiedenen Nullstellen $\xi_1, \xi_2, \dots, \xi_r$, so notiert man nach Satz 6.3 die allgemeine Lösung der Differenzgleichung (6.53) in der Form

$$\eta_i = \gamma_1 \xi_1^i + \gamma_2 \xi_2^i + \dots + \gamma_r \xi_r^i. \quad (7.25)$$

Offenbar ist das Abklingen der Folge η_i gegen Null genau dann gewährleistet, wenn die Beträge aller Nullstellen ξ_k kleiner als 1 sind. Dasselbe gilt auch, wenn die Nullstellen nicht paarweise verschieden sind. Wir erkennen in dieser Aussage auch das Stabilitätskriterium aus dem Satz 6.5. Man unterscheidet also die Konvergenz des MSV bei $h = \frac{T-t_0}{N} \rightarrow 0$, d.h. $N \rightarrow \infty$, in dem endlichen Intervall $[t_0, T]$ von seinem Langzeitverhalten bei gegebener Schrittweite h und $t_i \rightarrow \infty$ bzw. $i \rightarrow \infty$.

Dies führt zu folgender Definition.

Definition 7.3 *Zu dem MSV (6.56) heißt die Menge der Werte $z = h\lambda$, für welche das charakteristische Polynom*

$$\phi(\xi) = \phi(\xi; z) = \rho(\xi) - z\sigma(\xi) \quad (7.26)$$

nur Nullstellen $\xi_k = \xi_k(z)$ mit Beträgen $|\xi_k| < 1$ besitzt, das Gebiet der absoluten Stabilität. Das MSV heißt dann absolut stabil.

Wegen der Konsistenzbedingung $\rho(1) = 0$ aus (6.59) und $\phi(\xi) = \rho(\xi)$ bei $z = 0$ hat man immer den speziellen Wert $\xi_k(0) = 1$.

Im Allg. ist es schwierig, die Gebiete der absoluten Stabilität analytisch exakt anzugeben. Zur numerisch-grafischen Darstellung wird man versuchen, den Rand des Stabilitätsgebiets zu bestimmen bzw. bei Verdacht auf A-Stabilität einfach festzustellen, ob der Rand nicht in \mathbb{C}^- liegt.

Der Rand $\partial\mathcal{E}$ ist dadurch bestimmt, dass das Polynom $\phi(\xi)$ Nullstellen der Form

$$\xi = e^{i\varphi} = \cos(\varphi) + i\sin(\varphi), \quad \varphi \in [0, 2\pi),$$

hat.

Der Rand ist also gegeben durch die Lösung der Gleichung

$$\phi(e^{i\varphi}) = \rho(e^{i\varphi}) - z\sigma(e^{i\varphi}) = 0. \quad (7.27)$$

Damit lässt sich das Gebiet der absoluten Stabilität eingrenzen durch die numerische Auswertung des Ausdrucks

$$z = z(\varphi) = z_1(\varphi) + i z_2(\varphi) = \frac{\rho(e^{i\varphi})}{\sigma(e^{i\varphi})}, \quad \varphi \in [0, 2\pi]. \quad (7.28)$$

Der Wert $z(0)$ ist hierbei in der Regel bekannt.

Nach der ersten der beiden Konsistenzbedingungen (6.59) gilt nämlich $\rho(e^0) = \rho(1) = 0$. Ist außerdem $\rho'(1) \neq 0$, so folgt aus der zweiten Bedingung $\sigma(1) = \rho'(1) \neq 0$ und damit $z(0) = 0$.

Die ESV der Form $\eta_{i+1} = F(z)\eta_i$ (7.8) lassen sich als einstufige MSV in die Theorie einordnen. Das charakteristische Polynom dazu ist $\phi(\xi) = \xi - F(z)$ und seine einzige Nullstelle beträgt $\xi_1 = F(z)$, die zwecks Stabilität vom Betrag kleiner als Eins sein muss. Die Bedingung $F(z) = e^{i\varphi}$ wird für die Bestimmung des Randes des Stabilitätsgebiets als Konturkurve (Niveaulinie) $|F(z)| = |F(x + iy)| = 1$ verwendet. Diese haben wir in der Abbildung 7.8 bereits dargestellt. Die Vorgehensweise entspricht genau der Aussage in der Definition 7.2.

7.2.1 Stabilitätsgebiet von Mehrschrittverfahren

Wir haben die Parameterschemata von vier Klassen von MSV in den Abschnitten 6.1, 6.2 gezeigt. Dazu wollen wir die Stabilitätsgebiete berechnen.

Wie in Bemerkung 7.1 (b) würde die Darstellung von \mathcal{E} im Bereich \mathbb{C}^- bei nicht negativem Imaginärteil genügen. Auch hier haben wir die Symmetrie des Gebietes zur reellen Achse, denn es gilt im Komplexen für $\varphi \in [0, 2\pi]$

$$z = \frac{\rho(e^{i\varphi})}{\sigma(e^{i\varphi})}, \quad \bar{z} = \frac{\rho(\overline{e^{i\varphi}})}{\sigma(\overline{e^{i\varphi}})} = \frac{\rho(e^{-i(\varphi-2\pi)})}{\sigma(e^{-i(\varphi-2\pi)})} = \frac{\rho(e^{i(2\pi-\varphi)})}{\sigma(e^{i(2\pi-\varphi)})} = \frac{\rho(e^{i\zeta})}{\sigma(e^{i\zeta})}, \quad \zeta \in (0, 2\pi].$$

Da jedoch manche Konturen $\partial\mathcal{E}$ teilweise oder ganz in \mathbb{C}^+ liegen, interessieren wir uns für ihren vollständigen Verlauf. Dabei ist es sinnvoll, manche Stücke mit feiner Winkelschrittweite $\Delta\varphi > 0$ zu rechnen oder gewisse Gebiete sich vergrößert zu betrachten.

In den Gebieten, wo das MSV nicht absolut stabil ist, hat die Funktion $\phi(\xi)$ auf jeden Fall mindestens eine Nullstelle ξ_k mit Betrag größer als Eins. Beträge von weiteren Nullstellen können dort durchaus sowohl kleiner als auch größer als Eins sein.

Wir werden dazu auch Abbildungen mit Verläufen mehrerer Konturen gemäß

$$z = \frac{\rho(c e^{i\varphi})}{\sigma(c e^{i\varphi})}, \quad |c e^{i\varphi}| = c, \quad c \in (0, +\infty), \quad (7.29)$$

erstellen.

Die Berechnungen kann man sowohl in Maple also auch in MATLAB machen, was wir auch im ersten Fall tun wollen. Dabei muss man auf die in den CAS üblichen Indexgrenzen bei Feldern achten und eventuell geeignete Indexverschiebungen bei den Koeffizienten der Parameterschemata einbeziehen.

(1) Explizites n -stufiges Adams-Bashforth-Verfahren n -ter Ordnung

Die Koeffizienten der MSV AB(n), $n = 1, 2, \dots, 6$, entnehmen wir der Tabelle 6.2.

Maple-Berechnungen

```

> # Parameterschema zu AB(n), n=1,2,...,6
m_alpha:=vector(7,[1,-1,0,0,0,0,0]):
m_beta :=matrix(6,7,
  [[0, 1, 0, 0, 0, 0, 0],
   [0, 3/2, -1/2, 0, 0, 0, 0],
   [0, 23/12, -16/12, 5/12, 0, 0, 0],
   [0, 55/24, -59/24, 37/24, -9/24, 0, 0],
   [0, 1901/720, -2774/720, 2616/720, -1274/720, 251/720, 0],
   [0, 4277/1440, -7923/1440, 9982/1440, -7298/1440, 2877/1440, -475/1440]]):

> Digits:=16:
> n:=6:
plv:=vector(n,[]):
> # Rand der Stabilitaetsgebiete
> for k from 1 to n do
  alpha:=vector(k+1,[seq(m_alpha[i],i=1..k+1)]);
  beta :=vector(k+1,[seq(m_beta[k,i],i=1..k+1)]);
  i:='i':
  prho :=unapply(sum(alpha[i]*xi^(k+1-i),i=1..k+1),xi);
  psigma:=unapply(sum(beta[i]*xi^(k+1-i),i=1..k+1),xi);

  l:=NULL:
  if k<n-1 then m:=200; elif k<n then m:=400; else m:=3000; end if;
  dphi:=evalf(2*Pi/m);
  for i from 0 by 1 to m do
    phi_i:=i*dphi;
    w :=evalf(exp(I*phi_i));
    z :=prho(w)/psigma(w);
    z1:=evalf(Re(z));
    z2:=evalf(Im(z));
    # lprint(z1,z2); # Kontrolle, Zwischenausgabe
    l:=l,[z1,z2];
  end do;
  if k=1 then l1:=l; farbe:=color=black;
  elif k=2 then l2:=l; farbe:=color=blue;
  elif k=3 then l3:=l; farbe:=color=red;
  elif k=4 then l4:=l; farbe:=color=green;
  elif k=5 then l5:=l; farbe:=color=magenta;
  elif k=6 then l6:=l; farbe:=color=cyan;
  end if;
  pp:=plot([l],farbe);
  print(pp); # Ausgabe der Stabilitaetsgebiete zu AB(1..6)
  plv[k]:=pp;
end do:

```

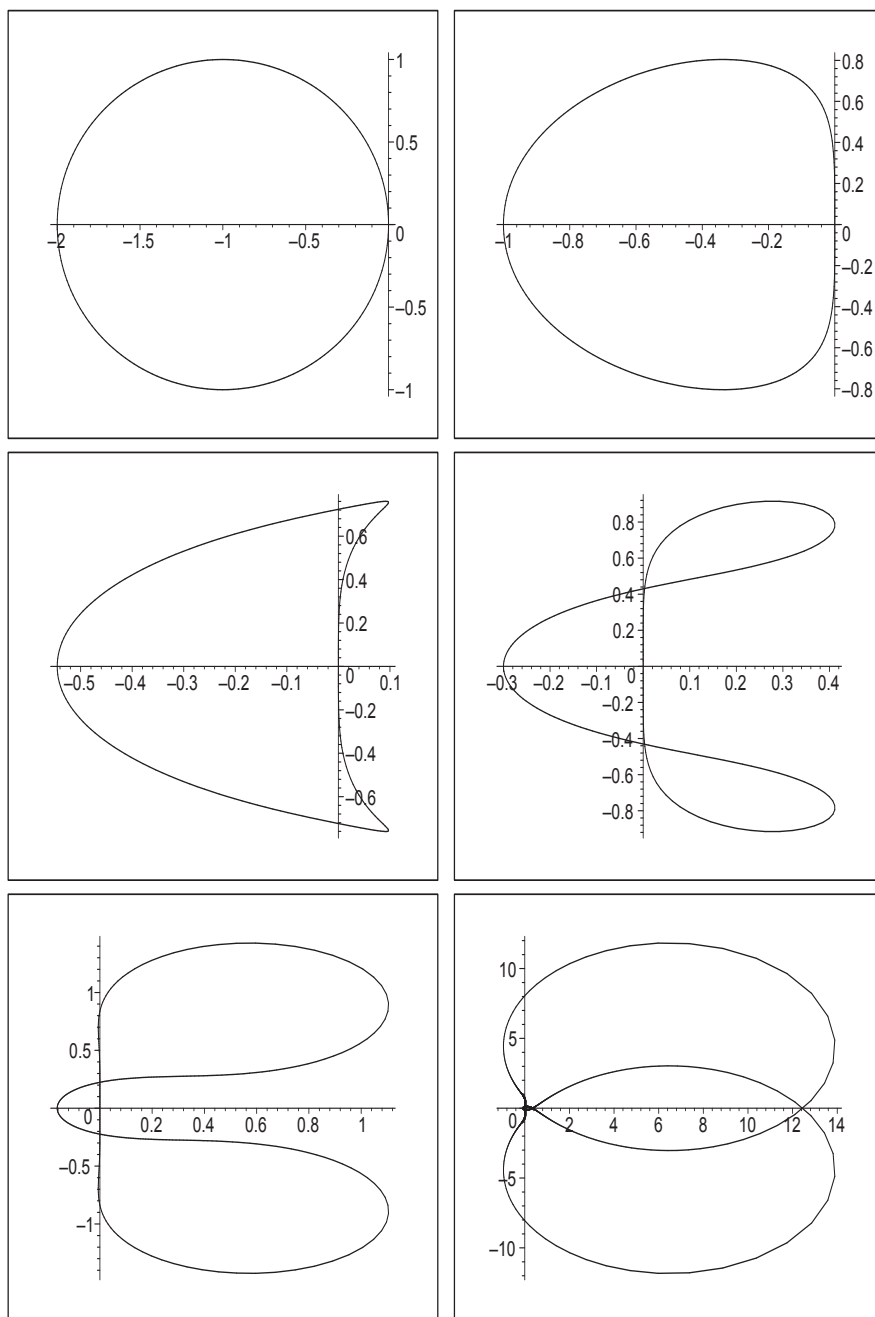


Abb. 7.9 Dateien *ab_stab01..06.eps*, Stabilitätsgebiete \mathcal{E} ("Inneres") gemäß der Nullstellen $|\xi_k| < 1$ von $\phi(\xi)$ zu den MSV $AB(n)$, $n = 1, 2, \dots, 6$

Wir fassen alle sechs Stabilitätsgebiete jeweils in einer Abbildung bei Wahl verschiedener Ausschnitte zusammen.

```
> i:='i':
display(seq(plv[i],i=1..n),labels=['Re z','Im z'],
title='Stabilitaetsgebiete zu AB(n), n=1,2,...,6');
```

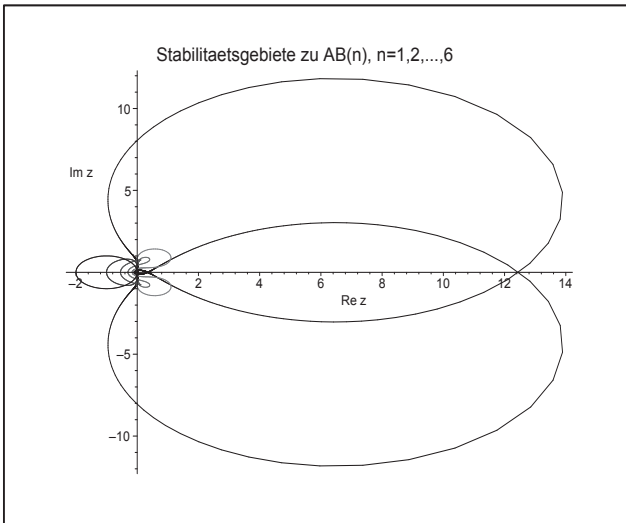


Abb. 7.10

Datei *ab_stab07.ps*,
Stabilitätsgebiete \mathcal{E}
zu $AB(n)$, $n = 1, 2, \dots, 6$

```
> pt:=textplot([[[-1.7,0.4,'AB(1)'],[-0.7,0.4,'AB(2)'],[-0.35,0.4,'(3)'],
[-0.15,0.25,'(4)'],[-0.07,0.13,'(5)'],[-0.03,0.05,'(6)']]):
display(seq(plv[i],i=1..n),pt,view=[-2..2,-2..2],labels=['Re z','Im z'],
title='Stabilitaetsgebiete zu AB(n), n=1,2,...,6');
```

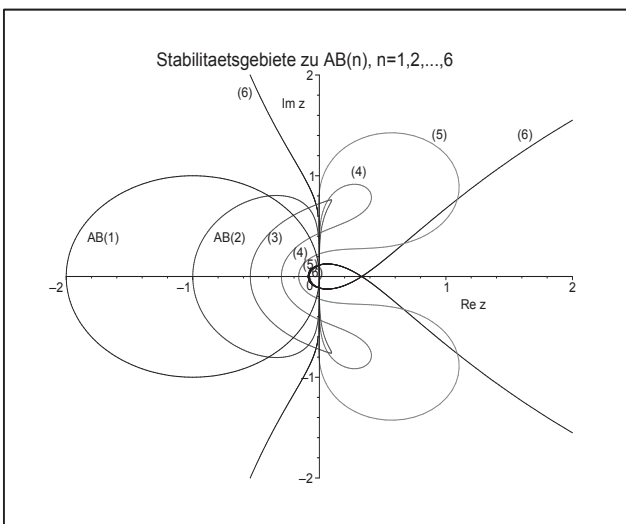


Abb. 7.11

Datei *ab_stab08.ps*,
Stabilitätsgebiete \mathcal{E}
zu $AB(n)$, $n = 1, 2, \dots, 6$
(Ausschnittsvergrößerung)


```
> display(seq(plv[i],i=1..n),pt,view=[-2..0,-1..1],labels=['Re z','Im z'],
          title='Stabilitaetsgebiete zu AB(n), n=1,2,...,6');
```

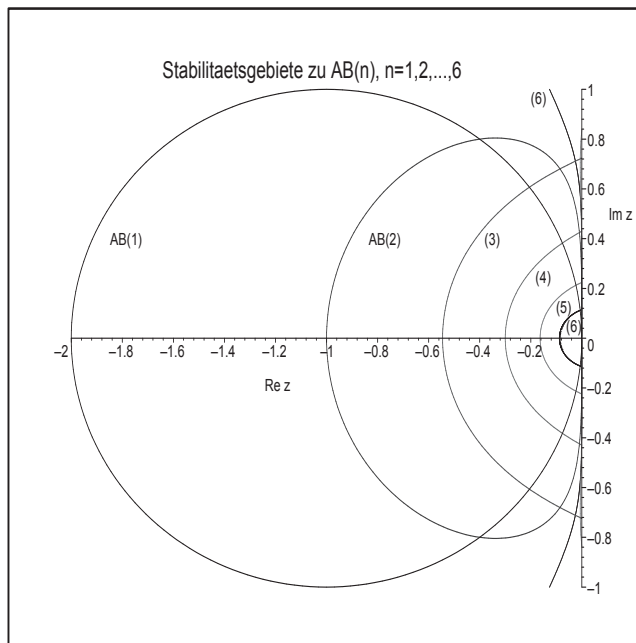


Abb. 7.12

Datei *ab_stab09.ps*,
Stabilitätsgebiete \mathcal{E}
zu $AB(n)$, $n = 1, 2, \dots, 6$

Man erkennt, dass die Stabilitätsgebiete zu $AB(n)$ als Bereiche innerhalb der Konturen in \mathbb{C}^- mit wachsendem n systematisch kleiner werden. Damit wird die Bedingung für die Schrittweite h immer strenger.

Beim Erzeugen des Randes des Stabilitätsgebiets mit den Winkelwerten $\varphi_i = i\Delta\varphi$, $i = 0, 1, \dots, m$, $\Delta\varphi = \frac{2\pi}{m}$, ist eine weitere Besonderheit zu erkennen. Die 1-Konturen "starten" von $z = 0$ nach oben entweder mit einer Krümmung nach links wie bei $AB(1)$, $AB(2)$, $AB(5)$ und $AB(6)$, oder nach rechts wie $AB(3)$ und $AB(4)$. Bei $AB(5)$ und $AB(6)$ entfernen sich die 1-Konturen jedoch nur wenig von der imaginären Achse, um bald in den Bereich \mathbb{C}^+ zu wechseln. Dort machen sie eine "Schleife" und kehren in den Bereich \mathbb{C}^- zurück, um schließlich das eigentliche Stabilitätsgebiet \mathcal{E} zu beschreiben.

Bei $AB(5)$ hat dieser "Rückkehrpunkt" (Schnittpunkt mit der imaginären Achse) die Koordinaten $(0, 0.224\dots)$, bei $AB(6)$ ist es $(0, 0.114\dots)$. Bei $AB(5)$ verläuft die Kontur zum Wert $c = 1.004886$ anfänglich in \mathbb{C}^+ und "berührt" bei rund $(0, 0.685452)$ die imaginäre Achse, um im weiteren Verlauf eine Schleife im \mathbb{C}^+ zu beschreiben.

Nur bei $AB(3)$ und $AB(4)$ gehören kleine symmetrische Gebiete von \mathbb{C}^+ zum Stabilitätsgebiet. Das würde aber eine negative Schrittweite h bedeuten. Solche Gebiete werden einfach "abgeschnitten", sodass dort ein kleines symmetrisches Intervall auf der imaginären Achse ein Randstück des Stabilitätsgebiets bildet. In der Literatur wird bei Abbildungen mit Stabilitätsgebieten meistens die komplexe Halbebene \mathbb{C}^+ nicht einbezogen (vergl. Abb. 7.12).

Analog macht man die Berechnungen in MATLAB.

Die Rechenzeiten sind wesentlich kürzer als bei Maple.

Außerdem kann man die Entstehung der einzelnen Konturen sehr schön durch die Kombination der MATLAB-Kommandos `hold on`, `pause` und `hold off` verfolgen.

Oder hat man z.B. die 1-Kontur zum einem $AB(n)$, so können weitere Konturen zu den Werten $c \in (0, 2)$ mit $c = |ce^{i\varphi}|$ hinzugefügt werden. Dadurch erkennt man neben dem Stabilitätsgebiet, in welchen Teilgebieten die Funktion $\phi(\xi)$ ihre weiteren Nullstellen ξ_k mit Betrag größer bzw. kleiner als Eins hat. Insbesondere, wenn die 1-Kontur irgendwelche Schleifen macht, ist es interessant, was innerhalb dieser passiert.

Rahmenprogramm zum Test von Stabilitätsgebieten zu $AB(n)$

```
% ab_stab.m
% AB(n), Stabilitaetsgebiete

clear all
clc

% Parameterschema zu AB(n), n=1,2,...,6
m_alpha = [ 1  -1   0   0   0   0   0];

m_beta = ...
[0  1       0       0       0       0       0;
 0  3/2     -1/2     0       0       0       0;
 0  23/12   -16/12   5/12    0       0       0;
 0  55/24   -59/24   37/24   -9/24   0       0;
 0  1901/720 -2774/720 2616/720 -1274/720 251/720 0;
 0  4277/1440 -7923/1440 9982/1440 -7298/1440 2877/1440 -475/1440];

I = sqrt(-1);

a = 2;
b = 1.5;
ta = [-a,0.75*a];
tb = [-b,b];
plot(ta,[0,0], 'k', [0,0],tb, 'k');
axis([-a 0.75*a -b b]); % axis([-2 14 -12 12]);

hold on

% Rand der Stabilitaetsgebiete
n = 6;
for k = 1:n
    alpha = m_alpha(1:k+1);
    beta = m_beta(k,1:k+1);

    if k<n-1, m = 200; elseif k<n, m = 400; else m = 3000; end;
    dphi = 2*pi/m;
    for i = 0:1:m
        phi_i = i*dphi;
        w = exp(I*phi_i);
        z = polyval(alpha,w)/polyval(beta,w);
        zr = real(z);
        zi = imag(z);
```

```

if i==0
    z1r = zr;
    z1i = zi;
end;

% fprintf('%12.9e %12.9e\n',zr,zi); % Kontrolle, Zwischenausgabe

switch k
    case 1, plot([zr,z1r],[zi,z1i],'k');
    case 2, plot([zr,z1r],[zi,z1i],'b');
    case 3, plot([zr,z1r],[zi,z1i],'r');
    case 4, plot([zr,z1r],[zi,z1i],'g');
    case 5, plot([zr,z1r],[zi,z1i],'m');
    case 6, plot([zr,z1r],[zi,z1i],'c');
end;
z1r = zr;
z1i = zi;
% pause
end;
end;

title('Stabilitätsgebiete zu AB(n)');
xlabel('Re z');
ylabel('Im z');
text(-1.8,0.5,'AB(1)');
text(-0.8,0.5,'AB(2)');
text(-0.45,0.3,'(3)');
text(-0.2,0.25,'(4)');
text(-0.1,0.1,'(5)');
text(-0.08,-0.01,'(6)');

print ab_stab1.ps -dpsc
hold off

```

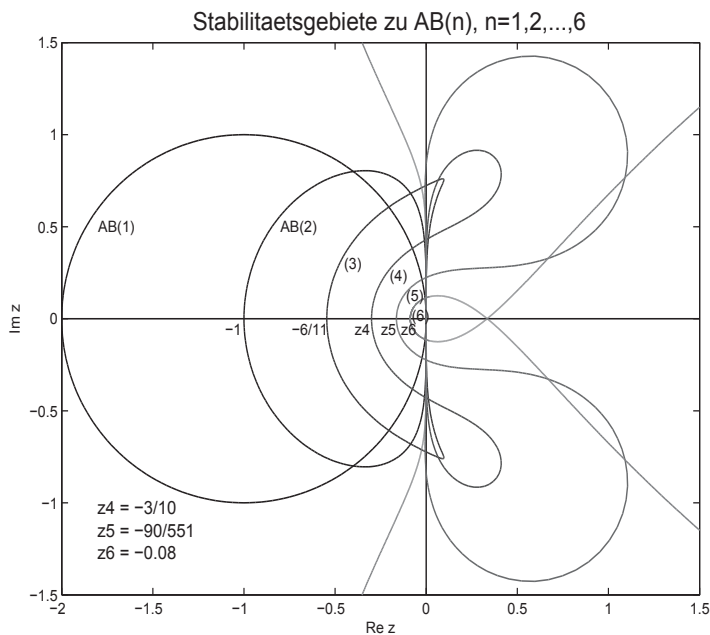


Abb. 7.13
 Datei *ab_stab1.ps*,
 Stabilitätsgebiete \mathcal{E}
 zu AB(n),
 $n = 1, 2, \dots, 6$

Wir betrachten nun die Situation zum Stabilitätsgebiet \mathcal{E} von $AB(5)$. Dabei zeichnen wir den Verlauf weiterer Konturen. Für Werte $c = |ce^{i\varphi}| < 1$ liegen die Konturen überall verstreut, wobei es "sehr eng" in \mathcal{E} zugeht, und relativ spärlich in den Schleifen. Für $c > 1$ ist "mehr Ordnung" bei den Konturen zu erkennen, wobei natürlich keine in \mathcal{E} auftreten dürfen und es relativ dicht in den Schleifen wird.

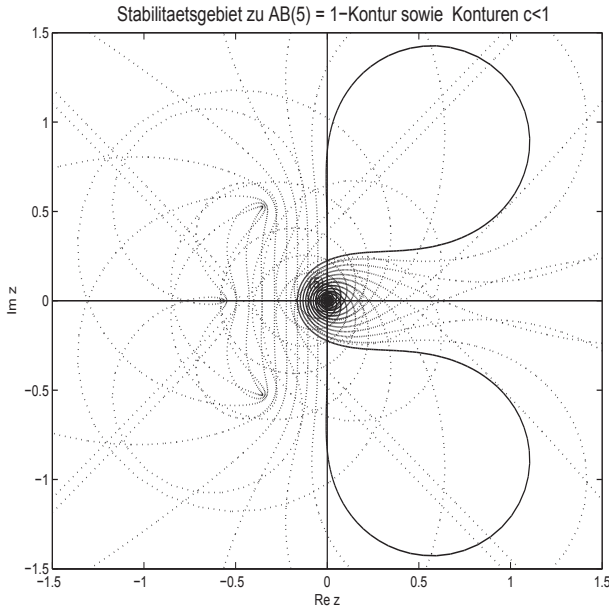


Abb. 7.14

Datei *ab_stab2a.ps*,
Stabilitätsgebiet
zu $AB(5)$ sowie
weitere Konturen
mit $c = 0.05(0.05)1$

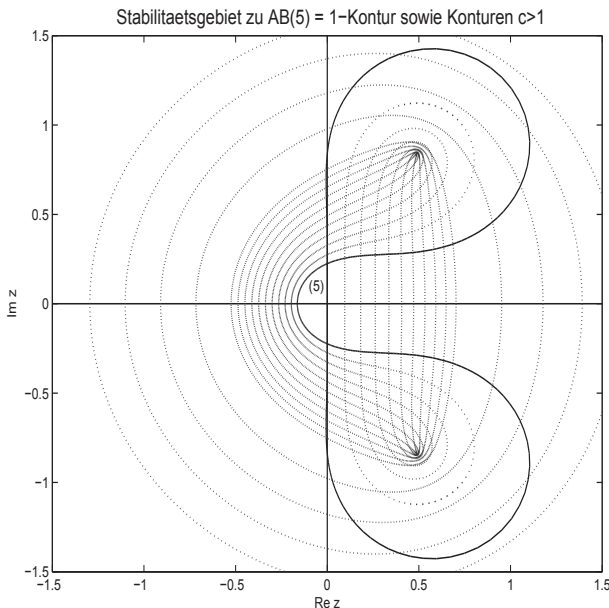


Abb. 7.15

Datei *ab_stab2b.ps*,
Stabilitätsgebiet
zu $AB(5)$ sowie
weitere Konturen
mit $c = 1(0.1)2(0.5)4$

Die nächsten Fälle machen wir nicht so ausführlich bzw. nur auszugsweise.

Zum MSV AB(2)

$$\eta_{i+1} = \eta_i + \frac{h}{2}[3f(t_i, \eta_i) - f(t_{i-1}, \eta_{i-1})], \quad i = 1, 2, \dots, \quad (7.30)$$

machen wir einige zusätzliche Betrachtungen und Rechnungen, die für andere einfache MSV in ähnlicher Weise möglich sind.

Dazu haben wir

$$\begin{aligned} \eta_{i+1} - (1 + \frac{3}{2}h\lambda)\eta_i + \frac{1}{2}h\lambda\eta_{i-1} &= 0, \\ \rho(\xi) &= \xi^2 - \xi = \xi(\xi - 1), \quad \text{Stabilitätskriterium erfüllt,} \\ \sigma(\xi) &= \frac{3}{2}\xi - \frac{1}{2}, \\ \phi(\xi) &= \xi^2 - (1 + \frac{3}{2}z)\xi + \frac{1}{2}z, \quad z = h\lambda, \\ \xi_{1,2}(z) &= \frac{1}{2}\left(1 + \frac{3}{2}z \pm \sqrt{(1 + \frac{3}{2}z)^2 - 2z}\right), \quad \text{Nullstellen von } \phi(\xi) \\ &= \frac{1}{4}\left(2 + 3z \pm \sqrt{9z^2 + 4z + 4}\right) \\ &= \frac{1}{4}\left(2 + 3z \pm 3\sqrt{(z + \frac{2}{9})^2 + \frac{32}{81}}\right). \end{aligned}$$

Der Rand des Stabilitätsgebiets ist in Abb. 7.12 gezeigt.

Das daraus erkennbare reelle Stabilitätsintervall $(-1, 0)$, wo $|\xi_{1,2}(z)| < 1$, kann man durch den grafischen Verlauf der beiden reellwertigen Nullstellenkurven $\xi_{1,2}(z)$ deutlich machen.

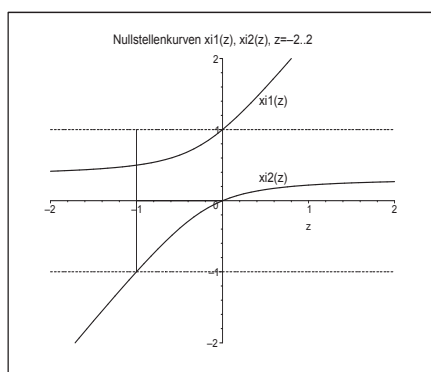


Abb. 7.16

Datei *ab_stab22.ps*,

AB(2), Nullstellenkurven $\xi_{1,2}(z)$, $z \in [-2, 2]$,
und Stabilitätsintervall $(-1, 0)$

Interessant ist, was an der Stelle $z = -1$ passiert.

Dazu nehmen wir der Einfachheit halber $\lambda = -1$, sodass $z = -h$ ist.

AB(2) ist absolut stabil für $h < 1$ und wächst unbeschränkt und oszillierend für $h > 1$. Die allgemeine Lösung des MSV als Differenzgleichung betrachtet ist

$$\eta_i = c_1 \xi_1^i + c_2 \xi_2^i = c_1 \left(\frac{1}{4}(2 - 3h + \sqrt{9h^2 - 4h + 4})\right)^i + c_2 \left(\frac{1}{4}(2 - 3h - \sqrt{9h^2 - 4h + 4})\right)^i,$$

deren Koeffizienten c_1, c_2 sich aus den AB η_0, η_1 ergeben.

Wir nehmen $h = 1$ und $\eta_0 = 1$, $\eta_1 = 1 - h$ und erhalten $\xi_1 = \frac{1}{2}$, $\xi_2 = -1$ sowie

$$\eta_0 = c_1 + c_2 = 1, \quad \eta_1 = c_1 \xi_1 + c_2 \xi_2 = 1 - h,$$

$$c_1 = \frac{2}{3}, \quad c_2 = \frac{1}{3},$$

$$\eta_i = \frac{1}{2}(\eta_{i-2} - \eta_{i-1}) = \frac{2}{3} \left(\frac{1}{2}\right)^i + \frac{1}{3}(-1)^i.$$

Damit wird das oszillierende Verhalten, die fehlende Asymptotik gegen Null und die Divergenz der Näherungsfolge deutlich.

$$\lim_{i \rightarrow \infty} \eta_i = \begin{cases} \frac{1}{3}, & \text{falls } i \text{ gerade,} \\ -\frac{1}{3}, & \text{falls } i \text{ ungerade.} \end{cases}$$

Genauere Untersuchungen mit den AB $\eta_0 = 1$, $\eta_1 = 0$ führen zu den Ungleichungen

$$1 = \eta_0 > \eta_2 > \eta_4 > \dots > \eta_{2i} > \eta_{2i+2} > \dots > \frac{1}{3},$$

$$0 = \eta_1 > \eta_3 > \eta_5 > \dots > \eta_{2i-1} > \eta_{2i+1} > \dots > -\frac{1}{3}.$$

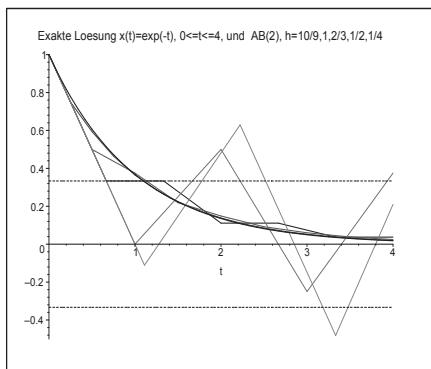


Abb. 7.17

Datei *ab_stab20.ps*,

exakte Lösung $x(t) = e^{-t}$, $0 \leq t \leq 4$,

und Näherungslösungen mittels

AB(2) mit $h = \frac{10}{9}$, 1 , $\frac{2}{3}$, $\frac{1}{2}$, $\frac{1}{4}$

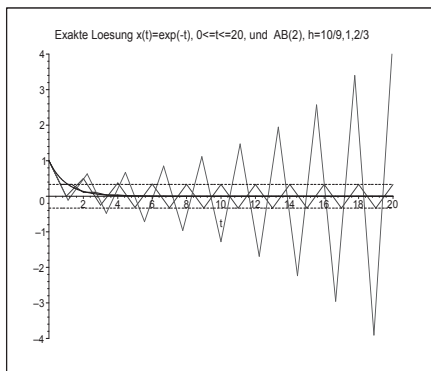


Abb. 7.18

Datei *ab_stab21.ps*,

exakte Lösung $x(t) = e^{-t}$, $0 \leq t \leq 20$,

und Näherungslösungen mittels

AB(2) mit $h = \frac{10}{9}$, 1 , $\frac{2}{3}$

(“Langzeitverhalten“)

Es ist i. Allg. nicht sinnvoll, eine Schrittweite h zu wählen, die sich aus einem z -Wert auf der Grenze des Stabilitätsgebiets ergibt, auch wenn für manche AWP und MSV noch akzeptable Ergebnisse zu erwarten wären.

(2) Implizites s -stufiges Adams-Moulton-Verfahren n -ter Ordnung

Die Koeffizienten des MSV AM(n), $n = 1, 2, \dots, 6$, $s = \max(n - 1, 1)$, entnehmen wir der Tabelle 6.4.

```
% am_stab.m
% AM(n), Stabilitaetsgebiete
% Parameterschema zum AM
m_alpha = [ 1  -1  0  0  0  0];
m_beta  = [ 1      0      0      0      0      0;
           1/2    1/2    0      0      0      0;
           5/12   8/12   -1/12  0      0      0;
           9/24   19/24  -5/24  1/24  0      0;
           251/720 646/720 -264/720 106/720 -19/720 0;
           475/1440 1427/1440 -798/1440 482/1440 -173/1440 27/1440];

I = sqrt(-1);
a = 2; b = 4;
ta = [-3*a,a]; tb = [-b,b];
plot(ta,[0,0], 'k', [0,0],tb, 'k');
axis([-3*a a -b b]);
hold on
% Rand der Stabilitaetsgebiete
n = 6;
for k = 1:n
    if k<3, s = 1; else s = k-1; end;
    alpha = m_alpha(1:s+1); beta = m_beta(k,1:s+1);
    m = 200;
    ...
```

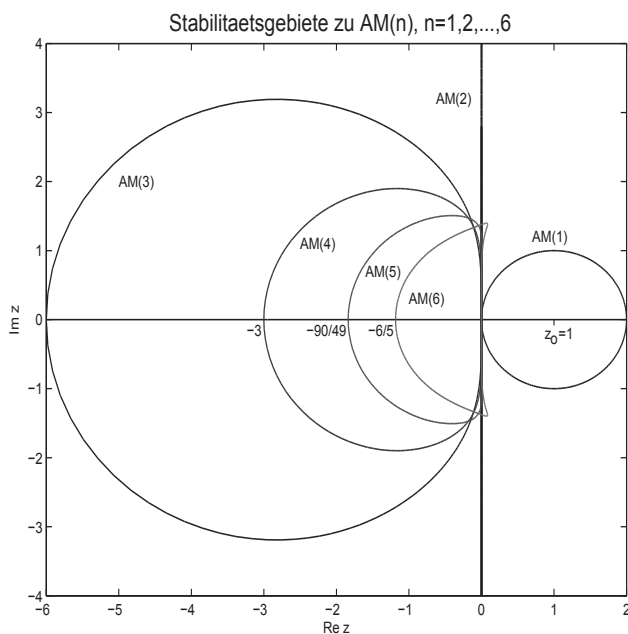


Abb. 7.19

Datei *am_stab1.ps*,
Stabilitätsgebiete \mathcal{E}
zu AM(n), $n = 1, 2, \dots, 6$

Nur AM(1) und AM(2) sind A-stabile MSV, weil die linke komplexe Halbebene \mathbb{C}^- zum Stabilitätsgebiet \mathcal{E} gehört. Die anderen vier AM-Formeln führen auf entsprechende Beschränkungen der Schrittweite h .

Da AM(1) und AM(2) einstufige MSV sind, ist das charakteristische Polynom $\phi(\xi)$ (7.26) problemlos zu berechnen.

AM(1): Es gilt

$$\phi(\xi) = \rho(\xi) - z\sigma(\xi) = \xi - 1 - z\xi = (1-z)\xi - 1,$$

sodass seine einzige Nullstelle $\xi_1 = \frac{1}{1-z}$ ist. Die Stabilitätsbedingung $|\xi_1| < 1$ ist einfach auszuwerten, genauso die Berechnung der Konturen $|\frac{1}{1-z}| = c$.

Damit ist $|z-1| = 1$ die Kreisgleichung für den Rand $\partial\mathcal{E}$ des Stabilitätsgebiets \mathcal{E} .

Die Konturen $|\frac{1}{1-z}| = c$, $c > 0$, sind Kreisumfänge zum Mittelpunkt $z_0 = 1$ und mit dem Radius $\frac{1}{c}$, sodass man eine Schar von konzentrischen Kreisen um den Punkt $z_0 = 1$ zeichnen kann. Die Kreise mit dem Radius $\frac{1}{c} > 1$, d.h. $c < 1$, liegen in \mathcal{E} .

Wichtig ist, dass damit \mathbb{C}^- zum Stabilitätsgebiet \mathcal{E} gehört.

AM(2): Hier haben wir

$$\phi(\xi) = \rho(\xi) - z\sigma(\xi) = (1 - \frac{z}{2})\xi - (1 + \frac{z}{2}),$$

sodass die einzige Nullstelle $\xi_1 = \frac{1+z/2}{1-z/2}$ ist. Die Stabilitätsbedingung $|\xi_1| < 1$ ist einfach auszuwerten, genauso die Berechnung der Konturen $|\frac{1+z/2}{1-z/2}| = c$.

Die Bedingung $|\frac{1+z/2}{1-z/2}| = 1$ für den Rand des Stabilitätsgebiets kann nur für imaginäre Werte z erfüllt werden, denn dann sind Zähler und Nenner zueinander konjugiert komplexe Zahlen mit gleichem Betrag. Man kann auch die Gleichung $\frac{1+z/2}{1-z/2} = e^{i\varphi}$ nehmen.

Ihre Umstellung nach z ergibt $z = 2\frac{e^{i\varphi}-1}{e^{i\varphi}+1} = 2\frac{e^{i\varphi/2}-e^{-i\varphi/2}}{e^{i\varphi/2}+e^{-i\varphi/2}} = i 2 \tan(\varphi/2)$.

Die Konturen $|\frac{1+z/2}{1-z/2}| = c$ sind entweder Kreise in \mathbb{C}^- bei $c < 1$ oder symmetrisch dazu Kreise in \mathbb{C}^+ bei $c > 1$.

Man kann mittels des Ansatzes $z = \Re z + i \Im z = z_1 + i z_2$ diese Kreise bestimmen. Es gilt

$$\begin{aligned} \left| \frac{1+z/2}{1-z/2} \right| &= c, \\ \left| 1 + \frac{z_1}{2} + i \frac{z_2}{2} \right| &= c \left| 1 - \frac{z_1}{2} - i \frac{z_2}{2} \right|, \\ (1 + \frac{z_1}{2})^2 + (\frac{z_2}{2})^2 &= c^2 [(1 - \frac{z_1}{2})^2 + (\frac{z_2}{2})^2], \\ (1 - c^2) + (1 + c^2)z_1 + (1 - c^2)(\frac{z_1}{2})^2 + (1 - c^2)(\frac{z_2}{2})^2 &= 0, \\ z_1^2 + 4\frac{1+c^2}{1-c^2}z_1 + 4 + z_2^2 &= 0, \\ (z_1 + 2\frac{1+c^2}{1-c^2})^2 + z_2^2 &= 4 \left[\left(\frac{1+c^2}{1-c^2} \right)^2 - 1 \right], \\ (z_1 + 2\frac{1+c^2}{1-c^2})^2 + z_2^2 &= \left[2\sqrt{\left(\frac{1+c^2}{1-c^2} \right)^2 - 1} \right]^2. \end{aligned}$$

Dies ist die Formel einer Kreisgleichung mit dem Mittelpunkt $m(c) = -2\frac{1+c^2}{1-c^2}$ auf der reellen Achse und dem Radius $r(c) = 2\sqrt{\left(\frac{1+c^2}{1-c^2}\right)^2 - 1}$.

In \mathbb{C}^- kann man die Kreise verfolgen vom “unendlichen“ Kreis für $c = 1$ (das ist $\Im z$) kleiner werdend (der Radius nimmt stetig ab) und ineinander liegend bis zum Kreis für $c = 0$ mit dem Mittelpunkt bei -2 und dem Radius 0 . Dabei vergrößert sich der minimale Abstand eines Punktes auf dem Kreisumfang vom Koordinatenursprung von 0 auf 2 . Die Abstandsfunktion wird beschrieben durch die Beziehung

$$abst(c) = -m(c) - r(c) = 2 \left[\frac{1+c^2}{1-c^2} - \sqrt{\left(\frac{1+c^2}{1-c^2}\right)^2 - 1} \right] \geq 0.$$

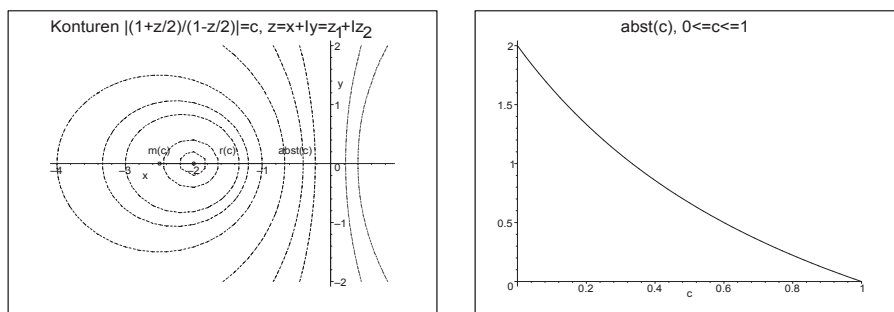


Abb. 7.20 Dateien *am_stab01.ps*, *am_stab02.ps*, Konturen $\left| \frac{1+z/2}{1-z/2} \right| = c$, $c = 0, \frac{1}{20}, \frac{1}{10}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{3}, \frac{4}{5}, 1, \frac{5}{4}, \frac{3}{2}$, sowie Abstandsfunktion $abst(c)$, $c \in [0, 1]$, zu AM(2)

Symmetrisch dazu erhalten wir in \mathbb{C}^+ für $c \geq 1$ Kreise gemäß der Kreisgleichung mit dem Mittelpunkt $m(c) = 2 \frac{c^2+1}{c^2-1}$ auf der reellen Achse und dem Radius $r(c) = 2 \sqrt{\left(\frac{c^2+1}{c^2-1}\right)^2 - 1}$. Für $c = 1$ ist das der “unendliche“ Kreis und mit wachsendem c kommen wir bei $c = +\infty$ zum Kreis mit dem Mittelpunkt bei 2 und dem Radius 0 .

Für reelle Werte z lässt sich aus dem Verlauf von $\xi_1(z) = \frac{1+z/2}{1-z/2}$ sowie des Betrags $|\xi_1(z)|$ erkennen, dass für alle $z < 0$ die Bedingung $|\xi_1(z)| < 1$ erfüllt ist, bei $z \geq 0$ nicht.

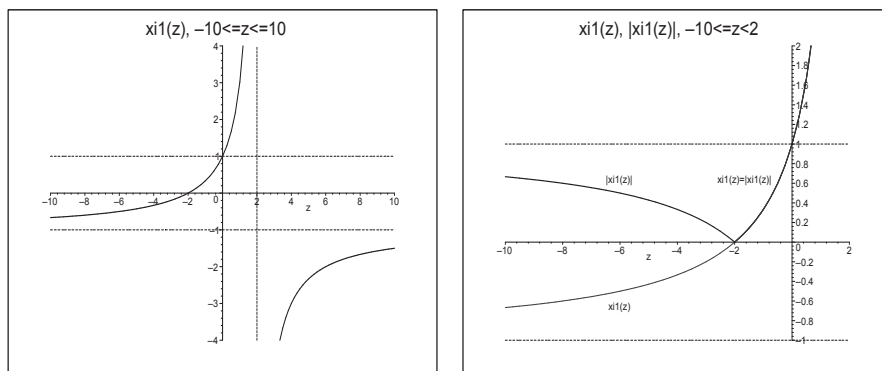


Abb. 7.21 Dateien *am_stab03.ps*, *am_stab04.ps*, Verlauf der Nullstellenfunktionen $\xi_1(z)$ bzw. $|\xi_1(z)|$, $z \in \mathbb{R}$, zu AM(2)

Nun vergleichen wir noch zu AM(5) den Verlauf der 1-Kontur mit dem weiterer Konturen sowohl bei $c = |ce^{t\varphi}| < 1$ als auch $c > 1$. Dabei treffen ähnliche Bemerkungen wie für das MSV AB(5) zu.

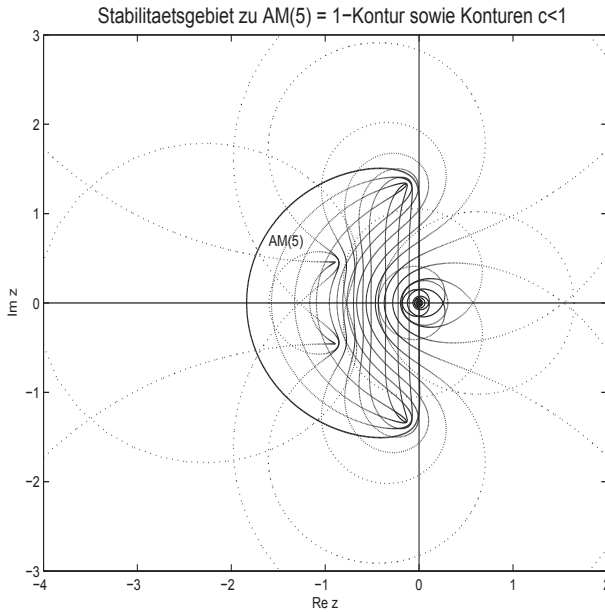


Abb. 7.22

Datei *am_stab2a.ps*,
 Stabilitätsgebiet
 zu AM(5) sowie
 weitere Konturen
 mit $c = 0.05(0.05)0.6(0.1)1$

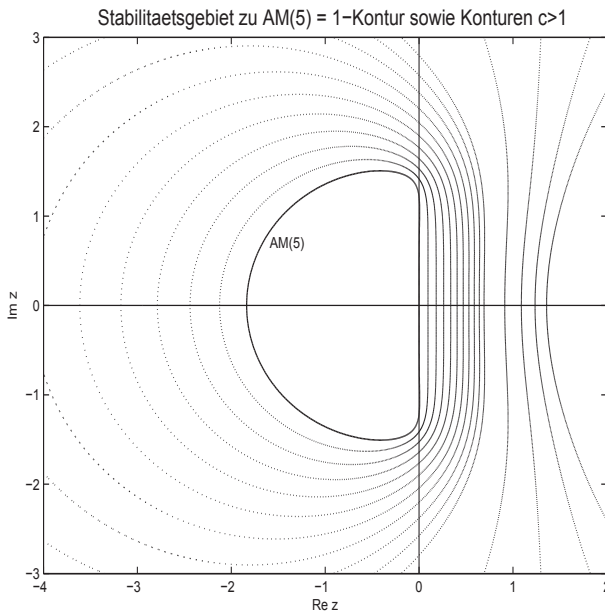


Abb. 7.23

Datei *am_stab2b.ps*,
 Stabilitätsgebiet
 zu AM(5) sowie
 weitere Konturen
 mit $c = 1(0.1)2(0.5)4$

(3) Explizite n -stufiges Adams-Nyström-Verfahren n -ter Ordnung

Die Koeffizienten der MSV AN(n), $n = 2, 3, 4$, entnehmen wir der Tabelle 6.5.

Die MSV sind konsistent. Ihre Darstellungen sind

$$\text{AN}(2) : \eta_{i+1} = \eta_{i-1} + 2h f(t_i, \eta_i),$$

$$\text{AN}(3) : \eta_{i+1} = \eta_{i-1} + \frac{h}{3} [7f(t_i, \eta_i) - 2f(t_{i-1}, \eta_{i-1}) + f(t_{i-2}, \eta_{i-2})],$$

$$\text{AN}(4) : \eta_{i+1} = \eta_{i-1} + \frac{h}{3} [8f(t_i, \eta_i) - 5f(t_{i-1}, \eta_{i-1}) + 4f(t_{i-2}, \eta_{i-2}) - f(t_{i-3}, \eta_{i-3})].$$

AN(2): Die zugehörigen charakteristischen Polynome findet man in der Beziehung

$$\phi(\xi) = \rho(\xi) - z\sigma(\xi) = (\xi^2 - 1) - z(2\xi) = \xi^2 - 2z\xi - 1, \quad z = h\lambda.$$

Das Stabilitätskriterium des Satzes 6.5 zu $\rho(\xi) = \xi^2 - 1$ ist erfüllt, denn seine beiden Nullstellen sind ± 1 . Dieser Umstand bzw. auch die Wachstumsbeschränkung (6.55) garantieren i. Allg. jedoch nicht die absolute Stabilität, genauer gesagt, dass man ein Stabilitätsgebiet findet. Es ist zwar schön, dass man theoretisch die Konvergenz von AN(n) bei $h \rightarrow 0$ hat, aber was passiert, wenn man mit einem festen $h > 0$ integriert und das noch über einen großen Bereich $[t_0, T]$.

Das stetig fallende Verhalten der Lösungen eines MSV ergibt sich aus dem Umstand, dass es eben Werte $z \in \mathbb{C}^-$ bzw. $z \in \mathbb{R}^-$ gibt, wo alle Nullstellen ξ_i von $\phi(\xi)$ betragsmäßig kleiner als Eins sind.

Wir berechnen also das Stabilitätsgebiet \mathcal{E} durch die "Randbedingung"

$$z = \frac{\rho(\xi)}{\sigma(\xi)} = \frac{\xi^2 - 1}{2\xi}, \quad \xi = e^{i\varphi},$$

und erhalten nach wenigen Umformungen

$$z = i \sin(\varphi), \quad \varphi \in [0, 2\pi),$$

und damit nur imaginäre Werte z im Intervall $[-i, i]$.

Damit ist das Gebiet \mathcal{E} , wenn es sich innerhalb der Kontur befinden soll, eine leere Menge. Da aber auch außerhalb der Kontur kein z liegt, wo $|\xi_{1,2}| < 1$ gilt, ist das Stabilitätsgebiet \mathcal{E} damit leer. Somit ist AN(2) in dem Sinne nicht stabil, dass wenn man mit einem noch so kleinen aber festen $h > 0$ lange integriert, die Näherungslösung η_i divergiert.

Natürlich kann man auch die beiden Nullstellen

$$\xi_{1,2}(z) = z \pm \sqrt{z^2 + 1}$$

der quadratischen Gleichung $\xi^2 - 2z\xi - 1 = 0$ als Funktionen von z betrachten. Wegen

$$-1 = \xi_1 \xi_2 = (z + \sqrt{z^2 + 1})(z - \sqrt{z^2 + 1})$$

folgt $\xi_1 = -\frac{1}{\xi_2}$ und somit gilt stets entweder $|\xi_1| > 1$ und $|\xi_2| < 1$ oder umgekehrt. Es gibt also keinen Bereich, wo beide Beträge kleiner als Eins sind. Für reelles z sieht man es in der folgenden Abbildung.

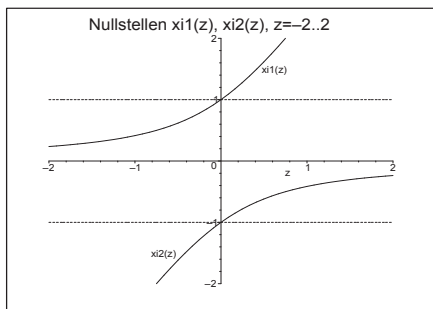


Abb. 7.24

Datei *ny_stab04.ps*,

Nullstellenkurven

$$\xi_1(z) = z + \sqrt{z^2 + 1} \text{ und}$$

$$\xi_2(z) = z - \sqrt{z^2 + 1}, \quad z \in [-2, 2]$$

Man kann AN(2) auch als Differenzengleichung

$$\eta_{i+1} = \eta_{i-1} + 2h f(t_i, \eta_i), \quad i = 1, 2, \dots, \eta_0, \eta_1 \text{ gegeben,}$$

betrachten und für das Test-AWP $x' = \lambda x$, $x(0) = 1$, einfach als rekursive Formel auswerten bzw. mit dem Ansatz $\eta_i = \xi^i$ lösen.

Für $\lambda = -1$ ist

$$\eta_{i+1} = \eta_{i-1} - 2h\eta_i, \quad i = 1, 2, \dots, n - 1, \quad t_i = ih, \quad h = T/n, \quad \eta_0, \eta_1 \text{ gegeben.}$$

Die Genauigkeitsordnung des MSV ist $\mathcal{O}(h^2)$. Die Folge der Näherungen η_i konvergiert bei $h \rightarrow 0$ sowie guten oder exakten Startwerten η_0, η_1 gegen die exakte Lösung $x(t) = e^{-t}$.

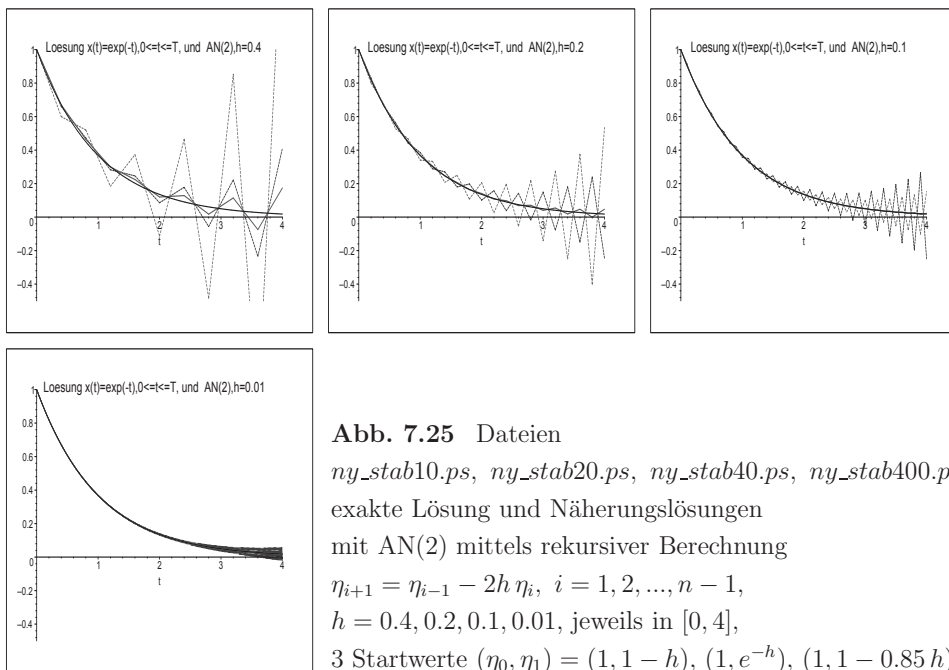


Abb. 7.25 Dateien

ny_stab10.ps, ny_stab20.ps, ny_stab40.ps, ny_stab400.ps,

exakte Lösung und Näherungslösungen

mit AN(2) mittels rekursiver Berechnung

$$\eta_{i+1} = \eta_{i-1} - 2h \eta_i, \quad i = 1, 2, \dots, n - 1,$$

$$h = 0.4, 0.2, 0.1, 0.01, \text{ jeweils in } [0, 4],$$

$$3 \text{ Startwerte } (\eta_0, \eta_1) = (1, 1 - h), (1, e^{-h}), (1, 1 - 0.85 h)$$

Wegen des leeren Stabilitätsgebiets kann die Folge der Näherungen η_i des MSV trotz guter oder exakter Startwerte η_0, η_1 über ein längeres Intervall die Asymptotik der exakten Lösung $x(t) = e^{-t}$ nicht nachvollziehen. Im Prozess der Berechnung werden diese “früher oder später“ divergieren und oszillierendes Verhalten zeigen.

Wir testen ebenfalls die drei Varianten mit den Startwerten $\eta_0 = 1$ und $\eta_1 = 1 - h$, e^{-h} , $1 - 0.85h$.

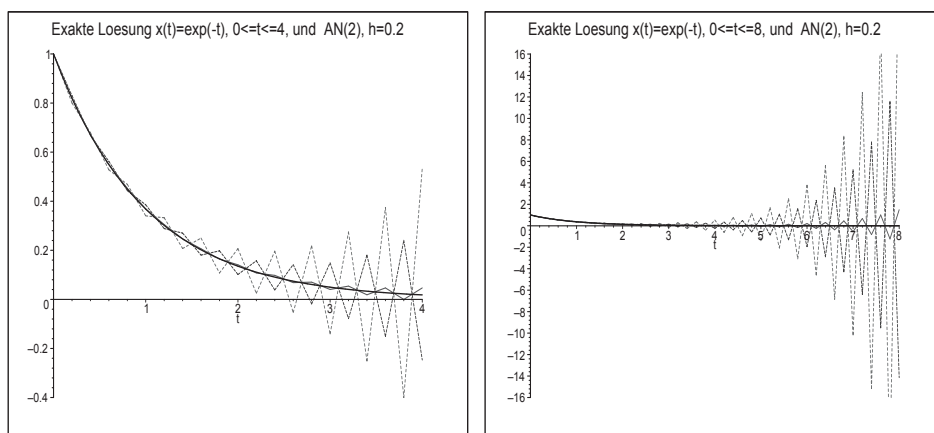


Abb. 7.26 Dateien *ny_stab01.ps*, *ny_stab02.ps*, AN(2) mittels rekursiver Berechnung $\eta_{i+1} = \eta_{i-1} - 2h\eta_i$, $i = 1, 2, \dots, n$, $h = 0.2$, in $[0, 4]$ bzw. $[0, 8]$, 3 Startwerte $(\eta_0, \eta_1) = (1, 1 - h)$, $(1, e^{-h})$, $(1, 1 - 0.85h)$

Man sollte sich also nicht von dem guten anfänglichen Verlauf der Näherungswerte irritieren lassen. Die Divergenz für $t \rightarrow \infty$ tritt ein. Bei den exakten Startwerten $(\eta_0, \eta_1) = (1, e^{-h})$ beobachtet man das Aufschaukeln der Werte erst relativ spät.

Im reellen Fall kommen wir mit dem Lösungsansatz $\eta_i = \xi^i$ zur bekannten quadratischen Gleichung $\phi(\xi) = \xi^2 + 2h\xi + 1 = 0$ und den beiden Nullstellen

$$\xi_{1,2} = -h \pm \sqrt{h^2 + 1}.$$

Die allgemeine Lösung ist nun $\eta_i = c_1 \xi_1^i + c_2 \xi_2^i$, deren Koeffizienten c_1, c_2 sich aus den AB ergeben. Wir nehmen $\eta_0 = 1$, $\eta_1 = 1 - h$ und erhalten

$$\begin{aligned} \eta_0 &= c_1 + c_2 = 1, & \eta_1 &= c_1 \xi_1 + c_2 \xi_2 = 1 - h, \\ c_1 &= \frac{h^2 + 1 + \sqrt{h^2 + 1}}{2(h^2 + 1)}, & c_2 &= \frac{h^2 + 1 - \sqrt{h^2 + 1}}{2(h^2 + 1)}, \\ \eta_i &= \frac{1}{2} \left(1 + \frac{1}{\sqrt{h_1}} \right) \left(-h + \sqrt{h_1} \right)^i + \frac{1}{2} \left(1 - \frac{1}{\sqrt{h_1}} \right) \left(-h - \sqrt{h_1} \right)^i, & h_1 &= h^2 + 1. \end{aligned}$$

Die Grafik des Polygonzugs (t_i, η_i) , $i = 1, 2, \dots, n$, mit der AB $(\eta_0, \eta_1) = (1, 1 - h)$ ist die gestrichelte “größere“ Zick-Zack-Kurve in den Bildern 7.26.

AN(3): Die zugehörigen Polynome sind

$$\phi(\xi) = \rho(\xi) - z\sigma(\xi) = (\xi^3 - \xi) - \frac{z}{3}(7\xi^2 - 2\xi + 1) = \xi^3 - \frac{7z}{3}\xi^2 + \left(\frac{2z}{3} - 1\right)\xi - \frac{z}{3}, \quad z = h\lambda.$$

Das Stabilitätskriterium des Satzes 6.5 mit $\rho(\xi) = \xi^3 - \xi$ ist erfüllt, denn seine drei Nullstellen sind $0, \pm 1$ und vom Betrag ≤ 1 . Weitere Untersuchungen ergeben eine Randkontur $\partial\mathcal{E}$ in Form einer Doppelschleife zu einem Gebiet \mathcal{E} in \mathbb{C}^+ . $\partial\mathcal{E}$ enthält als einzigen reellen Punkt $z = 0$. Sowohl im Gebiet \mathcal{E} als auch außerhalb dieses verlaufen Konturen zu den Werten $c = |e^{z\varphi}| > 1$ und $c < 1$. AN(4) zeigt analoges Verhalten. Daher kann nicht die Rede sein von einem Stabilitätsgebiet.

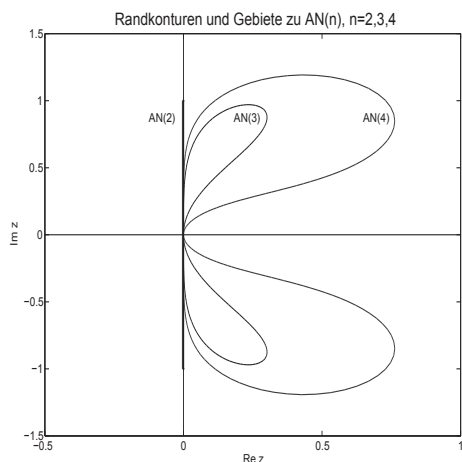


Abb. 7.27

Datei *ny_stab1.ps*,
 AN(2,3,4) mit Randkonturen $\partial\mathcal{E}$
 gemäß $z = \frac{\rho(\xi)}{\sigma(\xi)}$, $\xi = e^{i\varphi}$, $\varphi \in [0, 2\pi)$

Damit können z.B. für reelles $z < 0$ die Beträge nicht aller Nullstellenkurven $\xi_{1,2,3}$ gleichzeitig kleiner als Eins werden.

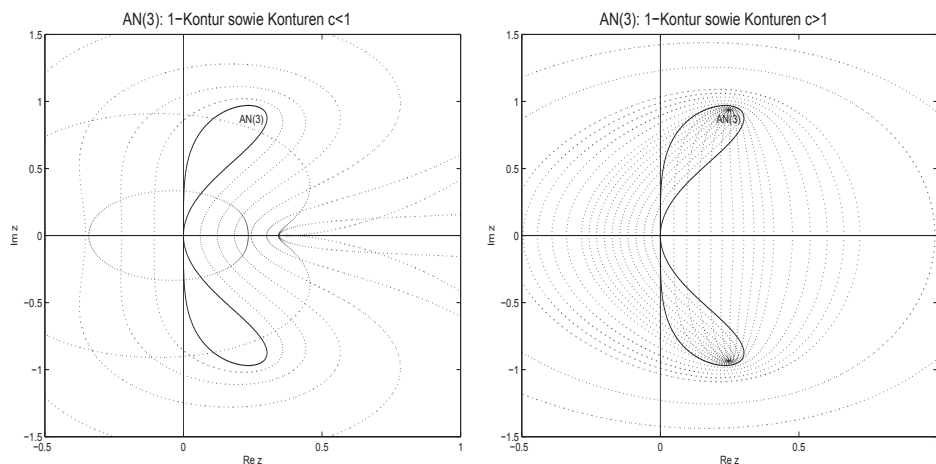


Abb. 7.28 Dateien *ny_stab3a.ps*, *ny_stab3b.ps*, AN(3) mit Konturen $\partial\mathcal{E}$
 bei $c = 0.1(0.1)1$ (links) und bei $c = 1(0.05)1.5(0.1)2(0.5)3$ (rechts)

Wir zeichnen nun noch in einer Abbildung die Kurven $\xi_{1,2,3}(z)$, $z \in \mathbb{R}$, solange diese reellwertig sind, und gehen im Fall von $\Im \xi_k(z) \neq 0$ auf den Betrag über. Es kann nur die Situation eintreten, dass in gewissen z -Intervallen alle $\xi_{1,2,3}(z)$ reell sind bzw. eine reell und zwei konjugiert komplex mit gleichem Betrag sind.

So gelten $\xi_1(z) = \overline{\xi_3(z)}$ bei $z < -0.800475$ sowie $\xi_2(z) = \overline{\xi_3(z)}$ bei $z > 0.367430$.

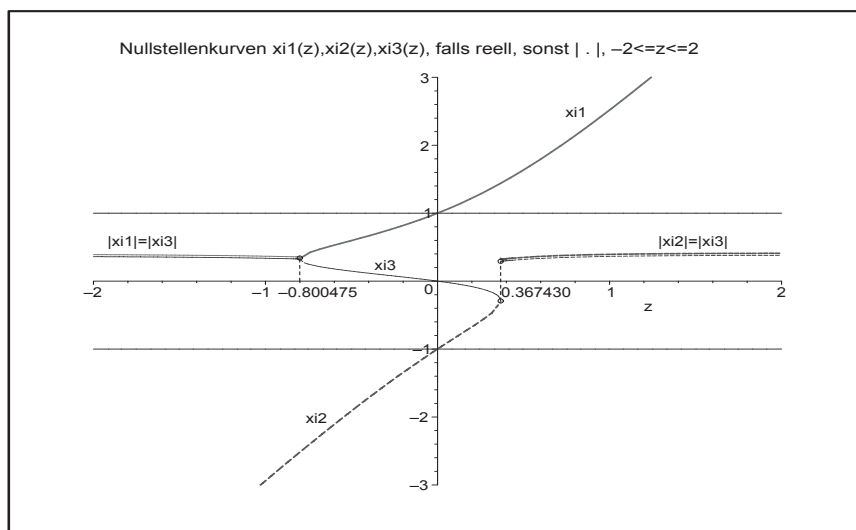


Abb. 7.29 Datei *ny_stab05.ps*, AN(3), Nullstellenkurven $\xi_{1,2,3}(z)$, $z \in [-2, 2]$, dort wo $\xi_k(z) \in \mathbb{C}$, ist $|\xi_k(z)|$ eingezeichnet

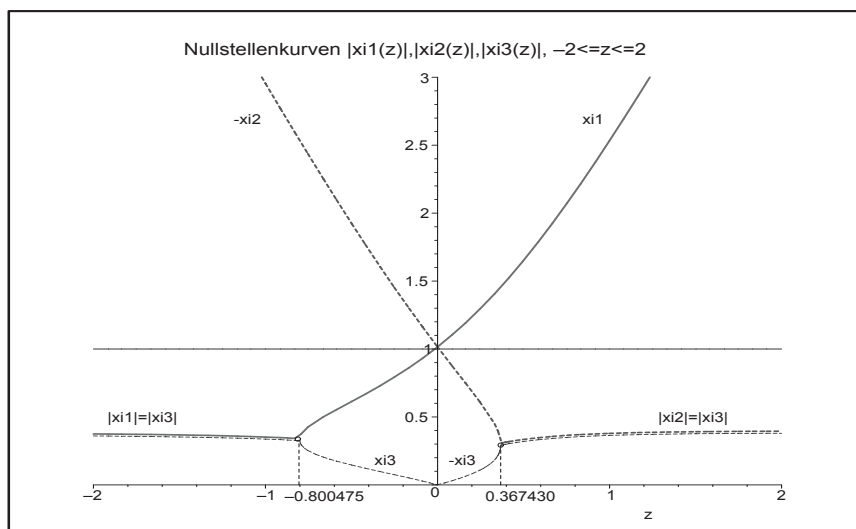


Abb. 7.30 Datei *ny_stab06.ps*, AN(3), Nullstellenkurven $|\xi_{1,2,3}(z)|$, $z \in [-2, 2]$

AN(4): Die zugehörigen Polynome sind

$$\begin{aligned}\phi(\xi) &= \rho(\xi) - z\sigma(\xi), \quad z = h\lambda \\ &= (\xi^4 - \xi^2) - \frac{z}{3}(8\xi^3 - 5\xi^2 + 4\xi - 1) \\ &= \xi^4 - \frac{8z}{3}\xi^3 + \left(\frac{5z}{3} - 1\right)\xi^2 - \frac{4z}{3}\xi + \frac{z}{3}.\end{aligned}$$

Das Stabilitätskriterium des Satzes 6.5 mit $\rho(\xi) = \xi^4 - \xi^2$ ist erfüllt, denn seine vier Nullstellen sind $1, 0, 0, -1$ und vom Betrag ≤ 1 .

Weitere Untersuchungen ergeben eine Randkontur $\partial\mathcal{E}$ in Form einer Doppelschleife zu einem Gebiet \mathcal{E} in \mathbb{C}^+ und Bemerkungen wie zu AN(3). AN(4) hat ebenfalls ein leeres Stabilitätsgebiet. In der Abb. 7.27 ist die Randkontur $\partial\mathcal{E}$ von AN(4) schon eingetragen worden.

Weitere Konturen zeigen die folgenden Grafiken.

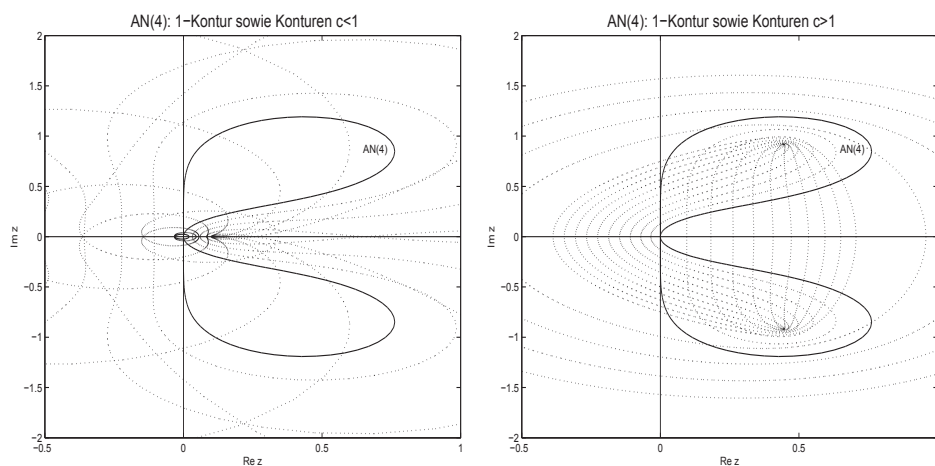


Abb. 7.31 Dateien *ny_stab4a.ps*, *ny_stab4b.ps*, AN(4) mit Konturen $\partial\mathcal{E}$ bei $c = 0.1(0.1)1, 0.55$ (links) und bei $c = 1(0.1)2(0.5)4$ (rechts)

Wir zeichnen nun noch in einer Abbildung die Kurven $\xi_{1,2,3,4}(z)$, $z \in \mathbb{R}$, solange diese reellwertig sind, und gehen im Fall von $\Im(\xi_k(z)) \neq 0$ auf den Betrag über. Es tritt nur die Situation ein, dass in gewissen z -Intervallen alle $\xi_{1,2,3,4}(z)$ reell sind bzw. zwei reell und zwei konjugiert komplex mit gleichen Betrag sind.

So gelten $\xi_1(z) = \overline{\xi_2(z)}$ bei $z < -0.316279$, $\xi_3(z) = \overline{\xi_2(z)}$ für $z \in (-0.316279, 0)$, für $z \in [0, 0.107322)$ alle $\xi_k(z)$ reell sowie $\xi_3(z) = \overline{\xi_4(z)}$ bei $z > 0.107322$.

Die meisten Veränderungen erfährt die Nullstellenfunktion $\xi_3(z)$. Es gilt zwar stets $|\xi_3(z)| < 1$, aber $\xi_3(z)$ ist erst positiv, dann komplex, dann Null und weiter negativ, sowie rechts wieder komplex.

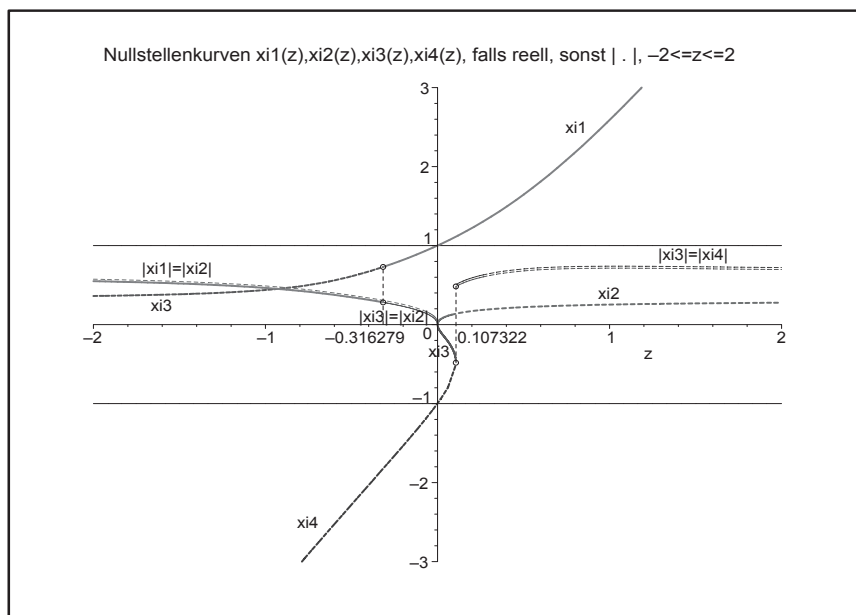


Abb. 7.32 Datei *ny_stab07.ps*, Nullstellenkurven $\xi_{1,2,3,4}(z)$, $z \in [-2, 2]$, dort wo $\xi_k(z) \in \mathbb{C}$, ist $|\xi_k(z)|$ gezeichnet

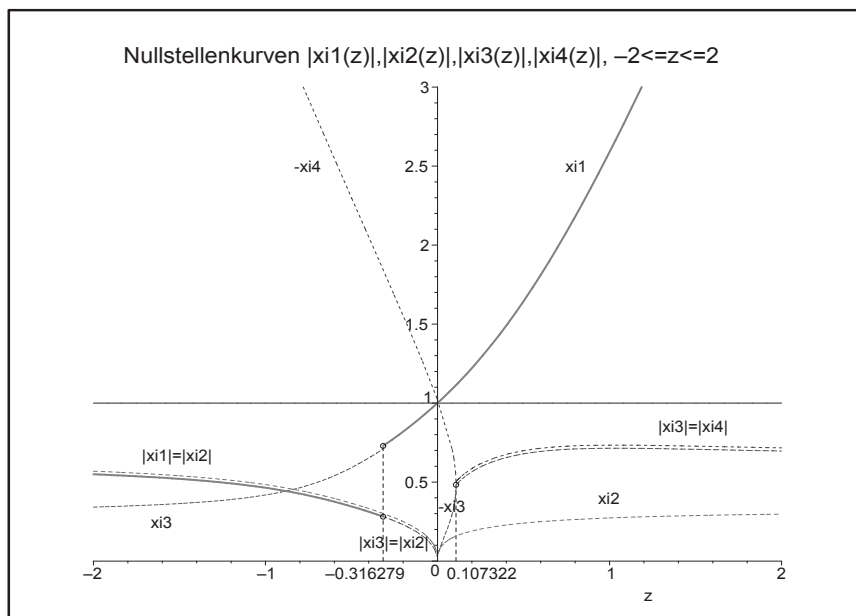


Abb. 7.33 Datei *ny_stab08.ps*, Nullstellenkurven $|\xi_{1,2,3,4}(z)|$, $z \in [-2, 2]$, dort wo $\xi_k(z) \in \mathbb{R}^-$, ist $|\xi_k(z)| = -\xi_k(z)$ eingezeichnet

(4) Implizites n -stufiges BDF-Verfahren n -ter Ordnung

Die Koeffizienten der MSV BDF(n), $n = 1, 2, \dots, 6$, entnehmen wir der Tabelle 6.7.

```
% bdf_stab.m
% BDF(n), Stabilitaetsgebiete
% Parameterschema zum BDF
m_alpha = [ 1  -1  0  0  0  0  0;
            3  -4  1  0  0  0  0;
            11 -18  9  -2  0  0  0;
            25 -48  36 -16  3  0  0;
            137 -300 300 -200 75 -12  0;
            147 -360 450 -400 225 -72 10];
m_beta = [ 1 0 0 0 0 0 0;
           2 0 0 0 0 0 0;
           6 0 0 0 0 0 0;
           12 0 0 0 0 0 0;
           60 0 0 0 0 0 0;
           60 0 0 0 0 0 0];

I = sqrt(-1);
a = 12; % 1
b = 12; % 1.5
ta = [-a,a]; tb = [-b,b]; plot(ta, [0,0], 'k', [0,0], tb, 'k');
axis([-a a -b b]); hold on
% Rand der Stabilitaetsgebiete
n = 6; for k=1:n
    alpha = m_alpha(k,1:k+1);
    beta = m_beta(k,1:k+1);
    m = 200;
    ...
```

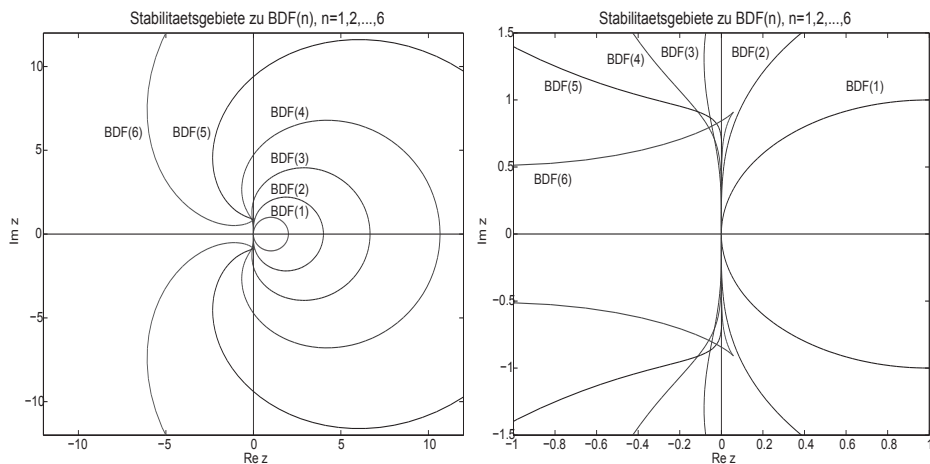


Abb. 7.34 Dateien *bdf_stab1.ps*, *bdf_stab2.ps*, Stabilitätsgebiete \mathcal{E} (rechts mit Vergrößerung nahe $z = 0$) zu BDF(n), $n = 1, 2, \dots, 6$

Nur BDF(1), das identisch mit AM(1) ist, und BDF(2) sind A-stabile MSV, weil die linke komplexe Halbebene \mathbb{C}^- zum Stabilitätsgebiet \mathcal{E} gehört. Die anderen vier BDF-Verfahren können auf entsprechende Beschränkungen der Schrittweite h führen.

Maple-Berechnung und Grafik

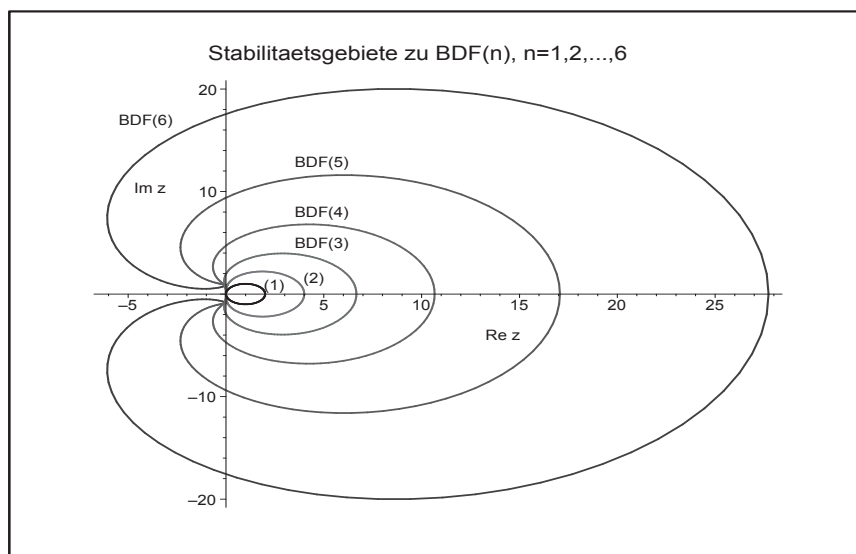


Abb. 7.35 Datei *bdf_stab01.ps*, Stabilitätsgebiete \mathcal{E} zu BDF(n), $n = 1, 2, \dots, 6$, außerhalb der Ränder $\partial\mathcal{E}$

Die BDF-Verfahren haben im Gegensatz zu den anderen MSV einige weitere Merkmale, insbesondere was den Charakter ihrer Stabilität angeht.

Zuerst erkennen wir, dass bei allen BDF-Verfahren die negative reelle Achse zum Stabilitätsgebiet gehört. Das bedeutet, dass bei $\lambda \in \mathbb{R}^-$ keine Schrittweitenbeschränkung auftritt. Mehr noch, die Stabilitätsgebiete sind nicht beschränkt. Es gibt zwei Sichten, wie man ihre "Unendlichkeit nach links" erfassen kann, einmal wie einen Winkelausschnitt von $z = 0$ aus oder anders als Halbebene, die weiter links von einer vertikalen Geraden aus beginnt.

Zur A-Stabilität in den Definitionen 7.2 und 7.3 machen wir die folgende Ergänzungen.

Definition 7.4 (a) Ein MSV (6.56) heißt $A(\alpha)$ -stabil, falls für einen Winkel α mit $0 < \alpha \leq \frac{\pi}{2}$ gilt, dass

$$\mathcal{E} \supseteq \mathbb{C}_\alpha = \{z \in \mathbb{C}^- : |\pi - \arg(z)| < \alpha\}, \quad (7.31)$$

$A(90^\circ)$ -Stabilität ist damit äquivalent zur A-Stabilität,

(b) $A(0)$ -stabil, falls ein α_0 existiert, sodass es $A(\alpha_0)$ -stabil ist, also falls $A(\alpha)$ -Stabilität für alle hinreichend kleine $\alpha > 0$ vorliegt,

(c) $A(\gamma)$ -stabil, falls ein $\gamma < 0$ existiert, sodass

$$\mathcal{E} \supseteq \mathbb{C}_\gamma = \{z \in \mathbb{C}^- : \Re z \leq \gamma\}, \quad (7.32)$$

(d) steif-stabil, falls das MSV $A(0)$ -stabil und $A(\gamma)$ -stabil ist.

Zur A-Stabilität gibt es einen allgemeinen Satz.

Satz 7.1 G. Dahlquist¹ 1956, 1963

Betrachtet werden r -stufige lineare MSV (6.8) bzw. (6.56) ($r \geq 1$).

(1) Kein explizites MSV ist A-stabil.

(2) A-stabile implizite MSV haben höchstens die Konvergenzordnung 2.

Für eine höhere Konvergenzordnung ist nur $A(\alpha)$ -Stabilität erreichbar.

Bezüglich des Satzes und der Definition 7.4 lassen sich die ersten BDF(n) wie folgt einordnen.

BDF(n) n	A-stabil	$A(\alpha)$ -stabil α	$A(\gamma)$ -stabil γ	$A(0)$ -stabil steif-stabil
1	ja	90°	0	ja
2	ja	90°	0	ja
3	nein	86.03°	-0.0833	ja
4	nein	73.35°	$-\frac{2}{3}$	ja
5	nein	51.84°	-2.327	ja
6	nein	17.84°	-6.074	ja

Tab. 7.1 Stabilität von BDF(n), $n = 1, 2, \dots, 6$ (Dezimalzahlen abgeschnitten)

Das BDF(2) ist A-stabil, was mittels der charakteristischen Polynome

$$\begin{aligned}\phi(z) &= (\alpha_1 \xi^2 + \alpha_2 \xi + \alpha_3) - z(\beta_1 \xi^2) = \rho(\xi) - z\sigma(\xi) = (3\xi^2 - 4\xi + 1) - z(2\xi^2) \\ &= (3 - 2z)\xi^2 - 4\xi + 1 = (3 - 2z)\left(\xi^2 - \frac{4}{3-2z}\xi + \frac{1}{3-2z}\right)\end{aligned}$$

oder der Nullstellen dieser gezeigt wird. Die 1-Kontur liegt in \mathbb{C}^+ , weil die zu den Größen $\xi = e^{i\varphi}$, $\varphi \in [0, 2\pi)$, gehörigen Werte $z = \frac{\rho(\xi)}{\sigma(\xi)}$ nur positiven Realteil haben. Die einzige Ausnahme bildet $\varphi = 0$ mit $\xi = 1$ und $z = 0$. Wir rechnen also

$$\begin{aligned}z &= \frac{\rho(\xi)}{\sigma(\xi)} = \frac{3\xi^2 - 4\xi + 1}{2\xi^2} = \frac{3e^{2i\varphi} - 4e^{i\varphi} + 1}{2e^{2i\varphi}} = \frac{3}{2} - 2e^{-i\varphi} + \frac{1}{2}e^{-2i\varphi} \\ &= \frac{3}{2} - 2[\cos(\varphi) - i\sin(\varphi)] + \frac{1}{2}[\cos(2\varphi) - i\sin(2\varphi)] \\ &= \frac{3}{2} - 2\cos(\varphi) + \frac{1}{2}(2\cos^2(\varphi) - 1) + i[2\sin(\varphi) - \sin(\varphi)\cos(\varphi)] \\ &= [1 - \cos(\varphi)]^2 + i\sin(\varphi)[2 - \cos(\varphi)] \\ &= \Re z + i\Im z,\end{aligned}$$

sodass $\Re z = [1 - \cos(\varphi)]^2 > 0$, $\varphi \neq 0$.

¹Germund Dahlquist (1925-2005), schwedischer Mathematiker, Theorie der gDGL und AWP

Für den Fall $z \in \mathbb{R}$ betrachten wir die beiden Nullstellen von $\phi(\xi)$ und erhalten

$$\xi_{1,2}(z) = \frac{2 \pm \sqrt{1+2z}}{\alpha_1 - \beta_1 z} = \frac{2 \pm \sqrt{1+2z}}{3-2z}.$$

Für $z \in (-\infty, -\frac{1}{2})$ sind die Werte konjugiert komplex mit dem gleichen Betrag

$$|\xi_1| = |\xi_2| = \sqrt{\left(\frac{2}{3-2z}\right)^2 + \left(\frac{2z+1}{3-2z}\right)^2} = \sqrt{\frac{(1+2z)^2 + 4}{(3-2z)^2}} = \sqrt{\frac{4z^2 + 4z + 5}{4z^2 - 6z + 9}}.$$

Da der Nenner unter der Wurzel stets größer als der Zähler ist gilt $|\xi_{1,2}| < 1$.

Für $z \geq -\frac{1}{2}$ sind die Werte reell und positiv. Es gelten die Beziehungen $\xi_{1,2}(-\frac{1}{2}) = \frac{1}{2}$, $\xi_1(0) = 1$, $\xi_2(0) = \frac{1}{3}$, $\xi_1'(z) > 0$ bei $z < \frac{3}{2}$ und $\xi_2'(z) < 0$, $\xi_2(\frac{3}{2}) = \frac{1}{16}$, was leicht nachzuprüfen ist.

Damit erhalten wir für $z \in [-\frac{1}{2}, 0)$ die gewünschte Ungleichung $0 < \xi_{1,2}(z) < 1$.

Es gibt keine Schrittweitenbeschränkung im Reellen und die A-Stabilität des Verfahrens bestätigt sich.

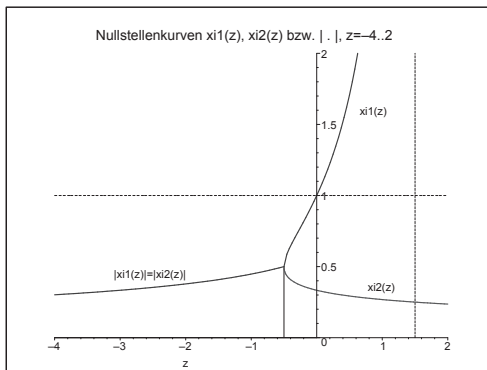


Abb. 7.36

Datei *bdf_stab20.ps*,

Verlauf der Nullstellenfunktionen

$\xi_{1,2}(z)$ bzw. $|\xi_{1,2}(z)|$ zu BDF(2)

Ab BDF(3) geht die A-Stabilität verloren. Der Rand der Stabilitätsgebiete von BDF(3) und BDF(4) startet von $z = 0$ in den Bereich \mathbb{C}^- hinein, um später die imaginäre Achse zu schneiden und im \mathbb{C}^+ -Bereich seinen weiteren Verlauf zu nehmen. Damit muss sich die Schrittweite für gewisse komplexe Werte λ entsprechend "anpassen".

Etwas anders verläuft die Startphase des Randes der Stabilitätsgebiete von BDF(5) und BDF(6), wo es von $z = 0$ in den Bereich \mathbb{C}^+ kurz hineingeht und erst nach einer großen Schleife in \mathbb{C}^- der restliche Verlauf wieder im \mathbb{C}^+ -Bereich stattfindet.

Wir fassen die Nullstellen der charakteristischen Polynome noch einmal zusammen. Da $\rho(\xi)$ immer die einfache Nullstelle 1 hat, nehmen wir nur das reduzierte Polynom $\frac{\rho(\xi)}{\xi-1}$, dessen weitere reelle oder konjugiert komplexe Nullstellen betragsmäßig kleiner als Eins sind.

BDF(n)	reduziertes Polynom $\frac{\rho(\xi)}{\xi-1}$ Nullstellen $\xi_1 = 1$ und $\xi_k, k = 2, \dots, n$	Beträge $ \xi_k , k = 2, \dots, n$
2	$3\xi - 1$ $\frac{1}{3}$	$\frac{1}{3}$
3	$11\xi^2 - 7\xi + 2$ $\frac{7 \pm \sqrt{39}i}{22} = 0.318\ 182 \pm 0.283\ 864i$	$\frac{\sqrt{22}}{11} = 0.426\ 401$
4	$25\xi^3 - 23\xi^2 + 13\xi - 3$ $\frac{1}{75}(3842 + 450\sqrt{511})^{1/3} - \frac{446}{75}(3842 + 450\sqrt{511})^{-1/3} + \frac{23}{75}$ $0.269\ 261 \pm 0.492\ 000i$	0.381 478 0.560 862
5	$137\xi^4 - 163\xi^3 + 137\xi^2 - 63\xi + 12$ $0.384\ 847 \pm 0.162\ 121i$ $0.210\ 044 \pm 0.676\ 870i$	0.417 601 0.708 711
6	$147\xi^5 - 213\xi^4 + 237\xi^3 - 163\xi^2 + 62\xi - 10$ 0.406 123 $0.145\ 275 \pm 0.851\ 070i$ $0.376\ 154 \pm 0.288\ 474i$	0.406 123 0.863 380 0.474 035

Tab. 7.2 Nullstellen zu $\frac{\rho(\xi)}{\xi-1}$ für BDF(n), $n = 2, 3, \dots, 6$ (Dezimalzahlen gerundet)

BDF(n), $n = 2, 3, \dots, 6$: Nullstellenfunktionen $\xi_{1,2,\dots,n}(z)$ für $z \in \mathbb{R}$

Der wichtige Bereich ist $z \in \mathbb{R}^-$. Für komplexes $\xi_k(z)$ werden seine Beträge gezeichnet, meistens als gestrichelte Linie. Gilt $|\xi_k(z)| = |\xi_j(z)|$, so werden ihre Grafen übereinander mit einem kleinen Abstand gezeichnet. Außerdem sind die Asymptote bei $z = \frac{\alpha_1}{\beta_1}$ und spezielle Punkte hervorgehoben.

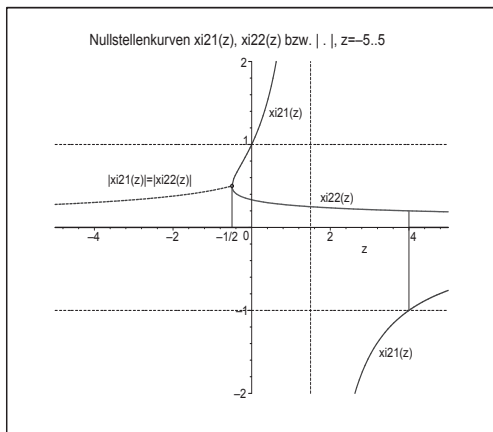


Abb. 7.37

Datei *bdf_stab22.ps*,
BDF(2): Verlauf
der Nullstellenfunktionen
 $\xi_{1,2}(z)$ bzw. $|\xi_{1,2}(z)|$,
Asymptote bei $z = \frac{3}{2}$

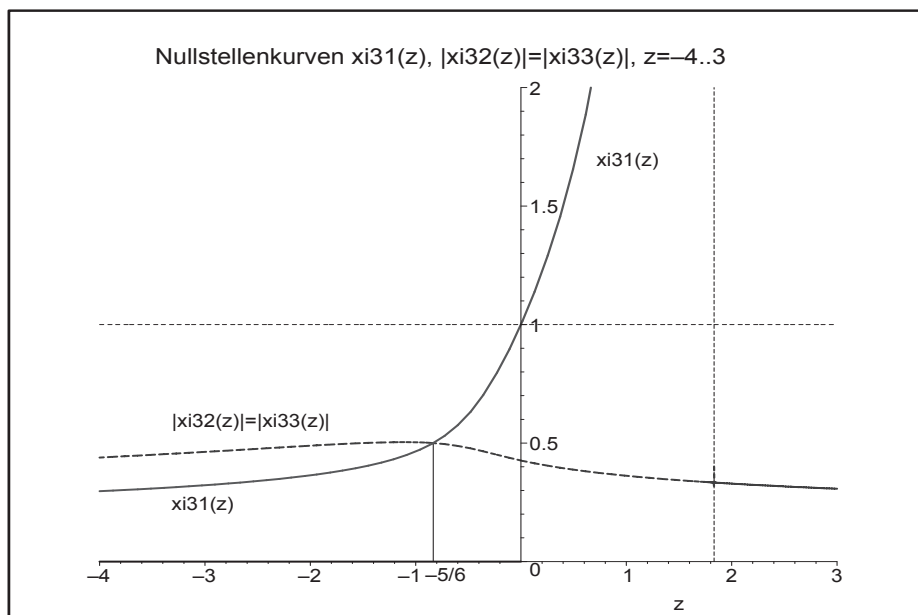


Abb. 7.38 Datei *bdf_stab23.ps*, BDF(3): Verlauf der Nullstellenfunktionen $\xi_1(z) \in \mathbb{R}$ und $|\xi_2(z)| = |\xi_3(z)|$ bei $\xi_{2,3}(z) \in \mathbb{C}$, Asymptote bei $z = \frac{11}{6}$

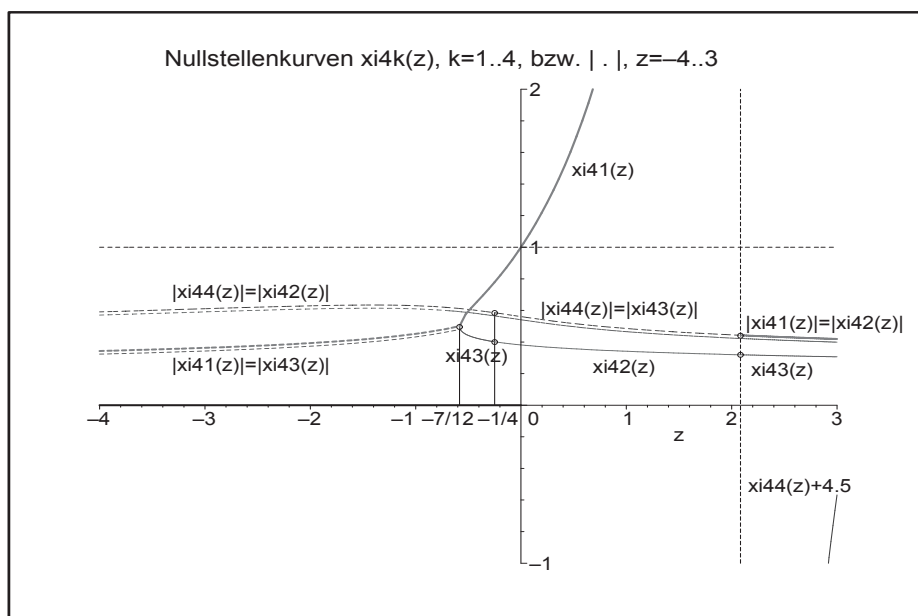


Abb. 7.39 Datei *bdf_stab24.ps*, BDF(4): Verlauf der Nullstellenfunktionen $\xi_k(z) \in \mathbb{R}$ und $|\xi_k(z)|$ bei $\xi_k(z) \in \mathbb{C}$, Asymptote bei $z = \frac{25}{12}$

Man erkennt bei BDF(4) die Bedingung $|\xi_k(z)| < 1$ bei $z < 0$, das Verhalten der Nullstellen $\xi_k(z)$ in Abhängigkeit von z und die Übergänge zwischen reellen und komplexen Abschnitten. So gelten an den markanten Stellen u.a. die Werte bzw. an den Stellen $z \pm$ die einseitigen Grenzwerte

$$|\xi_1(-\frac{7}{12}-)| = |\xi_3(-\frac{7}{12}-)| = \xi_1(-\frac{7}{12}+) = \xi_3(-\frac{7}{12}+) = \frac{1}{2},$$

$$\xi_3(-\frac{1}{4}-) = \xi_2(-\frac{1}{4}+) = 0.401, \quad |\xi_4(-\frac{1}{4})| = |\xi_2(-\frac{1}{4}-)| = |\xi_3(-\frac{1}{4}+)| = 0.585,$$

$$|\xi_4(\frac{25}{12}-)| = |\xi_3(\frac{25}{12}-)| = |\xi_1(\frac{25}{12}+)| = |\xi_2(\frac{25}{12}+)| = 0.442, \quad \xi_2(\frac{25}{12}-) = \xi_3(\frac{25}{12}+) = 0.319.$$

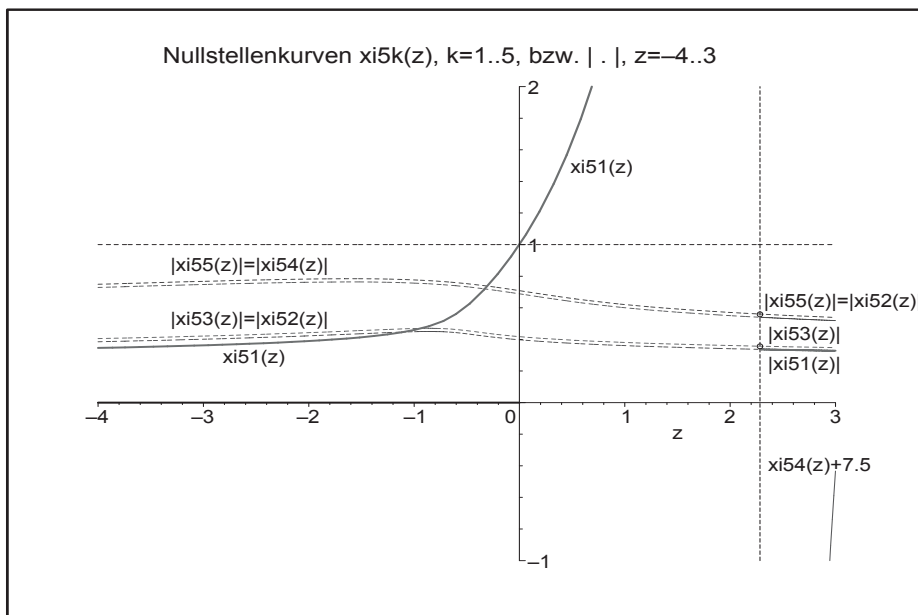


Abb. 7.40 Datei *bdf_stab25.ps*, BDF(5): Verlauf der Nullstellenfunktionen

$$\xi_k(z) \in \mathbb{R} \text{ und } |\xi_k(z)| \text{ bei } \xi_k(z) \in \mathbb{C}, \text{ Asymptote bei } z = \frac{137}{60}$$

Die Nullstellenkurven zu BDF(5) sind einfacher in ihrem Verhalten als bei BDF(4). Ansonsten gilt auch für BDF(5) analoges wie zu BDF(4).

$$|\xi_3(\frac{137}{60})| = |\xi_2(\frac{137}{60}-)| = |\xi_1(\frac{137}{60}+)| = 0.357,$$

$$|\xi_5(\frac{137}{60})| = |\xi_4(\frac{137}{60}-)| = |\xi_2(\frac{137}{60}+)| = 0.560,$$

$$\xi_1(-\frac{47}{60}) = \frac{1}{2}, \quad \xi_1(-1) = 0.459\,323,$$

$$\xi_1(-0.931\,560) = |\xi_{2,3}(-0.931\,560)| = 0.469\,786,$$

$$\xi_1(-0.304\,214) = |\xi_{4,5}(-0.304\,214)| = 0.737\,893.$$

Zu BDF(5) machen wir mit Maple noch einige Darstellungen bzgl. der Art der Stabilität und plotten die zugehörigen Gebiete.

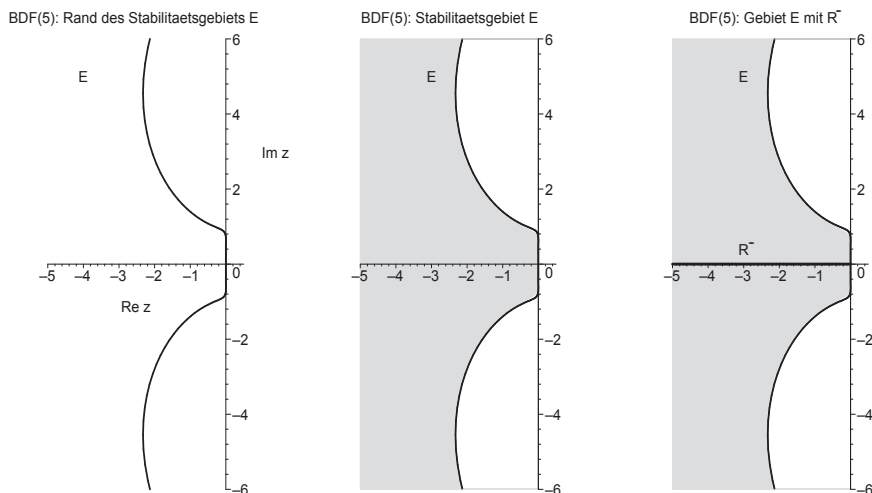


Abb. 7.41 Dateien *bdf_stab02.ps*, *bdf_stab03.ps*, *bdf_stab04.ps*, BDF(5): Stabilitätsgebiet \mathcal{E} , Rand $\partial\mathcal{E}$ sowie mit Stab.intervall $\mathbb{R}^- = (-\infty, 0)$

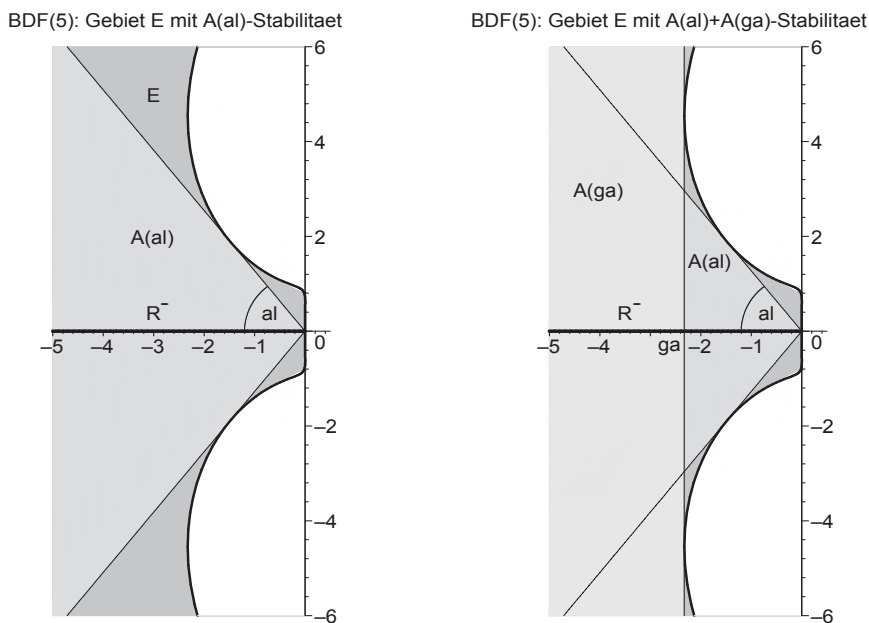


Abb. 7.42 Dateien *bdf_stab05.ps*, *bdf_stab06.ps*, BDF(5): Stabilitätsgebiet \mathcal{E} sowie $A(\alpha)$ - und $A(\gamma)$ -Stabilität

Wir kommen nun zur Darstellung der Nullstellenfunktionen von BDF(6).

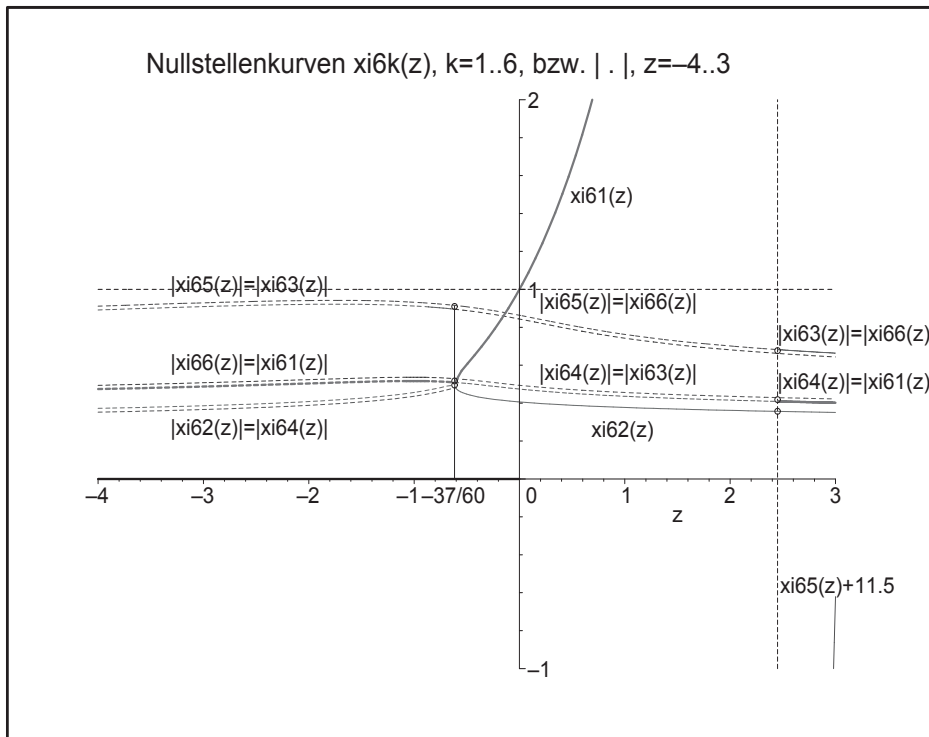


Abb. 7.43 Datei *bdf_stab26.ps*, BDF(6): Verlauf der Nullstellenfunktionen $\xi_k(z) \in \mathbb{R}$ und $|\xi_k(z)|$ bei $\xi_k(z) \in \mathbb{C}$, Asymptote bei $z = \frac{147}{60}$

Die Nullstellenkurven zu BDF(6) sind in ihrem Verhalten ähnlich wie bei BDF(4), nur dass noch ein Paar konjugiert komplexer hinzukommt.

Ansonsten gilt bzgl. der Nullstellen das bisher Gesagte.

$$|\xi_2(-\frac{37}{60}-)| = |\xi_4(-\frac{37}{60}-)| = \xi_1(-\frac{37}{60}+) = \xi_2(-\frac{37}{60}+) = \frac{1}{2},$$

$$|\xi_6(-\frac{37}{60}-)| = |\xi_1(-\frac{37}{60}-)| = |\xi_4(-\frac{37}{60}+)| = |\xi_3(-\frac{37}{60}+)| = 0.509,$$

$$|\xi_5(-\frac{37}{60})| = |\xi_3(-\frac{37}{60}-)| = |\xi_6(-\frac{37}{60}+)| = 0.915,$$

$$\xi_2(\frac{147}{60}) = 0.357,$$

$$|\xi_4(\frac{147}{60})| = |\xi_3(\frac{147}{60}-)| = |\xi_1(\frac{147}{60}+)| = 0.409,$$

$$|\xi_6(\frac{147}{60})| = |\xi_5(\frac{147}{60}-)| = |\xi_3(\frac{147}{60}+)| = 0.683.$$

(5) **Implizites 2-stufiges Milne-Simpson-Verfahren 4-ter Ordnung**

Dieses MSV

$$\eta_{i+1} - \eta_{i-1} = \frac{h}{3} (f_{i+1} + 4f_i + f_{i-1})$$

ist bzgl. seiner Genauigkeit im Abschnitt 6.2 untersucht worden.

Die zugehörigen charakteristischen Polynome findet man in der Beziehung

$$\phi(\xi) = \rho(\xi) - z\sigma(\xi) = (\xi^2 - 1) - z\left(\frac{1}{3}\xi^2 + \frac{4}{3}\xi + \frac{1}{3}\right) = \left(1 - \frac{z}{3}\right)\xi^2 - \frac{4z}{3}\xi - 1 - \frac{z}{3},$$

wobei $z = h\lambda$. Das Stabilitätskriterium des Satzes 6.5 zu $\rho(\xi) = \xi^2 - 1$ ist erfüllt, denn seine beiden Nullstellen sind ± 1 . Dieser Umstand bzw. auch die Wachstumsbeschränkung (6.55) garantieren i. Allg. jedoch nicht die absolute Stabilität und damit ein akzeptables Langzeitverhalten der Näherungslösung.

Wir berechnen das Stabilitätsgebiet \mathcal{E} durch die Randbedingung

$$z = \frac{\rho(\xi)}{\sigma(\xi)} = \frac{3(\xi^2 - 1)}{\xi^2 + 4\xi + 1}, \quad \xi = e^{i\varphi},$$

und erhalten nach wenigen Umformungen

$$z = i \frac{3 \sin(\varphi)}{2 + \cos(\varphi)}, \quad \varphi \in [0, 2\pi),$$

und damit nur imaginäre Werte z im Intervall $[-i\sqrt{3}, i\sqrt{3}]$.

Damit ist das Gebiet \mathcal{E} , wenn es sich innerhalb der Kontur befinden soll, eine leere Menge. Da aber auch außerhalb der Kontur kein z liegt, wo $|\xi_{1,2}(z)| < 1$ gilt, ist das Stabilitätsgebiet \mathcal{E} daher leer. Somit ist MS(4) nicht absolut stabil. Wenn man mit einem noch so kleinen aber festen $h > 0$ lange integriert, wird die Näherungslösung η_i divergieren.

Natürlich kann man auch die beiden Nullstellen

$$\xi_{1,2}(z) = \frac{2z \pm \sqrt{3z^2 + 9}}{3 - z}$$

der quadratischen Gleichung $(1 - \frac{z}{3})\xi^2 - \frac{4z}{3}\xi - (1 + \frac{z}{3}) = 0$ als Funktionen von z betrachten. Es gelten

$$\xi_1(0) = 1, \quad \xi_2(0) = -1$$

und für andere z -Werte stets entweder $|\xi_1(z)| > 1$ oder $\xi_2(z) < -1$.

Es gibt also keinen Bereich, wo beide Beträge kleiner als Eins sind. Für reelle Argumente z sieht man es in der folgenden Abbildung.

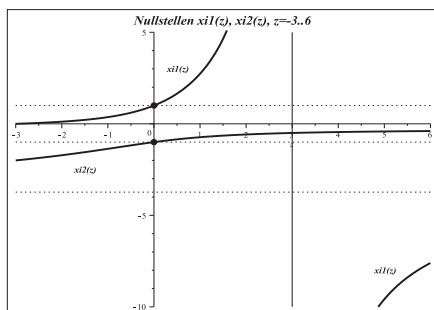


Abb. 7.44

Datei *ms_stab04.ps*,

MS(4): Nullstellenkurven

$$\xi_1(z) = \frac{2z + \sqrt{3z^2 + 9}}{3 - z} \quad \text{und}$$

$$\xi_2(z) = \frac{2z - \sqrt{3z^2 + 9}}{3 - z}, \quad z \in [-3, 6]$$

Man kann das MS(4) auch als Differenzengleichung

$$\eta_{i+1} = \eta_{i-1} + \frac{h}{3} (f(t_{i+1}, \eta_{i+1}) + 4f(t_i, \eta_i) + f(t_{i-1}, \eta_{i-1})), \quad i = 1, 2, \dots, \eta_0, \eta_1 \text{ geg.},$$

betrachten und für das Test-AWP $x' = \lambda x$, $x(0) = 1$, einfach als rekursive Formel auswerten bzw. mit dem Ansatz $\eta_i = \xi^i$ lösen.

Für $\lambda = -1$ ist

$$\eta_{i+1} = \frac{1}{1 + h/3} \left[-\frac{4h}{3} \eta_i + \left(1 - \frac{h}{3}\right) \eta_{i-1} \right], \quad i = 1, 2, \dots, n-1, \quad t_i = ih, \quad h = T/n.$$

Die Genauigkeitsordnung des MSV ist $\mathcal{O}(h^4)$. Die Folge der Näherungen η_i konvergiert bei $h \rightarrow 0$ sowie guten oder exakten Startwerten η_0, η_1 gegen die exakte Lösung $x(t) = e^{-t}$.

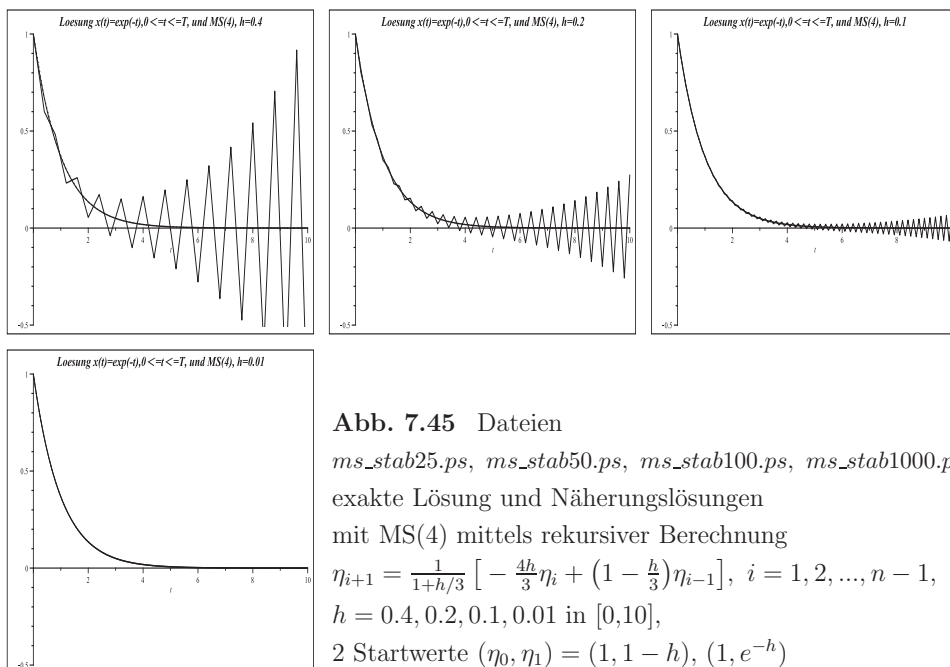


Abb. 7.45 Dateien

ms_stab25.ps, *ms_stab50.ps*, *ms_stab100.ps*, *ms_stab1000.ps*,

exakte Lösung und Näherungslösungen

mit MS(4) mittels rekursiver Berechnung

$$\eta_{i+1} = \frac{1}{1 + h/3} \left[-\frac{4h}{3} \eta_i + \left(1 - \frac{h}{3}\right) \eta_{i-1} \right], \quad i = 1, 2, \dots, n-1,$$

$h = 0.4, 0.2, 0.1, 0.01$ in $[0, 10]$,

2 Startwerte $(\eta_0, \eta_1) = (1, 1 - h), (1, e^{-h})$

Mit den exakten Startwerten $(\eta_0, \eta_1) = (1, e^{-h})$ sind hier grafisch keine Abweichungen von $x(t)$ erkennbar.

Wegen des leeren Stabilitätsgebiets kann die Folge der Näherungen η_i des MSV trotz guter oder exakter Startwerte η_0, η_1 über ein längeres Intervall die Asymptotik der exakten Lösung $x(t) = e^{-t}$ nicht nachvollziehen. Im Prozess der Berechnung der Größen werden diese "früher oder später" wachsend oszillierendes Verhalten zeigen, auch wenn es bei den AB $(\eta_0, \eta_1) = (1, e^{-h})$ erst nach sehr vielen Schritten bemerkbar wird.

Wir testen ebenfalls die zwei Varianten mit den Startwerten $\eta_0 = 1$ und $\eta_1 = 1-h, e^{-h}$.

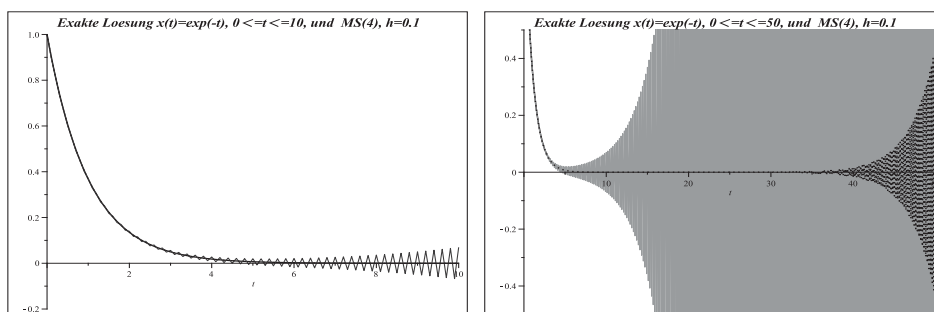


Abb. 7.46 Dateien *ms_stab01.ps*, *ms_stab02.ps*, MS(4) mittels rekursiver Berechnung

$$\eta_{i+1} = \frac{1}{1+h/3} \left[-\frac{4h}{3}\eta_i + \left(1 - \frac{h}{3}\right)\eta_{i-1} \right], \quad i = 1, 2, \dots, n, \quad h = 0.1,$$

in $[0, 10]$ bzw. $[0, 50]$, 2 Startwerte $(\eta_0, \eta_1) = (1, 1-h), (1, e^{-h})$

Man sollte sich also nicht von dem guten anfänglichen Verlauf der Näherungswerte irritieren lassen. Die Abweichungen werden immer größer.

Wir betrachten das AWP und das MS(4) im reellen Fall mit der Schrittweite $h \geq 0$ sowie den Parametern $\lambda = -1$ oder $\lambda = 1$.

(1) $\lambda = -1$, übliche Test-DGI

Mit dem Lösungsansatz $\eta_i = \xi^i$ kommen wir zur bekannten quadratischen Gleichung $\phi(\xi) = \xi^2 + \frac{4h}{3+h}\xi - \frac{3-h}{3+h} = 0$ und den beiden Nullstellen

$$\xi_{1,2}(h) = \frac{-2h \pm \sqrt{3h^2 + 9}}{3+h}, \quad h \geq 0, \quad \xi_{1,2}(0) = 1.$$

Wie kann man die beiden Funktionen charakterisieren?
 $|\xi_{1,2}(h)| < 1$ kann nicht gleichzeitig auftreten. Es gelten

$$\xi_1(h) \in (-2 + \sqrt{3}, 1], \quad \lim_{h \rightarrow \infty} \xi_1(h) = -2 + \sqrt{3}, \quad \lim_{i \rightarrow \infty} \xi_1(h)^i = 0 \text{ bei } h > 0,$$

mehr noch

$$\lim_{h \rightarrow 0} \xi_1(h)^{1/h} = e^{-1} = x(1),$$

was wie erwartet mit der Konvergenz des MSV im Punkt $t = 1$ korrespondiert. $\xi_1(h)$ ist der reguläre oder gutartige Anteil der allgemeinen Lösung des Differenzenverfahrens.

Anders ist es bei $\xi_2(h)$. Hier haben wir

$$\xi_2(h) \in (-2 - \sqrt{3}, -1], \quad \lim_{h \rightarrow \infty} \xi_2(h) = -2 - \sqrt{3}, \quad \lim_{i \rightarrow \infty} \xi_1(h)^i = \pm \infty \text{ bei } h > 0,$$

weiterhin

$$\lim_{h \rightarrow 0} \xi_2(h)^{1/h} \in [-1..1], \text{ d.h. keinen Grenzwert,}$$

$$\lim_{h \rightarrow 0} |\xi_2(h)|^{1/h} = e^{1/3} \neq x(1), \text{ aber endlich.}$$

$\xi_2(h)$ ist also der parasitäre oder “böartige“ Anteil der allgemeinen Lösung und kann diese anfangs nur dadurch wenig beeinflussen, weil sein Vorfaktor betragsmäßig klein ist. Aber später verursacht er wachsende Oszillationen.

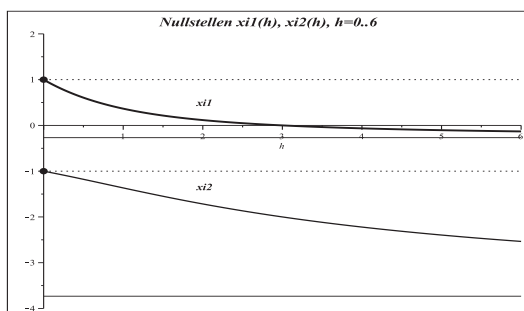


Abb. 7.47

Datei *ms_stab06.ps*,

Nullstellenfunktionen

$\xi_{1,2}(h)$, $h \geq 0$

Die allgemeine Lösung ist $\eta_i = c_1 \xi_1^i + c_2 \xi_2^i$, deren Koeffizienten c_1, c_2 sich aus den AB ergeben. Wir nehmen $\eta_0 = 1$, $\eta_1 = 1 - h$ und erhalten

$$\eta_0 = c_1 + c_2 = 1, \quad \eta_1 = c_1 \xi_1 + c_2 \xi_2 = 1 - h,$$

$$c_1 = \frac{h_1 + \sqrt{h_1}(3 - h^2)}{2h_1}, \quad c_2 = \frac{h_1 - \sqrt{h_1}(3 - h^2)}{2h_1}, \quad h_1 = 9 + 3h^2,$$

$$\eta_i = \frac{h_1 + \sqrt{h_1}(3 - h^2)}{2h_1} \left(\frac{-2h + \sqrt{h_1}}{3 + h} \right)^i + \frac{h_1 - \sqrt{h_1}(3 - h^2)}{2h_1} \left(\frac{-2h - \sqrt{h_1}}{3 + h} \right)^i.$$

Die Grafik des Polygonzugs (t_i, η_i) , $i = 1, 2, \dots, n$, mit $\eta_i = c_1 \xi_1^i + c_2 \xi_2^i$ und der AB $(\eta_0, \eta_1) = (1, 1 - h)$ ist die “größere“ Zick-Zack-Kurve in den Bildern 7.46.

T=10		n=100, h=0.1	n=1000, h=0.01
	xi1(h)	0.904837367826885	0.990049833748617
	xi2(h)	-1.033869625891401	-1.003338870293468
	c1	0.997504848415692	0.999975000486098
	c2	0.002495151584307	0.000024999513901

t[i]= 0	eta[0]	1	1
h	eta[1]	0.9	0.99
2h	eta[2]	0.8193548387096777	0.980199335548172
...			
(n-1)h	eta[n-1]	-0.0674351903404966	-0.000652572673740
nh=T=10	eta[n]	0.0698162257698247	0.000746158488349
exp(-T)		0.0000453999297624	

Der Koeffizient c_2 beim parasitären Lösungsanteil $\xi_2(h)$ tendiert gegen Null sowohl für $h \rightarrow 0$ als auch bei noch genaueren AB.

(2) $\lambda = 1$, Test-DGI mit wachsender Lösung

Der Ansatz $\eta_i = \xi^i$ liefert hier die quadratische Gleichung $\phi(\xi) = \xi^2 - \frac{4h}{3-h}\xi - \frac{3+h}{3-h} = 0$ und die beiden Nullstellen

$$\hat{\xi}_{1,2}(h) = \frac{2h \pm \sqrt{3h^2 + 9}}{3-h}, \quad h \geq 0, \quad \hat{\xi}_{1,2}(0) = 1.$$

Welche Eigenschaften haben diese beiden Funktionen?

$|\hat{\xi}_{1,2}(h)| < 1$ kann nicht gleichzeitig sein. Bei $h = 3$ gibt es eine Polstelle, die eigentlich eine Beschränkung des Schrittweite darstellt. Es gelten noch

$$\begin{aligned} \hat{\xi}_1(h) &\in [1, +\infty) \cup (-\infty, -2 - \sqrt{3}), \quad \lim_{h \rightarrow 3^\mp} \hat{\xi}_1(h) = \pm\infty, \\ \lim_{h \rightarrow \infty} \hat{\xi}_1(h) &= -2 - \sqrt{3}, \quad \lim_{i \rightarrow \infty} \hat{\xi}_1(h)^i = +\infty \text{ bei } h \in (0, 3), \end{aligned}$$

mehr noch

$$\lim_{h \rightarrow 0} \hat{\xi}_1(h)^{1/h} = e = x(1),$$

was wie erwartet mit der Konvergenz des MSV im Punkt $t = 1$ korrespondiert.

$\hat{\xi}_1(h)$ ist der reguläre Anteil der allgemeinen Lösung des Differenzenverfahrens und ist für das Wachstum zuständig.

Anders bei $\hat{\xi}_2(h)$. Hier haben wir

$$\hat{\xi}_2(h) \in [-1, -2 + \sqrt{3}), \quad \lim_{h \rightarrow \infty} \hat{\xi}_2(h) = -2 + \sqrt{3}, \quad \lim_{i \rightarrow \infty} \hat{\xi}_2(h)^i = 0 \text{ bei } h > 0,$$

weiterhin

$$\begin{aligned} \lim_{h \rightarrow 0} \hat{\xi}_2(h)^{1/h} &\in [-1..1], \quad \text{d.h. keinen Grenzwert,} \\ \lim_{h \rightarrow 0} |\hat{\xi}_2(h)|^{1/h} &= e^{-1/3} \neq x(1), \quad \text{aber endlich,} \\ \lim_{h \rightarrow 3} \hat{\xi}_2(h) &= -1/2. \end{aligned}$$

$\hat{\xi}_2(h)$ ist der parasitäre Anteil der Lösung, aber er kann das gewünschte Wachstum der Näherungslösung η_i nicht entscheidend beeinflussen, weil er langsam verschwindet.

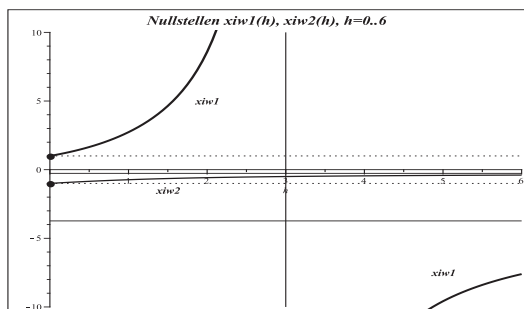


Abb. 7.48

Datei *ms_stab10.ps*,

Nullstellenfunktionen

$\hat{\xi}_{1,2}(h)$, $h \geq 0$

Die allgemeine Lösung ist nun $\eta_i = \hat{c}_1 \hat{\xi}_1^i + \hat{c}_2 \hat{\xi}_2^i$, deren Koeffizienten \hat{c}_1, \hat{c}_2 sich aus den AB ergeben. Wir nehmen $\eta_0 = 1, \eta_1 = 1 + h$ und erhalten

$$\begin{aligned} \eta_0 &= \hat{c}_1 + \hat{c}_2 = 1, \quad \eta_1 = \hat{c}_1 \hat{\xi}_1 + \hat{c}_2 \hat{\xi}_2 = 1 + h, \\ \hat{c}_1 &= c_1, \quad \hat{c}_2 = c_2, \quad h_1 = 9 + 3h^2, \\ \eta_i &= \frac{h_1 + \sqrt{h_1}(3 - h^2)}{2h_1} \left(\frac{2h + \sqrt{h_1}}{3 - h} \right)^i + \frac{h_1 - \sqrt{h_1}(3 - h^2)}{2h_1} \left(\frac{2h - \sqrt{h_1}}{3 - h} \right)^i. \end{aligned}$$

Die Grafik des Polygonzugs $(t_i, \eta_i), i = 1, 2, \dots, n$, mit $\eta_i = \hat{c}_1 \hat{\xi}_1^i + \hat{c}_2 \hat{\xi}_2^i$ und der AB $(\eta_0, \eta_1) = (1, 1 + h)$ ist eine exponentielle Wachstumsfunktion.

T=10		n=100, h=0.1	n=1000, h=0.01
	xi1w(h)	1.105170979401153	1.010050167084729
	xi2w(h)	-0.967239944918394	-0.996672240663324
	c1w	0.997504848415692	0.999975000486098
	c2w	0.002495151584307	0.000024999513901

t[i]= 0	eta[0]	1	1
h	eta[1]	1.1	1.01
2h	eta[2]	1.220689655172414	1.020200668896320
...			
(n-1)h	eta[n-1]	19880.75026463563	21806.75363756434
nh=T=10	eta[n]	21971.62843238524	22025.91515699306
exp(T)		22026.46579480672	

Der Koeffizient \hat{c}_2 beim parasitären Lösungsanteil $\hat{\xi}_2(h)$ tendiert gegen Null sowohl für $h \rightarrow 0$ als bei genaueren AB.

Der Rand des Stabilitätsgebiets ist die sogenannte 1-Kontur. Man bestimmt ihn mittels $z = \frac{\rho(\xi)}{\sigma(\xi)}$, wobei $\xi = e^{i\varphi}, \varphi \in [0, 2\pi)$. Das bedeutet zugleich die Umstellung der Formel $\hat{\xi}_{1,2}(z) = e^{i\varphi}$ nach z .

Wir rechnen also

$$z = \frac{\rho(\xi)}{\sigma(\xi)} = \frac{3(\xi^2 - 1)}{\xi^2 + 4\xi + 1} = \frac{3(e^{2i\varphi} - 1)}{e^{2i\varphi} + 4e^{i\varphi} + 1} = i \frac{3 \sin(\varphi)}{2 + \cos(\varphi)} = \Re z + i \Im z$$

mit $\Re z = 0$. Der Rand liegt somit auf der imaginären Achse und das Stabilitätsgebiet ist eine leere Menge. Seine numerische Berechnung für $\varphi \in [0, 2\pi)$ mit kleiner Schrittweite $\Delta\varphi > 0$ kann jedoch geringfügige Abweichungen zum exakten imaginären Ergebnis liefern. Wenn man die Werte

$$z = \frac{3(e^{2i\varphi} - 1)}{e^{2i\varphi} + 4e^{i\varphi} + 1}$$

z.B. für $\varphi = 0(\frac{\pi}{8})\pi$ mit `Digits=16` berechnet, erhält man

$$\begin{aligned} &0, -3.485\text{e-}17 + 0.392i, 0.783i, 1.215\text{e-}18 + 1.163i, 1.500i, \\ &-4.221\text{e-}17 + 1.713i, 1.640i, 7.304\text{e-}17 + 1.066i, 0. \end{aligned}$$

In der nachfolgenden Grafik wird die Situation durch den Maßstab in der Horizontalen sichtbar.

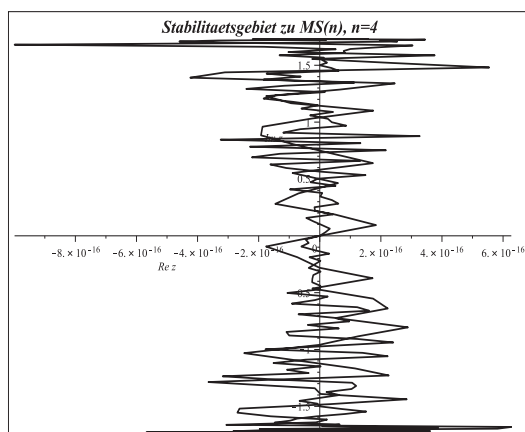


Abb. 7.49

Datei *ms_stab08.ps* (Maple 14),
 Stabilitätsgebiet zu MS(4),
 minimale Abweichungen
 von der imaginären Achse
 bei der Berechnung der 1-Kontur

Bei neueren Maple-Versionen wird das leere Stabilitätsgebiet $[-i\sqrt{3}, i\sqrt{3}]$ "genauer erfasst".

Verfolgt man einige andere Konturen, so wird man feststellen, dass es keine Bereiche gibt, wo $\xi_{1,2}(z)$ z.B. ausschließlich betragsmäßig kleiner als Eins werden. Überall verlaufen Konturen zu den Werten $c > 1$ sowie $c < 1$. Man rechnet einfach

$$z = \frac{\rho(\xi)}{\sigma(\xi)} = \frac{\rho(c e^{i\varphi})}{\sigma(c e^{i\varphi})}, \quad c = \dots, \quad \varphi \in [0, 2\pi).$$

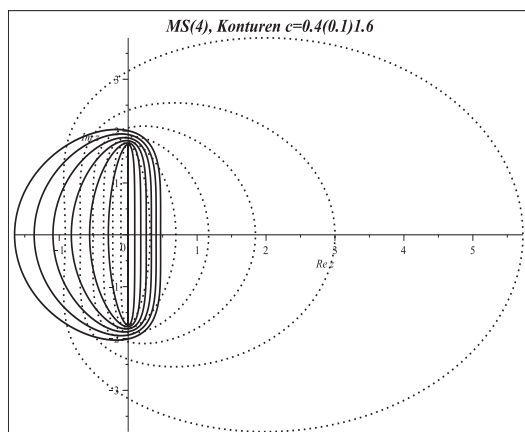


Abb. 7.50

Datei *ms_stab11.ps*,
 Konturen zu MS(4),
 $c = 0.4(0.1)0.9$ (dot),
 $c = 1$, Strecke $[-i\sqrt{3}, i\sqrt{3}]$,
 $c = 1.1(0.1)1.6$ (solid)

Übung 7.1

Schreiben Sie ein Programm zur Berechnung der komplexwertigen Funktion $z(\varphi)$, definiert durch (7.28). Stellen Sie die Stabilitätsgebiete für weitere MSV grafisch dar.

7.3 Integration steifer Systeme

Die Überlegungen der Abschnitte 7.1 und 7.2 liefern Obergrenzen an die Schrittweite h , sodass damit Instabilitäten für zu große h vermieden werden. Günstig ist es natürlich bei geringen Genauigkeitsanforderungen, gleich A-stabile Verfahren zu verwenden. Bei hoher Genauigkeit, also kleiner Toleranz, ist die Schrittweite meistens sehr klein zu wählen, sodass man damit i. Allg. unter der maximal zulässigen Grenze bleibt.

In der Test-DGI (7.1) ist der Realteil und der Betrag des Parameters $\lambda \in \mathbb{C}^-$ bzw. $\lambda \in \mathbb{R}^-$ für die Schrittweitenwahl ausschlaggebend. Für das PZV haben wir im Beispiel 7.2 (a) mit der Formel (7.13) ein wichtiges Ergebnis erhalten.

In homogenen linearen SysgDGI mit einer Matrix, die Eigenwerte mit betragskleinen Imaginärteilen hat, bestimmt der Eigenwert mit dem betragsgrößten negativen Realteil die Schrittweite (vergl. hierzu auch die Bemerkung und Merkgel 4.3).

Im Folgenden wollen wir an Beispielen ein Konzept der SWS entwerfen, welches zu einer möglichst genauen Approximation unter geringem Rechenaufwand führt. Hierzu betrachten wir zunächst das skalare AWP

$$x' = \lambda x, \quad x(t_i) = \eta_i, \quad t \geq t_i,$$

mit der exakten Lösung

$$x(t) = \Phi^{t,t_i} \eta_i = \eta_i e^{\lambda(t-t_i)}.$$

Die Anwendung eines ESV bedeutet die Rekursionsvorschrift

$$\eta_{i+1} = F(h\lambda) \eta_i, \quad i = 0, 1, \dots,$$

(vergl. Definition 7.2). Fordern wir neben der Stabilitätsbedingung $|F(h\lambda)| < 1$, dass $F(h\lambda)$ eine gute Approximation von $e^{h\lambda}$ ist, also eine entsprechende Konsistenzordnung erreicht wird, sowie evtl. Monotonieeigenschaften erfüllt werden sollen, so führt das gegebenenfalls zu einer weiteren Einschränkung der Schrittweite.

GDGI mit einem moderaten Lösungsverlauf werden als nicht steif eingestuft. Die Steifheit dagegen drückt sich in einer starken Veränderung des Lösungsverlaufs, z.B. der Lösungsfunktion $x(t)$, der Ableitung $x'(t)$ und/oder von $|f_x(t, x)|$ u.ä. aus.

Wir betrachten zuerst gDGI 2. Ordnung bzw. SysgDGI, die nicht steif sind, um dann den Charakter der Steifheit besser zu deuten. Berechnungen und Grafiken sind in Maple.

Beispiel 7.5

(a) Gegeben sei die homogene lineare gDGI 2. Ordnung

$$x''(t) + 5x'(t) + 6x(t) = 0. \tag{7.33}$$

Die charakteristischen Wurzeln sind $\lambda_1 = -2$ und $\lambda_2 = -3$, das Fundamentalsystem $\{e^{-2t}, e^{-3t}\}$, sodass ein moderater Verlauf der allgemeinen Lösung $x(t) = C_1 e^{-2t} + C_2 e^{-3t}$ bestehend aus zwei Lösungskomponenten vorliegt.

```
> de1:=diff(x(t),t,t)+5*diff(x(t),t)+6*x(t)=0:
dsolve({de1,x(0)=0,D(x)(0)=1},x(t),method=laplace);
dsolve({de1,x(0)=0,D(x)(0)=1},x(t),type=series);
```

$$x(t) = e^{(-2t)} - e^{(-3t)}$$

$$x(t) = t - \frac{5}{2}t^2 + \frac{19}{6}t^3 - \frac{65}{24}t^4 + \frac{211}{120}t^5 + O(t^6)$$

```
> loe1:=dsolve({de1,x(0)=0,D(x)(0)=1},x(t),method=laplace);
p1:=plot(rhs(loe1),t=0..4,0..0.2,color=black,thickness=2):
p2:=textplot([[1.8,0.1,'x(t)=exp(-2t)-exp(-3t)']]):
display(p1,p2);
```

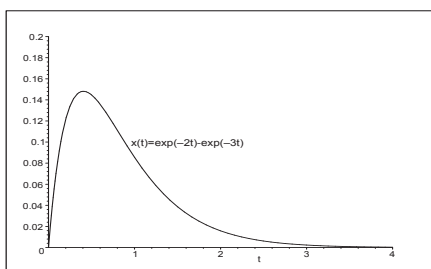


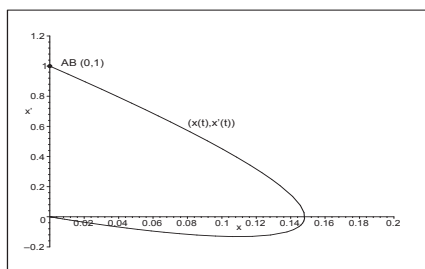
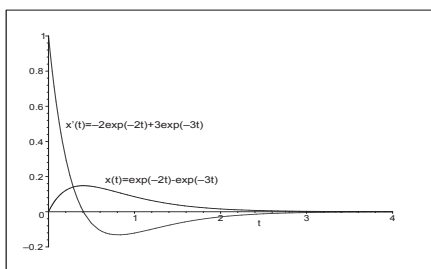
Abb. 7.51

Datei *nsteif01.ps*, AWP

AWP $x''(t) + 5x'(t) + 6x(t) = 0$, $x(0) = 0$, $x'(0) = 1$,
und Lösung $x(t) = e^{-2t} - e^{-3t}$, $t \in [0, 4]$

```
> xx:=unapply(rhs(loe1),t);
xx1:=unapply(diff(xx(t),t),t); # oder xx1:=D(xx);
> p1:=plot([xx(t),xx1(t)],t=0..4,-0.2..1,color=[black,blue],thickness=2):
p2:=textplot([[1.8,0.2,'x(t)=exp(-2t)-exp(-3t)'],
[1.3,0.5,'x'(t)=-2exp(-2t)+3exp(-3t)']]):
display(p1,p2);

> p1:=plot([xx(t),xx1(t),t=0..4],0..0.2,-0.2..1.2,
color=black,thickness=2,labels=['x','x']):
p2:=textplot([[2,0.2,'(x(t),x'(t))'],[0.03,1.05,'AB (1,0)']]):
p3:=pointplot([0,1],symbol=circle):
plots[display](p1,p2,p3);
```

Abb. 7.52 Dateien *nsteif02.ps*, *nsteif03.ps*,AWP $x''(t) + 5x'(t) + 6x(t) = 0$, $x(0) = 0$, $x'(0) = 1$,Lösung $x(t)$ und Ableitung $x'(t)$ sowie Phasenkurve $(x(t), x'(t))$, $t \in [0, 4]$

(b) Gegeben sei die autonome homogene lineare gDGL 2. Ordnung als AWP

$$x''(t) + x'(t) - x(t) = 0, \quad x(0) = 1, \quad x'(0) = 0. \quad (7.34)$$

Die charakteristischen Wurzeln sind $\lambda_{1,2} = \frac{1}{2}(-1 \pm \sqrt{5})$.

Die Lösung und ihre Ableitung lauten

$$\begin{aligned} x(t) &= \frac{1}{10}(5 + \sqrt{5})e^{t(-1+\sqrt{5})/2} + \frac{1}{10}(5 - \sqrt{5})e^{t(-1-\sqrt{5})/2}, \\ &= 0.723\,606\,798\,e^{0.618\,033\,988t} + 0.276\,393\,202\,e^{-1.618\,033\,988t}, \\ x'(t) &= \frac{\sqrt{5}}{5}(e^{t(-1+\sqrt{5})/2} - e^{t(-1-\sqrt{5})/2}) \\ &= 0.447\,213\,595\,(e^{0.618\,033\,988t} - e^{-1.618\,033\,988t}). \end{aligned}$$

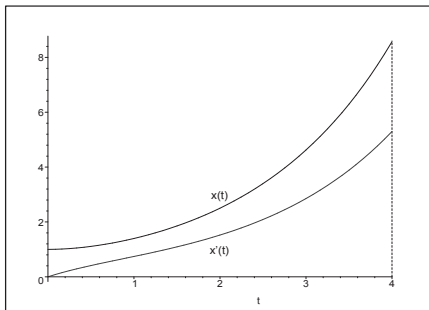


Abb. 7.53

Datei *nsteif04.ps*,

AWP $x''(t) + x'(t) - x(t) = 0$,

$x(0) = 1, x'(0) = 0$, sowie

Lösung $x(t)$ und Ableitung $x'(t)$ in $[0, 4]$

Es gilt $x(4) = 8.573\,522\,382$, $x'(4) = 5.297\,772\,797$, sodass man bei $t \in [0, 4]$ für das Phasenportrait $(x(t), x'(t))$ den Bereich $[0, 9] \times [0, 6]$ nehmen kann.

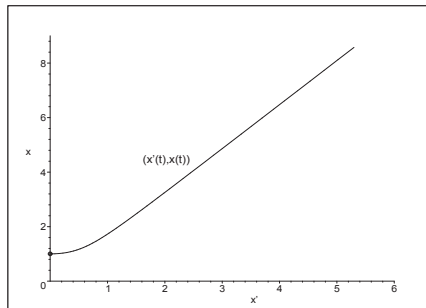
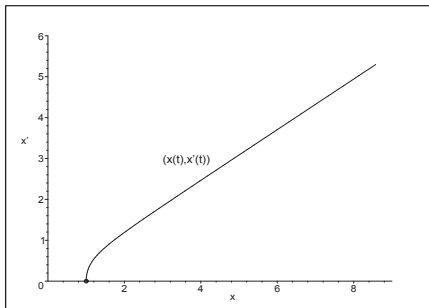


Abb. 7.54 Dateien *nsteif05.ps*, *nsteif06.ps*,

Phasenportraits $(x(t), x'(t))$ und $(x'(t), x(t))$ im jeweiligen Bereich

Man notiert die gDGL bei $x' = y$ und $\mathbf{x} = (x, x')^T$ auch als System von zwei autonomen linearen gDGL 1. Ordnung gemäß

$$\begin{aligned} x'(t) &= y(t), & x(0) &= 1, \\ y'(t) &= x(t) - y(t), & y(0) &= 0, \\ \mathbf{x}'(t) &= A\mathbf{x}(t), & A &= \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}, \mathbf{x}(0) = (1, 0)^T. \end{aligned} \quad (7.35)$$

Lösungsvarianten für (7.34) bzw. (7.35)

```
> # Loesung von gDGl 2. Ordnung
de2:=diff(x(t),t$2)+diff(x(t),t)-x(t)=0;
loe2:=dsolve({de2,x(0)=1,D(x)(0)=0},x(t));
xx:=unapply(rhs(loe2),t);
> # Loesung von SysgDGl 1. Ordnung
sys2:=diff(x(t),t)=y(t),diff(y(t),t)=x(t)-y(t);
fcns:=x(t),y(t):
init:=x(0)=1,y(0)=0:
loe2:=dsolve({sys2,init},{fcns}); # Reihenfolge x,y beachten
xx:=unapply(rhs(op(1,loe2)),t); # unapply(rhs(op(2,loe2)),t);
yy:=unapply(rhs(op(2,loe2)),t); # unapply(rhs(op(1,loe2)),t);
```

Nun wollen wir testen, inwieweit die Lösungstrajektorie eingebettet in ein Richtungsfeld bzw. die Lösung und ihre Ableitung als Phasenkurve gemeinsam mit einem Richtungsfeld (als Phasenportrait) unter Verwendung der Lösungsvariante sowie der Maple-Kommandos `phaseportrait`, `DEplot`, `dfieldplot`, `fieldplot` darstellbar sind.

(b1) GDGl 2. Ordnung und `phaseportrait`, `DEplot`, `dfieldplot`

```
> # Loesungstrajektorie = Phasenkurve (t,x(t))
p1:=phaseportrait(de2,x(t),t=0..4,[[x(0)=1,D(x)(0)=0]],stepsize=0.05,
  linecolor=black,view=[0..4,0..9]);
# analog
p1:=phaseportrait(de2,x(t),t=0..4,[[x(0)=1,D(x)(0)=0]],stepsize=0.05,
  linecolor=black,view=[0..4,0..9],scene=[t,x(t)]);
# identisch dazu ist
p1:=DEplot({de2},[x(t)],t=0..4,[[x(0)=1,D(x)(0)=0]],stepsize=0.05,
  linecolor=black,view=[0..4,0..9]);
p2:=textplot([[2,3,'x(t)']]);
display(p1,p2);
```

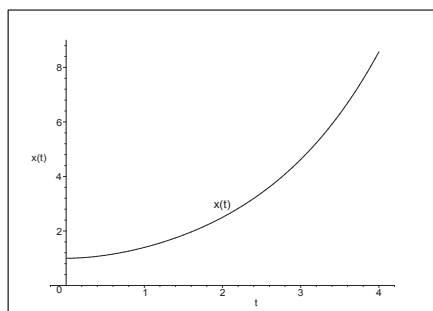


Abb. 7.55

Datei *nsteif07.ps*, AWP

$x''(t) + x'(t) - x(t) = 0$, $x(0) = 1$, $x'(0) = 0$,

Lösungstrajektorie $x(t)$ in $[0, 4]$,

hier erzeugt `DEplot` kein Richtungsfeld

Das Kommando `dfieldplot` ist nur für gDGl bzw. SysgDGl 1. Ordnung nutzbar.

```
> dfieldplot({de2},[x],t=0..4,x=0..9);
dfieldplot({de2},[x],t=0..4,[[x(0)=1,D(x)(0)=0]],x=0..9);
```

Error, (in DEtools/dfieldplot) Cannot produce plot non-autonomous DE(s) require initial conditions.

Error, (in DEtools/dfieldplot) invalid use of initial points or option - see phaseportrait

(b2) SysgDGL 1. Ordnung und DEplot, dfieldplot, fieldplot

Das Kommando DEplot, das im Abschnitt 4.1 mehrfach für gDGL 1. Ordnung genutzt wurde, kann auch für ein System von zwei gDGL 1. Ordnung eingesetzt werden.

Im Intervall der unabhängigen Variablen werden die Bereiche der beiden abhängigen Lösungsfunktionen $x(t)$ und $y(t)$ ermittelt und darin zu einer geeigneten AB die Phasenkurve $(x(t), y(t))$ gefunden. Wegen $x'(t) = y(t)$ handelt es sich hier natürlich um eine ganz spezielle Kurve.

Zum AWP (7.35) notieren wir

```
> p1:=DEplot({sys2}, [fcns], t=0..4, [[init]], linecolor=black,
             stepsize=0.01, view=[0..9, 0..6], labels=['x', 'x']):
display(p1);
```

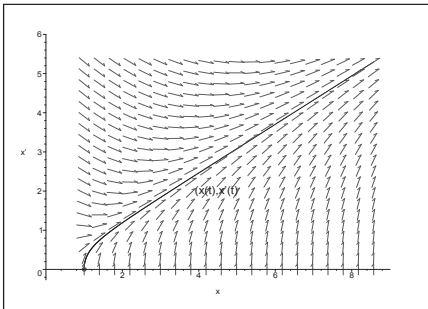


Abb. 7.56

Datei *nsteif08.ps*,

AWP (7.35),

Phasenkurve $(x(t), x'(t))$, $t \in [0, 4]$,

im Richtungsfeld (x, x')

Man kann den aufgespannten Bereich $[\min x(t), \max x(t)] \times [\min y(t), \max y(t)]$ mit einigen Funktionsauswertungen und grafischen Verläufen grob abschätzen.

```
> evalf(xx(4)), evalf(yy(4)); # 8.573522405, 5.297772819
> p:=array(1..2, []):
p[1]:=plot([xx(t), yy(t)], t=0..4, color=[black, blue]):
p[2]:=plot([xx(t), yy(t), t=0..4], color=green, labels=['x', 'x']):
display(p);
```

Ohne den DEplot-Parameter `view` sind die Display-Koordinaten die zum Verlauf der Phasenkurve $(x(t), y(t))$ gefundenen. Sonst enthält `view=[xmin..xmax, ymin..ymax]` die gewünschten Display-Koordinaten. Wählt man sie zu klein, hat man eine Ausschnittsvergrößerung, sind sie zu groß, entsteht ein "leerer Rand". In Tabellen-Plots wird die Option `view` ignoriert. Will man mehrere Phasenkurven mit verschiedenen AB in einem Richtungsfeld zeichnen, so ist zunächst die Wahl einer maximalen Ausdehnung des Display sinnvoll, bevor man sich in spezielle Bereiche "hineinzoomt".

```
> p1:=DEplot({sys2}, [fcns], t=0..4, [[x(0)=1, y(0)=0], [x(0)=1, y(0)=-1],
             [x(0)=0, y(0)=2], [x(0)=0, y(0)=-2], [x(0)=-2, y(0)=2]],
             linecolor=[black, blue, red, green, magenta], linestyle=1,
             stepsize=0.01, view=[-11..11, -7..7], labels=['x', 'x']):
p2:=pointplot([[1, 0], [1, -1], [0, 2], [0, -2], [-2, 2]], symbol=circle, color=black):
display(p1, p2);
```

Nachdem wir die Bereiche von Phasenkurven zu fünf AB ausgetestet haben, zeichnen wir nun diese im Richtungsfeld (x, x') .

(x_0, x'_0)	$[\text{xmin}, \text{xmax}] \times [\text{ymin}, \text{ymax}]$
$(1, 0)$	$[0, 5.298] \times [0, 8.574]$
$(1, -1)$	$[0.650, 3.276] \times [-1, 2.022]$
$(0, 2)$	$[0, 10.596] \times [1.300, 6.552]$
$(0, -2)$	$[-10.596, 0] \times [-6.552, -1.300]$
$(-2, 2)$	$[-6.552, -1.300] \times [-4.044, 2]$

Tab. 7.3

Bereich der Phasenkurve
mit AB $x(0) = x_0, x'(0) = x'_0$

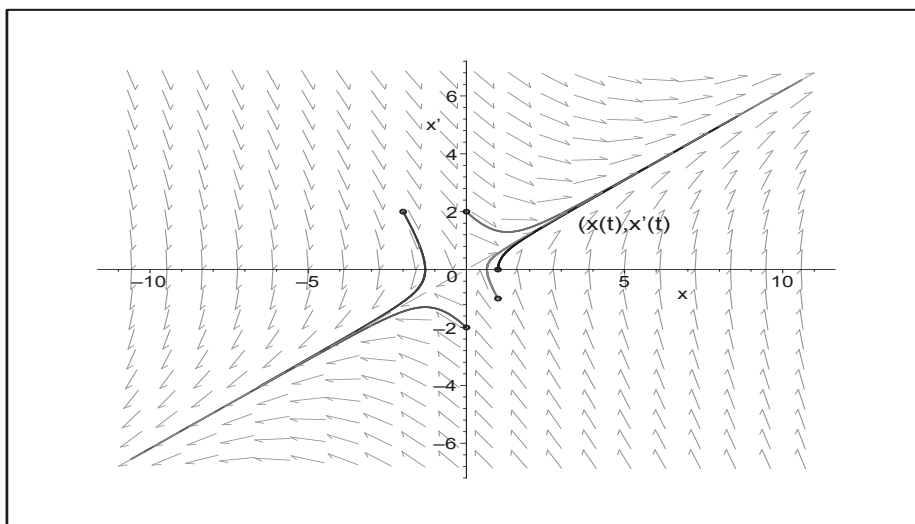


Abb. 7.57 Datei *nsteif10.ps*, AWP (7.35), Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$,
mit AB $(x(0), x'(0)) = (1, 0), (1, -1), (0, 2), (0, -2), (-2, 2)$
im Richtungsfeld (x, x') , Kommando `DEplot`

Das Kommando `dfieldplot` dient ausschließlich zur Anzeige des Richtungsfeldes (x, x') im vorgegebenen Bereich. Man kann natürlich noch ausgewählte und vorher berechnete Phasenkurven in die Abbildung einzeichnen, was nachfolgend auch getan wurde.

```
> p3:=dfieldplot([sys2],[fcns],t=0..4,x=-2..8,y=-2..8):

p1:=plot([xx(t),yy(t),t=0..4],color=black,thickness=2,labels=['x','x']):
p2a:=plot([xxa(t),yya(t),t=0..4],color=blue,thickness=2):
p2b:=plot([xxb(t),yyb(t),t=0..4],color=black,thickness=2,linestyle=3):
...
p4:=pointplot([[1,0],[1,-1],[1,1],[1,2],[1,3],[1,4],[1,5],[1,-1],
               [1,-2],[1,-1.5]],symbol=circle,color=black):
> display(p1,p2a,p2b,p2c,p2d,p2e,p2f,p2g,p2h,p2i,p3,p4);
```

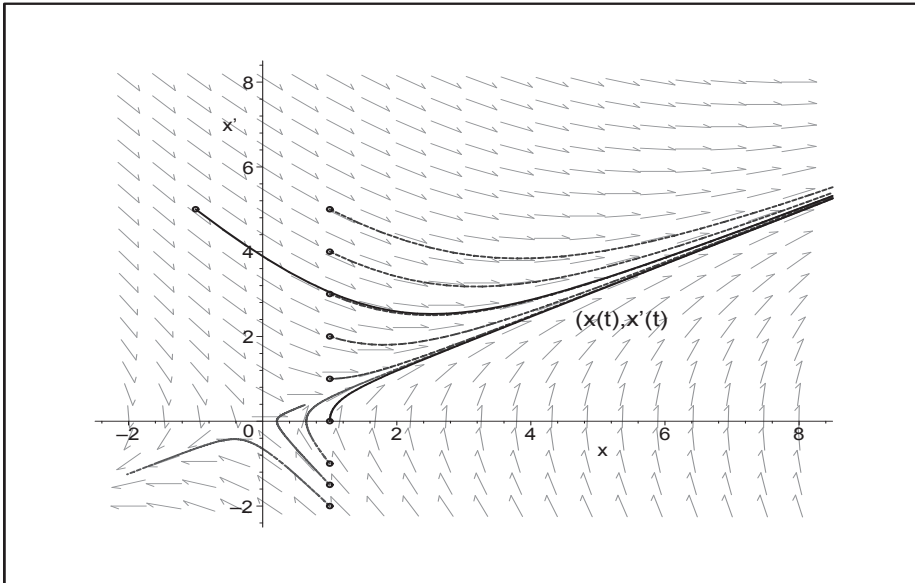


Abb. 7.58 Datei *nsteif12.ps*, AWP (7.35), Richtungsfeld $(x, x') \in [-2, 8] \times [-2, 8]$, Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$, mit AB $(1, 0)$, $(1, 1)$, $(1, 2)$, $(1, 3)$, $(1, 4)$, $(1, 5)$, $(1, -1)$, $(1, -1.5)$, $(1, -2)$, $(-1, 5)$, Kommando `dfieldplot`

Die Lösungstrajektorien $x(t)$ zu den genannten AB zeichnen wir in einer extra Grafik.

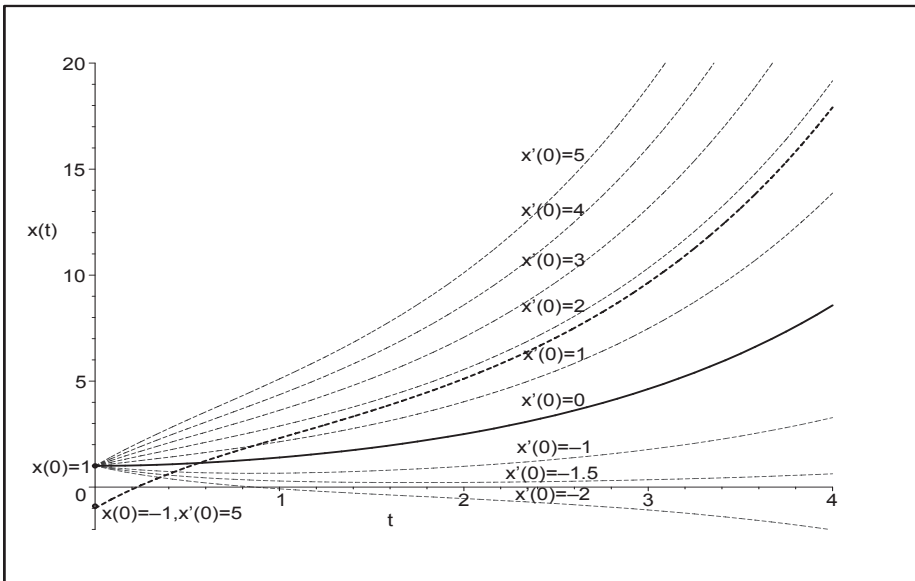


Abb. 7.59 Datei *nsteif11.ps*, AWP (7.35), Lösungskurven $x(t)$, $t \in [0, 4]$, mit AB $(1, 0)$, $(1, 1)$, $(1, 2)$, $(1, 3)$, $(1, 4)$, $(1, 5)$, $(1, -1)$, $(1, -1.5)$, $(1, -2)$, $(-1, 5)$

Das Kommando `fieldplot` ist eher nicht sinnvoll, denn welchen Anstieg als Linienelement soll man z.B. im Punkt $t = 0$ nehmen. Der Grund ist, dass von einem Punkt aus mehrere Lösungen mit entsprechenden Anstiegen ausgehen können, also die besondere Form der AB $(x(0), x'(0))$ im AWP. Lösungen mit der AB $x(0) = x_0 \neq 1$, $x'(0) = 0$ verlaufen auch nicht "parallel" zu der für $x(0) = 1$, $x'(0) = 0$. Zeichnen wir mit der Ableitung $x'(t)$ der Lösungstrajektorie $x(t)$ das Richtungsfeld, so liefert `fieldplot` als Isoklinen vertikale Linien. Seine Linienelemente passen natürlich nicht zu den drei anderen Lösungskurven in der Abb 7.60.

```
> p5:=fieldplot([1,yy(t)],t=0..4,x=-1..9,color=red):

p1:=plot(xx(t),t=0..4,color=black,thickness=2,view=[0..4,-1..9]):
p2:=plot(xxa(t),t=0..4,color=blue,thickness=2):
p3:=plot(xxb(t),t=0..4,color=magenta,thickness=2,linestyle=3):
p4:=plot(xxb1(t),t=0..4,color=maroon,thickness=2,linestyle=3):
p7:=textplot([[2.1,1.9,'x(t) mit AB (1,0)'],[0.42,2,'x(t) mit AB (1,1)'],
             [1.9,6.7,'x(t) mit AB (-1,5)'],[2.15,0.9,'x(t) mit AB (0.5,0)']]):
p8:=pointplot([[0,1],[0,-1],[0,0.5]],symbol=circle,color=black):
p9:=plot([[3.355,-0.5],[3.355,8.5]],color=black,thickness=1):
> display(p1,p2,p3,p4,p5,p7,p8,p9);
```

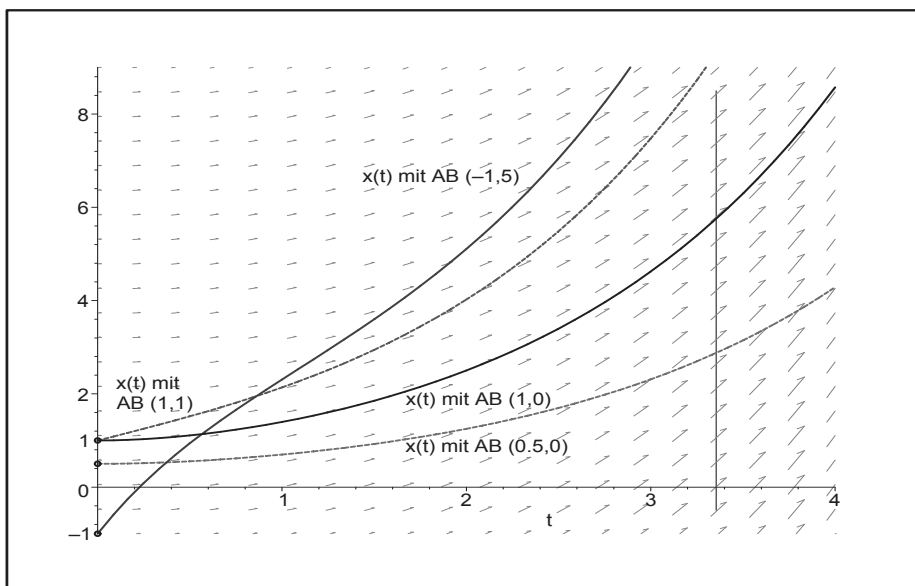


Abb. 7.60 Datei *nsteif13.ps*, AWP (7.35),
 Lösungskurven $x(t)$, $t \in [0, 4]$, mit AB $(1, 0)$, $(1, 1)$, $(0.5, 0)$, $(-1, 5)$,
 "Richtungsfeld" zur Kurve mit AB $x(0) = 1$, $x'(0) = 0$,
 Kommando `fieldplot`

(c) Gegeben sei die autonome inhomogene lineare gDGI 2. Ordnung als AWP

$$x''(t) + x'(t) - x(t) + 1 = 0, \quad x(0) = 1, \quad x'(0) = 0. \quad (7.36)$$

Die charakteristischen Wurzeln der homogenen gDGI bleiben $\lambda_{1,2} = \frac{1}{2}(-1 \pm \sqrt{5})$ wie im Teil (b). Eine spezielle Lösung der inhomogenen gDGI ist Eins, die auch die beiden AB erfüllt. Somit erhalten wir die Lösung $x(t) = 1$ und ihre Ableitung als Null.

Damit erübrigt sich die grafische Darstellung der Lösung und Ableitung, genauso der Phasenkurve, die sich auf den Punkt $(1, 0)$ reduziert.

Die allgemeine Lösung und ihre Ableitung lauten

$$\begin{aligned} x(t) &= 1 + C_1 e^{t(-1+\sqrt{5})/2} + C_2 e^{t(-1-\sqrt{5})/2}, \\ x'(t) &= \frac{-1+\sqrt{5}}{2} C_1 e^{t(-1+\sqrt{5})/2} - \frac{1+\sqrt{5}}{2} C_2 e^{t(-1-\sqrt{5})/2}. \end{aligned}$$

Es gelten

$$\lim_{t \rightarrow \infty} \frac{x'(t)}{x(t)} = \lambda_1 = \frac{1}{2}(-1 + \sqrt{5}), \quad \lim_{t \rightarrow -\infty} \frac{x'(t)}{x(t)} = \lambda_2 = \frac{1}{2}(-1 - \sqrt{5}),$$

siehe auch Geraden und Asymptotik der Phasenkurven in Abb. 7.62.

Zusammen mit noch anderen AB sollen mittels `DEplot` das Richtungsfeld (x, x') und darin liegenden Phasenkurven gezeichnet werden. Dazu testen wir die Bereiche von Phasenkurven zu fünf AB und verwenden als gemeinsamen Bereich $[-20, 3] \times [-12, 3]$.

(x_0, x'_0)	$[\text{xmin}, \quad \text{xmax}] \times [\text{ymin}, \quad \text{ymax}]$
$(1, 0)$	$[1, \quad 1] \times [0, \quad 0]$
$(1, -1)$	$[-4.298, \quad 1] \times [-3.276, \quad -0.650]$
$(0, 2)$	$[0, \quad 3.022] \times [0.422, \quad 2]$
$(0, -2)$	$[-18.169, \quad 0] \times [-11.849, \quad -1.803]$
$(-2, 2)$	$[-14.125, \quad -1.500] \times [-9.342, \quad 2]$

Tab. 7.4

Bereiche der Phasenkurven mit AB $x(0) = x_0, x'(0) = x'_0$

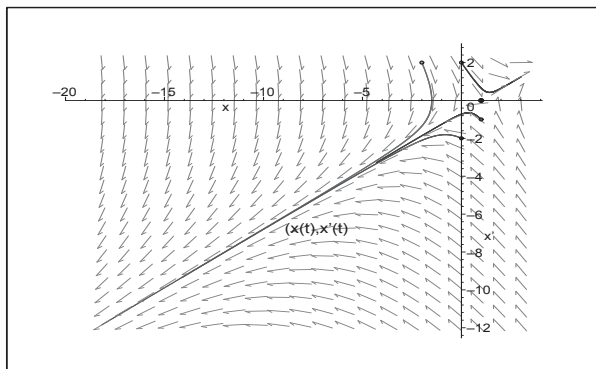


Abb. 7.61

Datei `nsteif14.ps`,
AWP (7.36),
Phasenkurven $(x(t), x'(t))$,
 $t \in [0, 4]$, mit AB
 $(x(0), x'(0)) = (1, 0), (1, -1),$
 $(0, 2), (0, -2), (-2, 2)$
im Richtungsfeld (x, x') ,
Kommando `DEplot`

Das Kommando `dfieldplot` verwenden wir zur Anzeige des Richtungsfeldes (x, x') im vorgegebenen Bereich und zeichnen mehrere vorher berechnete Phasenkurven ein.

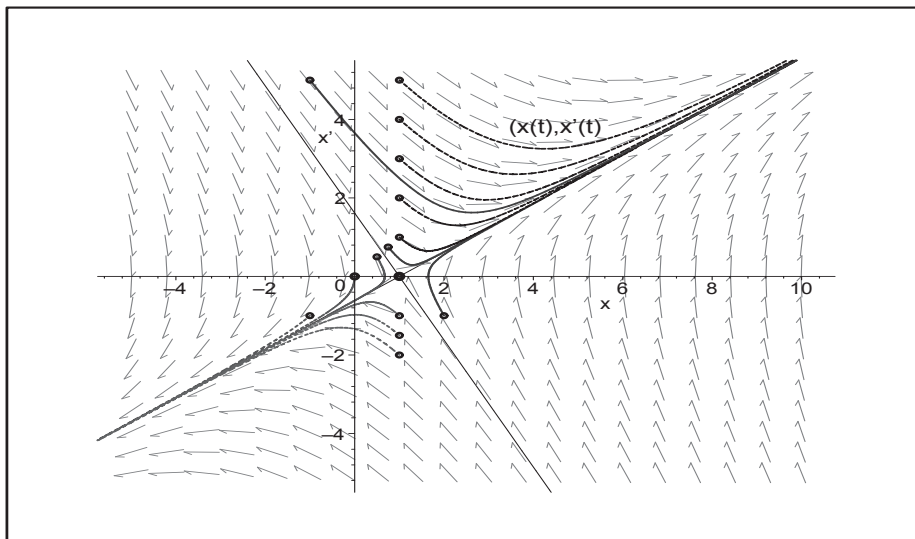


Abb. 7.62 Datei *nsteif16.ps*, AWP (7.36), Richtungsfeld $(x, x') \in [-5, 10] \times [-5, 5]$, Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$, mit AB $(1, 0)$, $(1, 1)$, $(1, 2)$, $(1, 3)$, $(1, 4)$, $(1, 5)$, $(1, -1)$, $(1, -1.5)$, $(1, -2)$, $(-1, 5)$, $(0, 0)$, $(0.5, 0.5)$, $(0.75, 0.75)$, $(2, -1)$

Die Lösungstrajektorien $x(t)$ zu den genannten AB stellen wir in einer extra Grafik dar.

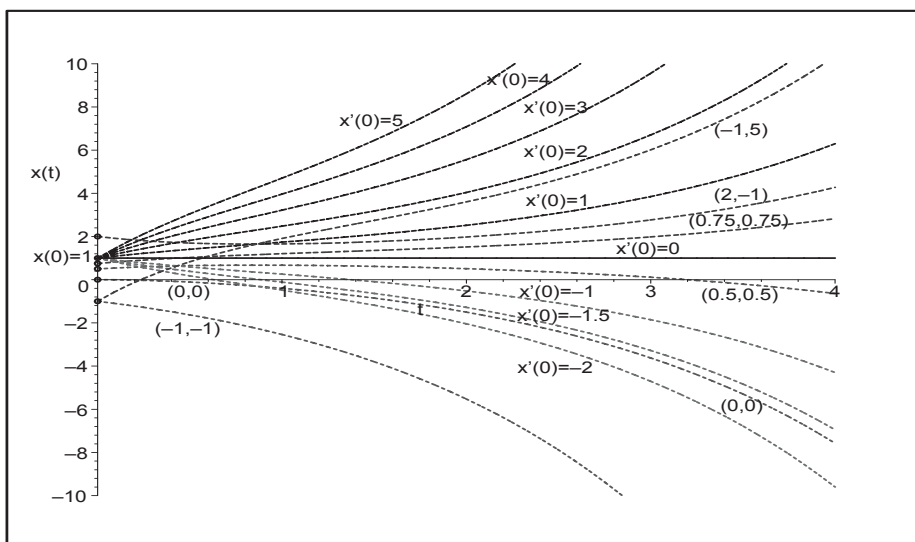


Abb. 7.63 Datei *nsteif15.ps*, AWP (7.36), Lösungskurven $x(t)$, $t \in [0, 4]$, mit AB aus vorheriger Abbildung

(d) Gegeben sei die nicht autonome inhomogene lineare gDGl 2. Ordnung als AWP

$$x''(t) + x'(t) - x(t) + t = 0, \quad x(0) = 1, \quad x'(0) = 0. \quad (7.37)$$

Die charakteristischen Wurzeln der homogenen gDGl bleiben $\lambda_{1,2} = \frac{1}{2}(-1 \pm \sqrt{5})$ wie im Teil (b). Eine spezielle Lösung der inhomogenen gDGl ist $t + 1$, sodass wir die Lösung und ihre Ableitung erhalten.

$$\begin{aligned} x(t) &= 1 + t + \frac{\sqrt{5}}{5} (-e^{t(-1+\sqrt{5})/2} + e^{t(-1-\sqrt{5})/2}), \\ &= 1 + t + 0.447\,213\,595 (-e^{0.618\,033\,988t} + e^{-1.618\,033\,988t}), \\ x'(t) &= 1 - \frac{\sqrt{5}}{10}(-1 + \sqrt{5}) e^{t(-1+\sqrt{5})/2} - \frac{\sqrt{5}}{10}(1 + \sqrt{5}) e^{t(-1-\sqrt{5})/2} \\ &= 1 - 0.276\,393\,201 e^{0.618\,033\,988t} - 0.723\,606\,797 e^{-1.618\,033\,988t}. \end{aligned}$$

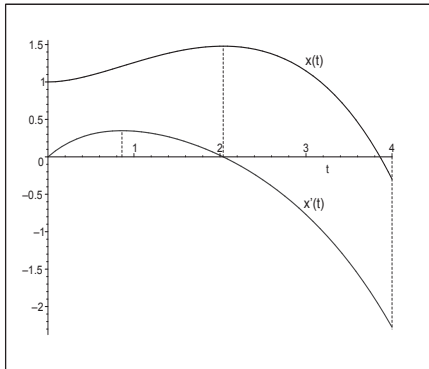


Abb. 7.64

Datei *nsteif17.ps*,

AWP $x''(t) + x'(t) - x(t) + t = 0$,

$x(0) = 1, x'(0) = 0$, sowie

Lösung $x(t)$ und Ableitung $x'(t)$ in $[0, 4]$

Es gilt gerundet $x'(0.861) = 0.350$, $x(2.037) = 1.479$, $x'(2.037) = 0$ sowie $x(4) = -0.298$, $x'(4) = -2.276$, sodass man bei $t \in [0, 4]$ für das Phasenportrait $(x(t), x'(t))$ den Bereich $[-0.3, 1.5] \times [-2.3, 0.4]$ bzw. etwas größer $[-0.5, 1.5] \times [-2.5, 0.5]$ nehmen kann.

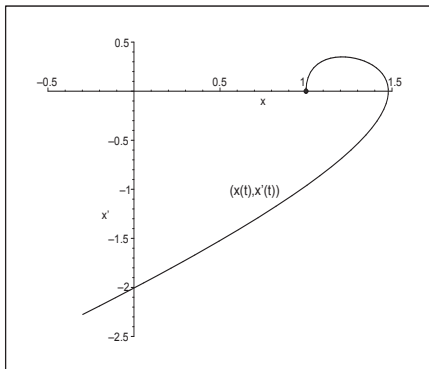


Abb. 7.65

Datei *nsteif18.ps*,

Phasenportrait $(x(t), x'(t))$

im Bereich $[-0.5, 1.5] \times [-2.5, 0.5]$

Man notiert die gDGl bei $x' = y$ und $\mathbf{x} = (x, x')^T$ auch als System von zwei nicht autonomen linearen gDGl 1. Ordnung gemäß

$$\begin{aligned} x'(t) &= y(t), & x(0) &= 1, \\ y'(t) &= x(t) - y(t) - t, & y(0) &= 0, \\ \mathbf{x}'(t) &= A\mathbf{x}(t) + \mathbf{g}(t), & A &= \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}, \mathbf{g}(t) = (0, -t)^T, \mathbf{x}(0) = (1, 0)^T. \end{aligned} \quad (7.38)$$

Natürlich kommt man dabei zur gleichen Lösung $x(t)$ mit ihrer Ableitung $x'(t) = y(t)$. Verschiedene Lösungsdarstellungen erhält man mit entsprechenden Optionen im Maple-Kommando `dsolve`.

```
> sys4:=diff(x(t),t)=y(t),diff(y(t),t)=x(t)-y(t)-t:
fcons:=x(t),y(t):
init:=x(0)=1,y(0)=0:
loe4:=dsolve({sys4,init},{fcons});

> # Loesungskomponenten als Funktionen
loe4:=dsolve({sys4,init},{fcons}); # Reihenfolge x,y beachten
xx:=unapply(rhs(op(1,loe4)),t): # unapply(rhs(op(2,loe4)),t):
yy:=unapply(rhs(op(2,loe4)),t): # unapply(rhs(op(1,loe4)),t):
```

$$\begin{aligned} loe4 := & \left\{ \begin{aligned} x(t) &= -\frac{1}{5}e^{\left(\frac{(\sqrt{5}-1)t}{2}\right)}\sqrt{5} + \frac{1}{5}e^{\left(-\frac{(\sqrt{5}+1)t}{2}\right)}\sqrt{5} + 1 + t, \\ y(t) &= -\frac{1}{5}\left(\frac{\sqrt{5}}{5} - \frac{1}{2}\right)e^{\left(\frac{(\sqrt{5}-1)t}{2}\right)}\sqrt{5} + \frac{1}{5}\left(-\frac{\sqrt{5}}{5} - \frac{1}{2}\right)e^{\left(-\frac{(\sqrt{5}+1)t}{2}\right)}\sqrt{5} + 1 \end{aligned} \right\} \end{aligned}$$

```
> dsolve({sys4,x(0)=1,y(0)=0},{fcons},method=laplace);
l4:=[collect(simplify(rhs(op(1,%))),exp),
collect(simplify(rhs(op(2,%))),exp)];
evalf(l4[1]);
evalf(l4[2]);
```

$$\begin{aligned} & \left\{ \begin{aligned} x(t) &= 1 + t - \frac{2}{5}e^{\left(-\frac{t}{2}\right)}\sqrt{5} \sinh\left(\frac{t\sqrt{5}}{2}\right), \\ y(t) &= 1 + \frac{1}{5}\left(-5 \cosh\left(\frac{t\sqrt{5}}{2}\right) + \sqrt{5} \sinh\left(\frac{t\sqrt{5}}{2}\right)\right)e^{\left(-\frac{t}{2}\right)} \end{aligned} \right\} \\ l4 := & \left[1 + t - \frac{2}{5}e^{\left(-\frac{t}{2}\right)}\sqrt{5} \sinh\left(\frac{t\sqrt{5}}{2}\right), 1 + \left(-\cosh\left(\frac{t\sqrt{5}}{2}\right) + \frac{1}{5}\sqrt{5} \sinh\left(\frac{t\sqrt{5}}{2}\right)\right)e^{\left(-\frac{t}{2}\right)} \right] \\ & 1. + t - 0.8944271908e^{(-0.5000000000t)} \sinh(1.118033988t) \\ & 1. + (-1. \cosh(1.118033988t) + 0.4472135954 \sinh(1.118033988t))e^{(-0.5000000000t)} \end{aligned}$$

```
> dsolve({sys4,x(0)=1,y(0)=0},fcns,type=series);
```

$$\{x(t) = t - t^2 + \frac{1}{2}t^3 - \frac{2}{24}t^4 + \frac{1}{15}t^5 + O(t^6), y(t) = 1 + \frac{1}{2}t^2 - \frac{1}{3}t^3 + \frac{1}{8}t^4 - \frac{1}{24}t^5 + O(t^6)\}$$

Im Gegensatz zum autonomen AWP (7.35) erhält man bei AWP (7.38) mit

```
> p1:=DEplot({sys4}, [fcns], t=0..4, [[init]], linecolor=black,
             stepsize=0.01, view=[-0.5..1.5, -2.5..0.5], labels=['x', 'x']):
display(p1);
```

nur den Plot der Phasenkurve $(x(t), x'(t))$, $t \in [0, 4]$. Nur bei autonomen SysgDgl wird auch das Richtungsfeld gezeigt.

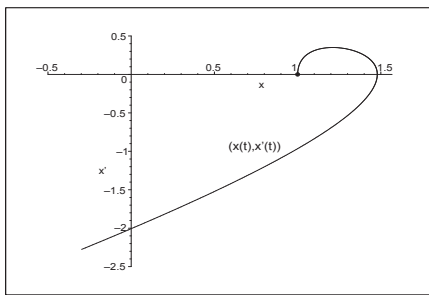


Abb. 7.66

Datei *nsteif20.ps*,

AWP (7.38),

Phasenkurve $(x(t), x'(t))$, $t \in [0, 4]$,

ohne Richtungsfeld (x, x')

Das gleiche gilt natürlich, wenn man mehrere Phasenkurven mittels `DEplot` darstellt.

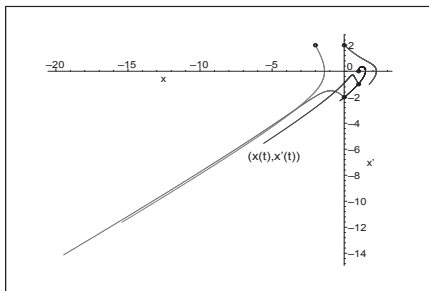


Abb. 7.67

Datei *nsteif21.ps*,

AWP (7.38),

Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$,

mit AB $(1, 0)$, $(1, -1)$, $(0, 2)$, $(0, -2)$, $(-2, 2)$,

ohne Richtungsfeld (x, x')

Auch mit `dfieldplot` ist die Darstellung des Richtungsfeldes nicht möglich.

Also nutzt man den üblichen Plot von Phasenkurven und Lösungstrajektorien.

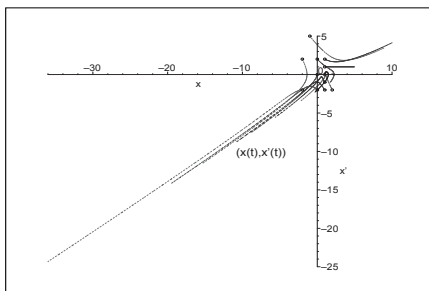


Abb. 7.68

Datei *nsteif23.ps*,

AWP (7.38),

Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$,

mit den AB $(1, 0)$, $(1, 1)$, $(1, 2)$, $(1, -1)$, $(1, -2)$,

$(0, 0)$, $(0, 2)$, $(0, -2)$,

$(-2, 2)$, $(2, -2)$, $(-2, -2)$, $(-1, 5)$

im Bereich $[-36, 10] \times [-25, 5]$

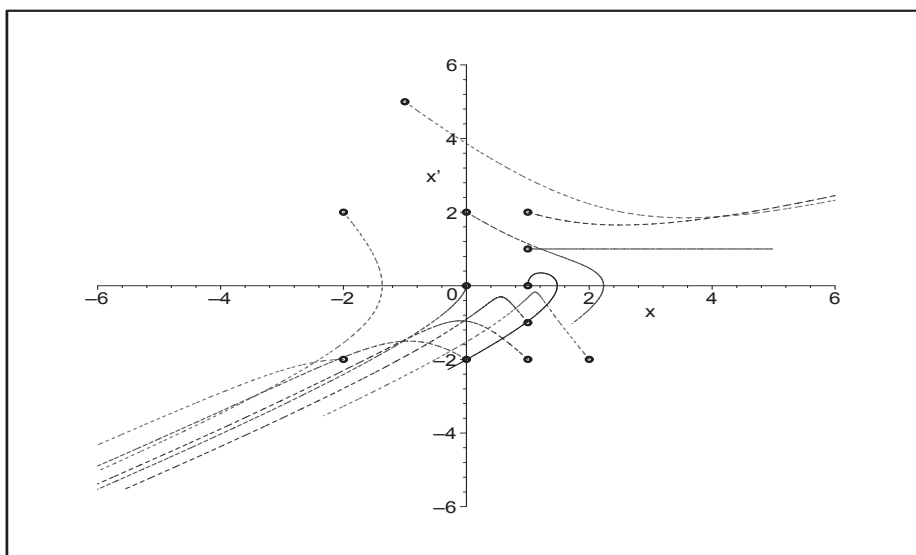


Abb. 7.69 Datei *nsteif24.ps*, AWP (7.38), Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$, zu den genannten AB im Bereichsausschnitt $[-6, 6] \times [-6, 6]$

Man sieht, dass natürlich kein Richtungsfeld (x, x') eingezeichnet werden kann. Zudem bemerkt man die relativ kurzen Phasenkurven für die AB $(1, 0)$, $(0, 2)$ sowie $(1, 1)$ mit der Lösung $x(t) = 1 + t$, während zur AB $(-2, -2)$ der Endpunkt $(x(4), x'(4)) = (-36.614, -24.721)$ erreicht wird. Inhomogenitäten, wodurch die gDGL zugleich nicht autonom wird, verhindern nicht nur die Grafik eines Richtungsfeldes, sondern haben auch starken Einfluss auf den Phasenverlauf. Ungeachtet dessen verlaufen die Lösungstrajektorien $x(t)$ zu den verschiedenen AB moderat.

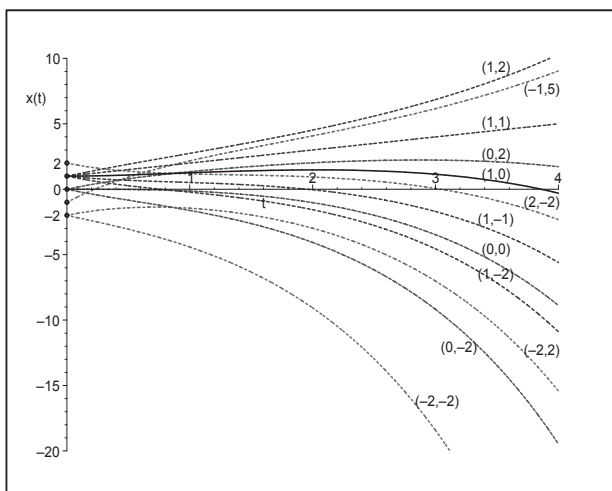


Abb. 7.70
Datei *nsteif22.ps*,
AWP (7.38),
Lösungstrajektorien
 $x(t)$, $t \in [0, 4]$,
zu den genannten AB

(e) Gegeben sei die autonome inhomogenen nichtlineare gDGL 2. Ordnung als Randwertproblem (RWP) mit den Randbedingungen (RB)

$$x''(t) - x'(t)x(t) - 1 = 0, \quad x(0) = 0, \quad x(1) = 0, \quad t \in [0, 1]. \quad (7.39)$$

Ziel ist es zunächst, dafür ein entsprechendes AWP zu formulieren und dieses als Grundlage für die Lösung des RWP zu nehmen.

Das abgeleitete AWP sei

$$x''(t) - x'(t)x(t) - 1 = 0, \quad x(0) = 0, \quad x'(0) = s, \quad t \geq 0, \quad (7.40)$$

wobei der reelle Parameter s noch geeignet zu wählen ist.

Wegen der Nichtlinearität ist weder das RWP noch das AWP (auch bei festem s) explizit oder analytisch lösbar. So liefert `dsolve` in den folgenden Kommandos kein Ergebnis.

```
> de5:=diff(x(t),t$2)-diff(x(t),t)*x(t)-1=0;
  init:=x(0)=0,D(x)(0)=s;
> dsolve({de5,init},x(t));

> init1:=x(0)=0,D(x)(0)=-0.5;          # s=-0.5
> dsolve({de5,init1},x(t));

> dsolve({de5,init1},x(t),method=laplace); # nur bei linearen DGL

> # als RWP
  bc:=x(0)=0,x(1)=0;
> dsolve({de5,bc},x(t));
```

Lediglich mit

```
> dsolve(de5,x(t),explicit);
  dsolve({de5,init1},x(t),implicit);
```

werden Darstellungen mit `RootOf` und hypergeometrische Funktionen erzeugt, die aber nicht ohne Weiteres zu Auswertungen verwendet werden können.

Da der Verdacht vorliegt, dass man mit dem Parameterwert $s = -0.5$ in den AB des AWP in die Nähe der Lösung des RWP kommt, testen wir den Lösungsansatz mittels Potenzreihe. Von der Reihe nehmen wir geschickterweise nur die ersten 5 Glieder, also

$$x_s(t) = -\frac{1}{2}t + \frac{1}{2}t^2 + \frac{1}{24}t^3 - \frac{1}{16}t^4 + \frac{1}{48}t^5, \quad (7.41)$$

und stellen fest, dass die Funktion $x_s(t)$ sowohl den AB als auch den RB genügt sowie die gDGL bis auf Glieder der Ordnung t^p , $p \geq 4$, erfüllt.

```
> Order:=10:      # fuer Anzahl der Reihenglieder
  dsolve({de5,init1},x(t),series);
```

$$x(t) = -\frac{1}{2}t + \frac{1}{2}t^2 + \frac{1}{24}t^3 - \frac{1}{16}t^4 + \frac{1}{48}t^5 + \frac{5}{576}t^6 - \frac{47}{8064}t^7 + \frac{1}{2304}t^8 + \frac{325}{290304}t^9 + O(t^{10})$$


```

> xxt:=unapply(-1/2*t+1/2*t^2+1/24*t^3-1/16*t^4+1/48*t^5,t):
xxt1:=D(xxt):
xxt2:=D(xxt1):

> xxt(0), xxt1(0);
xxt(1), xxt1(1);
simplify(eval(subs(x(t)=xxt(t),de5)));

```

$$0, \frac{-1}{2}$$

$$0, \frac{23}{48}$$

$$\frac{25}{96}t^4 + \frac{47}{192}t^5 - \frac{7}{128}t^6 - \frac{13}{576}t^7 + \frac{3}{256}t^8 - \frac{5}{2304}t^9 = 0$$

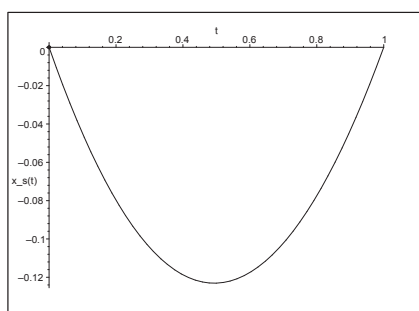


Abb. 7.71

Datei *nsteif25.ps*,
 AWP (7.40), AB mit $s = -0.5$,
 Näherungslösung (7.41)
 $x_s(t)$, $t \in [0, 1]$,
 mit Potenzreihenansatz

Zu Testzwecken soll das RWP mit guter Genauigkeit numerisch gelöst werden.

In Maple findet man dazu entsprechende Hinweise, auf die hier nicht weiter eingegangen wird.

```

> Digits:=20:
bsol:=dsolve({de5,bc},numeric,method=bvp,range=0..1,abserr=1e-16);
bsol(0);
bsol(0.5);
bsol(1.0);
> Digits:=10:

```

$bsol := \mathbf{proc}(x.bvp) \dots \mathbf{end proc}$

$$\left[t = 0., x(t) = 0., \frac{d}{dt}x(t) = -0.50416592791003595965 \right]$$

$$\left[t = 0.5, x(t) = -0.12498305218921320693, \frac{d}{dt}x(t) = 0.0036444537572298360201 \right]$$

$$\left[t = 1.0, x(t) = 0., \frac{d}{dt}x(t) = 0.49583407208996404024 \right]$$

Somit werden wir bei der Lösung des AWP auch mit dem recht genauen Parameterwert $s = -0.504166$ für die Anfangsableitung $x'(0)$ operieren.

Wir notieren das AWP (7.40) ebenfalls als System von zwei autonomen inhomogenen nichtlinearen gDGl 1. Ordnung gemäß

$$\begin{aligned}x'(t) &= y(t), & x(0) &= 0, \\y'(t) &= x(t)y(t) + 1, & y(0) &= s, \\ \mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)) = \mathbf{f}(\mathbf{x}(t)), & \mathbf{x}(0) &= (0, s)^T.\end{aligned}\tag{7.42}$$

```
> sys5:=diff(x(t),t)=y(t),diff(y(t),t)=x(t)*y(t)+1;
  fcns:=x(t),y(t);
  init:=x(0)=0,y(0)=s;
> init1:=x(0)=0,y(0)=-0.5;
```

Auch in der transformierten Form ist die analytische Lösung nicht möglich.

Damit bleibt nur die numerische Lösung im Zusammenhang mit den Kommandos `dsolve`, `phaseportrait`, `DEplot` und `dfieldplot`.

(e1) Der Maple-Befehl `dsolve` enthält eine Reihe von DGI-Lösern, mehr als bei `DEplot` und `phaseportrait`, sowie eine sinnvolle Ergebnisdarstellung bzw. -auswertung.

Details sind der Online-Hilfe zu entnehmen.

So verwendet man die Methoden `rkf45` (ist Voreinstellung), `classical[rk4,...]`, `gear`, `dverk78`, `rosenbrock`, `rosenbrock_dae`, `lsode`, `taylorseries`, `rkf45_dae`, `mebdfi`, `bvp`.

```
> loe51:=dsolve({de5,init1},x(t),type=numeric, # bzw. einfach numeric
  method=gear, # weitere Optionen moeglich
  output=array([seq(0.1*i,i=0..10)])); # Stellen der Auswertung:
  # hier relativ grob
```

$$loe51 := \begin{bmatrix} \left[t, x(t), \frac{d}{dt}x(t) \right] \\ \begin{bmatrix} 0. & 0. & -.5 \\ .1 & -.0449643668968620764 & -.398989102854782064 \\ .2 & -.0797595174060793112 & -.296819209691574270 \\ .3 & -.104325523530942959 & -.194558092569997010 \\ .4 & -.118693460210456888 & -.0929559312516558128 \\ .5 & -.122953307477378551 & .0075587579093452728 \\ .6 & -.117222138246121432 & .106870514847029362 \\ .7 & -.101613507979648396 & .205162652501499842 \\ .8 & -.0762082805158981053 & .302903851009129699 \\ .9 & -.0410265004321861623 & .400841586868400102 \\ 1.0 & .00400065301760353913 & .500008002611822877 \end{bmatrix} \end{bmatrix}$$

Analog ist der Aufruf einer anderen Methode mit geringen Unterschieden im Ergebnis.

```
> loe52:=dsolve({de5,init1},x(t),numeric,
  method=rkf45,
  output=array([seq(0.1*i,i=0..10)]));
```

$$\text{loe52} := \begin{bmatrix} \begin{bmatrix} t, x(t), \frac{d}{dt}x(t) \end{bmatrix} \\ \begin{bmatrix} 0. & 0. & -.5000000000000000 \\ .10000000000000004 & -.0449646130717692994 & -.398988608633262132 \\ .20000000000000010 & -.0797596151028408668 & -.296819172820215360 \\ .29999999999999998 & -.104325521843663698 & -.194558105717530100 \\ .40000000000000022 & -.118693411953866234 & -.0929559442475329096 \\ .50000000000000000 & -.122953350696438396 & .00755868737673635882 \\ .59999999999999976 & -.117222067729801482 & .106870495297289452 \\ .69999999999999954 & -.101613630914293943 & .205162623366426434 \\ .80000000000000042 & -.0762082207520253935 & .302903839647251372 \\ .90000000000000022 & -.0410266681722313842 & .400841797801013922 \\ 1. & .00400071376438856843 & .500008002309892952 \end{bmatrix} \end{bmatrix}$$

Der Vorteil eines solchen Ergebnisses ist, dass man es in verschiedene Teile zerlegen kann. So entnimmt man auch die Punktfolgen (Liste von Listen) für die drei möglichen Phasenkurven $(t, x(t))$, $(t, x'(t))$ sowie $(x(t), x'(t))$, $(x'(t), x(t))$.

```

> ko:=eval(loe51[1,1]); # Kopfteil
  lo:=eval(loe51[2,1]); # Wertetabelle
  nr:=rowdim(lo); # 11
> lo12:=delcols(lo,3..3); # 3. Spalte loeschen -> (t,x(t))
  lo13:=delcols(lo,2..2); # 2. Spalte loeschen -> (t,x'(t))
  lo23:=delcols(lo,1..1); # 1. Spalte loeschen -> (x(t),x'(t)),(x'(t),x(t))

> # Kurve (t,x(t)), analog die anderen
  lis1:=[seq([lo12[i,1],lo12[i,2]],i=1..nr)];
  p1:=plot(lis1,color=blue);
  p2:=pointplot([0,0],symbol=circle,color=black);
  display(p1,p2);

```

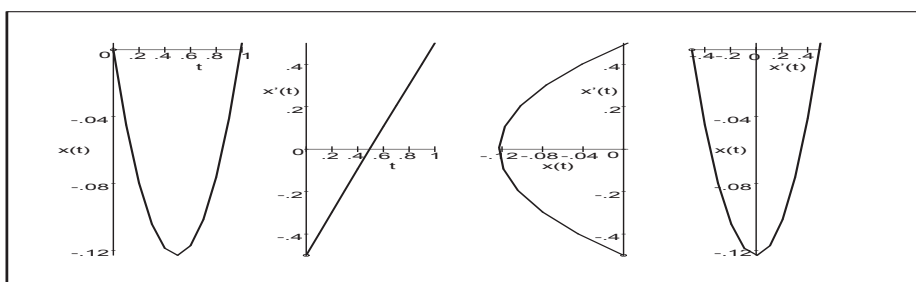


Abb. 7.72 Datei *nsteif26.ps*, AWP (7.40), AB mit $s = -0.5$,

Näherungslösung $x(t)$, $t \in [0, 1]$, als Polygonzug sowie $(t, x'(t))$, $(x(t), x'(t))$, $(x'(t), x(t))$ bei 11 äquidistanten Auswertungsstellen $t_i = \frac{i}{10}$, $i = 0(1)10$, mittels `dsolve({de5, ..., numeric, method=gear, ...})`

Da sich die Ableitung $x'(t)$ in $[0, 1]$ fast wie eine lineare Funktion verhält, sind die Kurven $(t, x(t))$ und $(x'(t), x(t))$ sehr ähnlich.

Eine andere Ergebnisform ist eine Funktion der Näherungslösung, welche an gewünschten Stellen ausgewertet und damit auch für grafische Darstellungen verwendet werden kann.

```
> loes53:=dsolve({de5,init1},numeric,range=0..1);
loes53(0); loes53(0.5); loes53(1);
seq(rhs(loes53(0.1*i)[2]),i=0..10);
```

$$\text{loes53} := \text{proc}(x_rkf45) \dots \text{end proc}$$

$$\left[t = 0., x(t) = 0., \frac{d}{dt}x(t) = -.500000000000000 \right]$$

$$\left[t = .5, x(t) = -.122953350696438396, \frac{d}{dt}x(t) = .00755868737673635622 \right]$$

$$\left[t = 1., x(t) = .00400073019471768043, \frac{d}{dt}x(t) = .500008005791624743 \right]$$

```
0., -.0449646130717692576, -.0797596151028408806, -.104325521843663712,
-.118693411953866178, -.122953350696438396, -.117222067729801454,
-.101613630914293915, -.0762082207520253380, -.0410266252286706490,
.00400073019471768043
```

(e2) Mit dem Maple-Kommando `phaseportrait` bestimmen wir zu (7.40) die Lösungstrajektorie als Phasenkurve $(t, x(t))$ bei numerischer Lösung der gDGL.

Die Lösungsmethoden sind `classical`, `dverk78`, `gear`, `lsode`, `rkf45`.

Ohne Angabe der Methode ist `method=rkf45` als Standard eingestellt.

Beim Verfahren `classical` hat man noch die Auswahl zwischen `foreuler` (PZV), `heunform` (HV, VPZV), `impoly` (MPZV), `rk2` (=heunform), `rk3`, `rk4` (KRKV), `adambash`, `abmoulton`, alles ESV und MSV, die schon behandelt wurden.

Die Angabe `method=classical` bedeutet soviel wie `method=classical[foreuler]`.

Aus der Vielfalt der Optionen soll nur ein Hinweis auf den Parameter `stepsize` als gewählte Schrittweite erfolgen. Standard ist beim Integrationsintervall $[t_0, T]$ die Schrittweite $h = (T - t_0)/20$, die auch bei zu großem vorgegebenen h verwendet wird.

Natürlich hat das PZV unter den genannten ESV und MSV die kleinste Genauigkeitsordnung $\mathcal{O}(h)$, was im Vergleich mit dem KRKV sehr deutlich wird.

```
> p11:=phaseportrait(de5,x(t),t=0..1,[[x(0)=0,D(x)(0)=-0.5]],
stepsize=0.05,linecolor=black,method=classical[rk4],
view=[0..1,-0.14..0.004]): # scene=[t,x(t)]
p12:=phaseportrait(de5,x(t),t=0..1,[[x(0)=0,D(x)(0)=-0.5]],
stepsize=0.05,linecolor=blue,method=classical[foreuler]):
p13:=textplot([[0.63,-0.08,'mit KRKV'],[0.77,-0.13,'mit PZV']]):
p14:=pointplot([0,0],symbol=circle,color=black):
display(p11,p12,p13,p14);

> # analog mit AB [[x(0)=0,D(x)(0)=-0.504166]]
```

Mit der AB der Ableitung $x'(0) = -0.5$ liefern sowohl das PZV als auch das KRKV eine Näherungslösung mit $x(1) \neq 0$.

Mit der AB der Ableitung $x'(0) = -0.504166$ kommen wir mit dem KRKV aber schon sehr nahe an die RB $x(1) = 0$ des RWP (7.39).

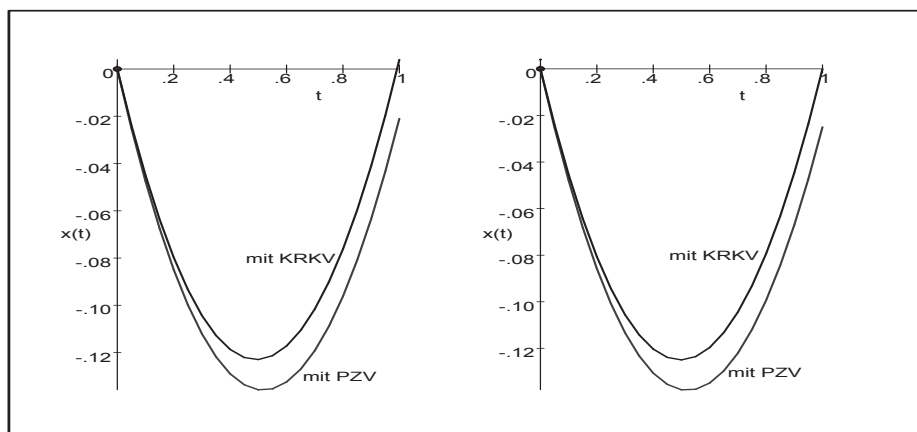


Abb. 7.73 Datei *nsteif27.ps*, AWP (7.40), AB mit $s = -0.5$ (linke Grafik) sowie $s = -0.504166$, Näherungslösungen $x(t)$, $t \in [0, 1]$, mittels `phaseportrait(de5, ..., method=classical[rk4], ...)` bzw. `phaseportrait(de5, ..., method=classical[foreuler], ...)`

(e3) Das Maple-Kommando `DEplot` hat analoge Parameter zu `phaseportrait`. Zur gDGI (7.40) können wir damit jedoch nur Phasenkurven $(t, x(t))$ näherungsweise bestimmen. Das Richtungsfeld wird dabei nicht dargestellt.

Ohne Angabe der Methode bedeutet das die Anwendung von `method=rkf45`.

```
> pp:=array(1..2, []): # Array-Plot
> p1:=DEplot({de5}, [x(t)], t=0..1, [[x(0)=0,D(x)(0)=-0.5]],
             stepsize=0.05, linecolor=black):
display(p1);
pp[1]:=p1:

> p21:=DEplot({de5}, [x(t)], t=0..1,
              [[x(0)=0,D(x)(0)=-0.5], [x(0)=0,D(x)(0)=-0.4],
               [x(0)=0,D(x)(0)=-0.6], [x(0)=0,D(x)(0)=-0.504166]],
              stepsize=0.05, linecolor=[blue,green,red,black], thickness=2):
p22:=textplot([[0.72,0.06,'AB (0,-0.4)'], [0.62,-0.08,'(0,-0.5)'],
               [0.6,-0.155,'(0,-0.6)'], [1,-0.08,'(0,-0.504166)']]):
display(p21,p22);
pp[2]:=display(p21,p22):

> display(pp);
```

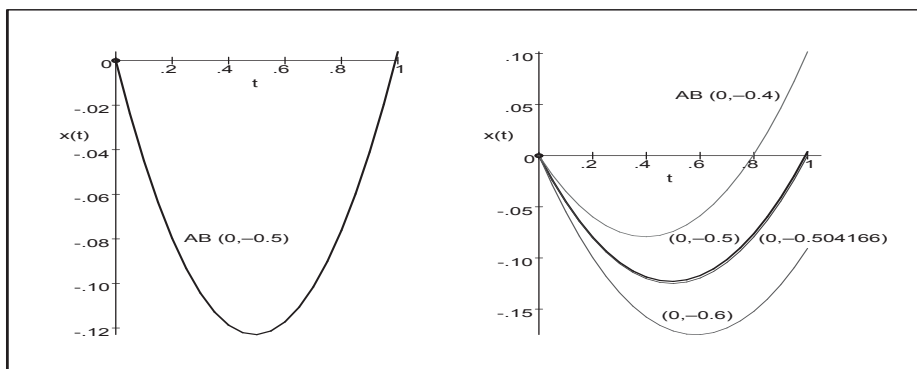


Abb. 7.74 Datei *nsteif28.ps*, AWP (7.40), AB mit $s = -0.5$ (linke Grafik) sowie $s = -0.6, -0.504166, -0.5, -0.4$, Näherungslösungen $x(t)$, $t \in [0, 1]$, mittels `DEplot({de5}, ...)` bei `method=rkf45`

(e4) Für das autonome SysgDGL (7.42) können wir mit den Kommandos `DEplot`, `dfieldplot`, `phaseportrait` das Richtungsfeld zeigen, evtl. auch Näherungslösungen.

```
> p11:=DEplot({sys5}, [fcns], t=0..1, [[init1]], stepsize=0.01,
  linecolor=black, labels=['x', 'x'],
  view=[-0.17..0.1, -0.6..0.6]): # Maple V : Rand um Gebiet (x,x')
p12:=pointplot([0, -0.5], symbol=circle):
display(p11, p12);

> p21:=DEplot({sys5}, [fcns], t=0..1, [[init1], [x(0)=0, y(0)=-0.4],
  [x(0)=0, y(0)=-0.6], [x(0)=0, y(0)=-0.504166]], stepsize=0.01,
  linecolor=[blue, green, red, black], labels=['x', 'x'], thickness=2):
p22:=pointplot([[0, -0.6], [0, -0.5], [0, -0.4]], symbol=circle):
display(p21, p22);
```

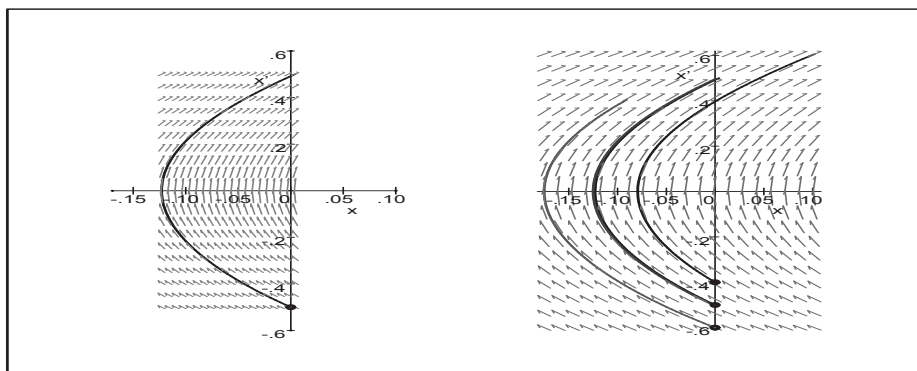


Abb. 7.75 Datei *nsteif31.ps*, AWP (7.42), AB mit $s = -0.5$ (linke Grafik) sowie $s = -0.6, -0.504166, -0.5, -0.4$, Näherungslösungen $x(t)$, $t \in [0, 1]$, und Richtungsfeld mittels `DEplot({sys5}, ...)`

Der Befehl `dfieldplot` dient zur alleinigen Darstellung des Richtungsfeldes mit x, x' . Natürlich kann man im Nachhinein noch Phasenkurven $(x(t), x'(t))$ bzw. $(x'(t), x(t))$ einzeichnen.

```
> dfieldplot([sys5],[fcns],t=0..1,x=-.2..0.1,y=-.6..0.6,labels=['x','x']):
dfieldplot([sys5],[y(t),x(t)],t=0..1,x=-.2..0.1,y=-.6..0.6,
labels=['x','x']):
```

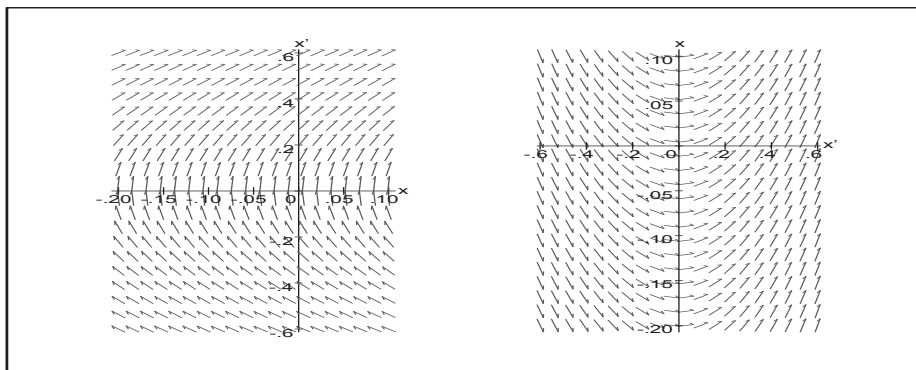


Abb. 7.76 Datei *nsteif32.ps*, AWP (7.42), AB mit $s = -0.5$,
Richtungsfeld mit `dfieldplot([sys5],...)` als (x, x') und (x', x)

Verwenden wir den Befehl `phaseportrait` für das autonome SysGDGL (7.42), so wird der Plot in Abhängigkeit vom gewünschten Szenarium `scene` erstellt.

```
> # Standard ist Phasenkurve (x(t),y(t)) mit Richtungsfeld
> p23:=phaseportrait([sys5],[fcns],t=0..1,[[init1]],stepsize=0.01,
# scene=[x(t),y(t)],
linecolor=black,method=classical[rk4],labels=['x','x']):
> p32:=phaseportrait([sys5],[fcns],t=0..1,[[init1]],stepsize=0.01,
scene=[y(t),x(t)],
linecolor=black,method=classical[rk4],labels=['x','x']):
> # Phasenkurve (t,x(t)) bzw. (t,y(t)) ohne Richtungsfeld
> p12:=phaseportrait([sys5],[fcns],t=0..1,[[init1]],stepsize=0.01,
scene=[t,x(t)],
linecolor=black,method=classical[rk4]):
> p13:=phaseportrait([sys5],[fcns],t=0..1,[[init1]],stepsize=0.01,
scene=[t,y(t)],
linecolor=black,method=classical[rk4]):

> pp:=array(1..3,[]):
pp[1]:=display(p23):
pt1:=textplot([[0.8,-0.15,'x(t)'],[0.8,0.2,'x'(t)]]):
pp[2]:=display(p12,p13,pt1,labels=['t','']):
pp[3]:=display(p32):
> display(pp);
```

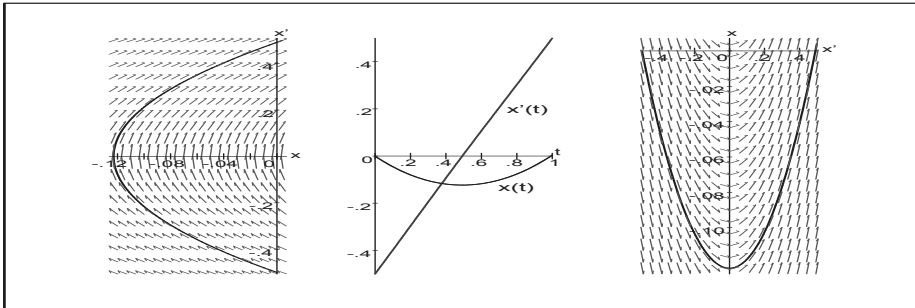


Abb. 7.77 Datei *nsteif33.ps*, AWP (7.42), AB mit $s = -0.5$,
`phaseportrait([sys5], ...)` als (x, x') , $\{(t, x), (t, x')\}$ und (x', x)

Zum Schluss folgt ein Vergleich der klassischen Lösungsmethoden mit den beiden AB
 $x(0) = 0$, $y(0) = x'(0) = -0.5$ und $x(0) = 0$, $y(0) = x'(0) = -0.504166$.

Die Näherungslösung mit der zweiten AB liegt stets knapp unter der ersten.

```
> pp:=array(1..2,1..4, []):
  init1:=x(0)=0,y(0)=-0.5; init2:=x(0)=0,y(0)=-0.504166;
  opt1:=stepsize=0.05,scene=[t,x(t)]:
> pp11:=phaseportrait([sys5],[fcs],t=0..1,[[init1],[init2]],opt1,
  linecolor=[black,gray],method=classical[foreuler]):
pp12:=plot([0.01,-0.14],t=0..1,color=white):
pp[1,1]:=display(pp11,pp12):
...
# heunform,impoly,rk2,rk3,rk4,adambash,abmoulton (minimale Unterschiede)
display(pp);
```

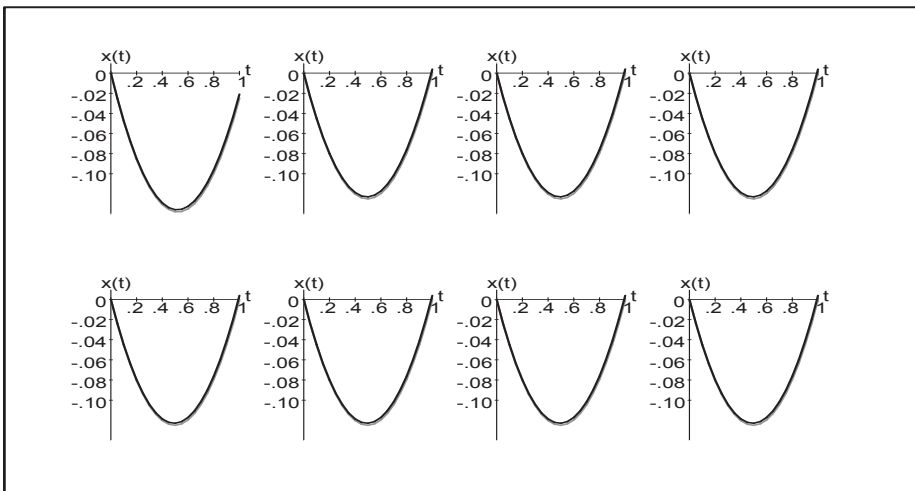


Abb. 7.78 Datei *nsteif34.ps*, AWP (7.42), AB mit $s = -0.5$, -0.504166 ,
`phaseportrait([sys5], ...)` für $(t, x(t))$ mit 8 klassischen Methoden

Wenden wir uns der Steifheit von gDGL zu.

Sie wird durch das Verhalten von Lösungen und Lösungskomponenten bzw. Lösungen zueinander charakterisiert. Diese finden wir bei gDGL 1. Ordnung als $x(t)$, bei SysgDGL als $\{x(t), y(t), \dots\}$ und gDGL der Ordnung größer 1 als $\{x(t), x'(t), \dots\}$.

GDGL 1. Ordnung nehmen dabei eine Sonderstellung ein. Zunächst ist es schon ein Unterschied, ob der Lösungsverlauf sich mehr wie $e^{\pm 10000t}$ oder eher wie $e^{\pm t}$ verhält und damit eine für den ganzen Verlauf der Näherungsrechnung erhaltene Schrittweitenbedingung mehr oder weniger streng ist. Schwieriger ist es, wenn man wegen der Kompliziertheit der gDGL keine solche Bedingung hat. Man beginnt die Näherungsrechnung mit einer Anfangsschrittweite und bemerkt aufgrund von Informationen aus dem Lösungsverhalten (Verfahrensmonitor, Grob- und Feinrechnung u.ä.) die Notwendigkeit ihrer Anpassung.

Nicht A-stabile Verfahren mit fester Schrittweite werden im Allgemeinen keine befriedigende Näherungslösung erzeugen. Sie benötigen eine SWS mit variablen Schrittweiten, die in Bereichen der Steifheit sehr klein werden. A-stabile Verfahren, die a priori keine Schrittweitenbeschränkung haben, kann man zwar nutzen, aber sie sind implizit und damit aufwändiger. Außerdem haben sie höchsten die Genauigkeitsordnung 2, was praktisch oft nicht ausreicht. Die Steifheit der gDGL zeigt sich in drastischen Veränderungen der Lösung und ihrer Ableitung, in auftretenden Nichtlinearitäten sowie der starken Abhängigkeit der partiellen Ableitungen der rechten Seite der gDGL von den Variablen. Eine mögliche Ansicht des Richtungsfeldes der gDGL kann nützliche Hinweise zum Lösungsverhalten geben. Darüber hinaus haben wir in Beispiel 7.2 (a) gesehen, dass sowohl Real- als auch Imaginärteil der Wachstumsgrößen als Parameter oder Eigenwerte der Systemmatrix eine wichtige Rolle spielen.

Diese Aspekte werden in den folgenden Beispielen untersucht. Dabei reicht es oft, mit einem System der Dimension zwei zu arbeiten. Bei höherdimensionalen SysgDGL können all die genannten Probleme gleichzeitig auftreten.

Beispiel 7.6

(a) Gegeben sei die nicht autonome nichtlineare gDGL 1. Ordnung als AWP

$$x'(t) = f(t, x(t)) = x^2(t) - t^6, \quad x(0) = -0.1, \quad t \geq 0. \quad (7.43)$$

Als erstes verweisen die partiellen Ableitungen der rechten Seite $f(t, x)$ auf mögliche Probleme und die Steifheit, denn in Bemerkung 4.2 (b) haben wir schon das explodierende Verhalten der Lösung der verwandten gDGL $x' = x^2$ festgestellt.

Maple kann das AWP unter Verwendung von Bessel-Funktionen 1. und 2. Art lösen.

```
> dgl1:=diff(x(t),t)=x(t)^2-t^6;          # Digits=16
  init:=x(0)=-0.1;
  loe1:=dsolve({dgl1,init},x(t));
  evalf(loe1);
```

Maple V

$$loe1 := x(t) = - \frac{t^3 \left(4.701307519771402 \operatorname{BesselI} \left(\frac{-7}{8}, \frac{1}{4} t^4 \right) - \operatorname{BesselK} \left(\frac{7}{8}, \frac{1}{4} t^4 \right) \right)}{4.701307519771402 \operatorname{BesselI} \left(\frac{1}{8}, \frac{1}{4} t^4 \right) + \operatorname{BesselK} \left(\frac{1}{8}, \frac{1}{4} t^4 \right)}$$

Maple 9.5

$$\begin{aligned} &loe1 := x(t) = \\ &t^3 \left(\frac{\frac{1}{160} \pi \left(10 \cdot 2^{\left(\frac{3}{4}\right)} \cdot 4^{\left(\frac{3}{4}\right)} \Gamma \left(\frac{7}{8} \right)^2 + \pi \operatorname{csc} \left(\frac{\pi}{8} \right) \right) 2^{1/4} 4^{1/4} \operatorname{BesselI} \left(\frac{-7}{8}, \frac{t^4}{4} \right)}{\Gamma \left(\frac{7}{8} \right)^2 \sin \left(\frac{\pi}{8} \right)} - \operatorname{BesselK} \left(\frac{7}{8}, \frac{t^4}{4} \right) \right) \\ &\frac{\frac{1}{160} \pi \left(10 \cdot 2^{\left(\frac{3}{4}\right)} \cdot 4^{\left(\frac{3}{4}\right)} \Gamma \left(\frac{7}{8} \right)^2 + \pi \operatorname{csc} \left(\frac{\pi}{8} \right) \right) 2^{1/4} 4^{1/4} \operatorname{BesselI} \left(\frac{1}{8}, \frac{t^4}{4} \right)}{\Gamma \left(\frac{7}{8} \right)^2 \sin \left(\frac{\pi}{8} \right)} + \operatorname{BesselK} \left(\frac{1}{8}, \frac{t^4}{4} \right) \\ &x(t) = \\ &\frac{1. t^3 (4.701307519771402 \operatorname{BesselI}(-0.8750\dots, 0.2500\dots t^4) - 1. \operatorname{BesselK}(0.8750\dots, 0.2500\dots t^4))}{4.701307519771402 \operatorname{BesselI}(0.1250\dots, 0.2500\dots t^4) + \operatorname{BesselK}(0.1250\dots, 0.2500\dots t^4)} \end{aligned}$$

Die Lösung definieren wir als Funktion. Dabei muss man diese an der Anfangsstelle 0 wegen der dortigen Singularität einiger Bessel-Funktionen extra behandeln.

Die Darstellung zeigt eine Ähnlichkeit zum Polynom $-t^3$. Im Verlauf der Lösung ist wegen der Gefahr des verschwindenden Nenners die Möglichkeit des Auftretens von Polstellen zu berücksichtigen.

```
> xx:=unapply(rhs(loe1),t):          # exakte Loesungsfunktion

xx(0);          # Error, (in BesselI) numeric exception: division by zero
limit(xx(t),t=0);          # Maple 9.5:   undefined
                        # Maple V:     -.099999999999999968

limit(xx(t),t=0,right): evalf(%); # -.099999999999999983
limit(xx(t),t=0,left):  evalf(%); # -.099999999999999983
```

Betrachten wir zuerst das Richtungsfeld mit

```
p1:=dfieldplot(dgl1,x(t),t=-3..3,x=-20..20,arrows=large,dirgrid=[30,20]).
```

Da die Anstiege der Linienelemente genau an den Trennkurven $x^2 - t^6 = 0$ ihr Vorzeichen ändern, sind zusätzlich diese Kurven $x = \pm t^3$ eingezeichnet. Sie sind zwar keine Lösungstrajektorien, aber kommen solchen bei $t \gg 0$ sehr nahe.

Man erkennt stabile und instabile Gleichgewichtslagen sowie extrem steile Linienelemente, die an stabilen Gleichgewichtslagen fast vertikal aufeinander treffen (Vorzeichenwechsel bei x'). Für wachsendes $t > 2$ kommen sich diese Linienelemente von oben und unten immer näher.

Da an der stabilen Gleichgewichtslage die zugehörigen Lösungstrajektorien sich zwar asymptotisch verhalten, aber betragsmäßig stark wachsen, werden bedingt stabile ESV und MSV mit konstanter Schrittweite "früher oder später" versagen müssen. Die Schrittweitenbeschränkung wird irgendwann verletzt und das Verfahren divergiert.

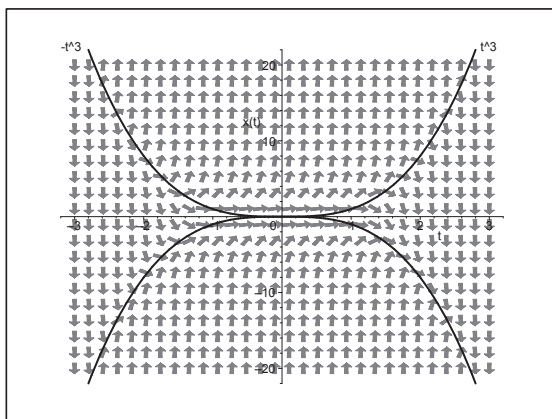


Abb. 7.79

Datei `nsteif35.ps`,
Richtungsfeld zur gDGI
des AWP (7.43),
Trennkurven $x = \pm t^3$,
Kommando `dfieldplot`

Noch deutlicher erkennt man die beschriebene Situation, wenn man einige Lösungstrajektorien dazu berechnet und einzeichnet.

Die AB seien $x(0) = 4, 2, 1, 0.7, 0.689, 0.688, \dots, 0.687992, 0.687991, 0.687, -0.1$.

Die Lösungstrajektorien mit größeren AB "explodieren" wertemäßig ins unendliche. Sie haben weiterhin eine Polstelle, somit eine Asymptote und die Funktion ist auch rechts davon definiert. Die Scheitelpunkte der "unteren" Funktionszweige liegen auf der Trennkurve $-t^3$. Mit entsprechenden AB werden diese Funktionen als Lösungen von der unteren stabilen Gleichgewichtslage angezogen.

AB $x(0) = s$	Polstelle bei t
4	0.250 000
2	0.500 007
1	1.004 065
0.7	1.643 025
0.689	1.874 810
0.688	2.166 494
0.687 999	2.172 811
0.687 998	2.179 992
0.687 993	2.252 903
0.687 992	2.315 404
0.687 991	keine
< 0.687 991	keine

Tab. 7.5

Polstellen zu Lösungen des AWP (7.43)
mit verschiedenen AB $x(0) = s$

Es macht natürlich keinen Sinn, die numerische Lösung des AWP über die Unstetigkeitsstellen hinweg durchzuführen.

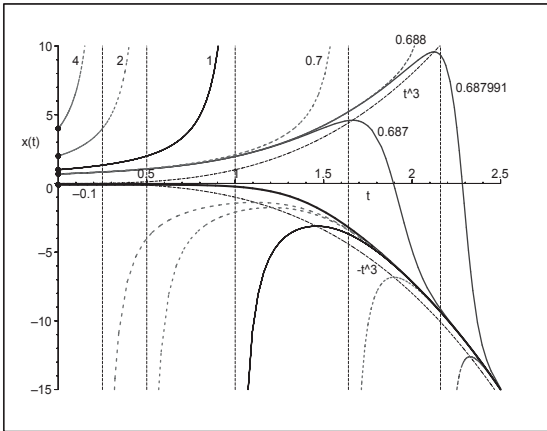


Abb. 7.80

Datei *nsteif39.ps*,
Lösungstrajektorien
des AWP (7.43) mit AB
 $x(0) = -0.1, 4, 2, 1, 0.7,$
 $0.688, 0.687991, 0.687$
und Asymptoten,
dazu Trennkurven $x = \pm t^3$
und zusätzliche Funktionszweige

```
> p1:=DEplot(dgl1,x(t),t=0..2.5,[[init],[x(0)=-15],[x(0)=4],[x(0)=2],
[x(0)=1],[x(0)=0.7],[x(0)=0.688],[x(0)=0.687991],[x(0)=0.687],
[x(2)=-10],[x(2.2)=-15]],x=-15..10,stepsize=0.01,method=classical[rk4],
linecolor=[black,blue,blue,blue,blue,blue,blue,cyan,cyan,brown,brown]):
p2:=pointplot([[0,-0.1],[0,-15],[0,0.7],[0,0.688],[0,1],[0,2],[0,4],
[2,-10],[2.2,-15]],symbol=circle,color=black):
display(p1,p2);
```

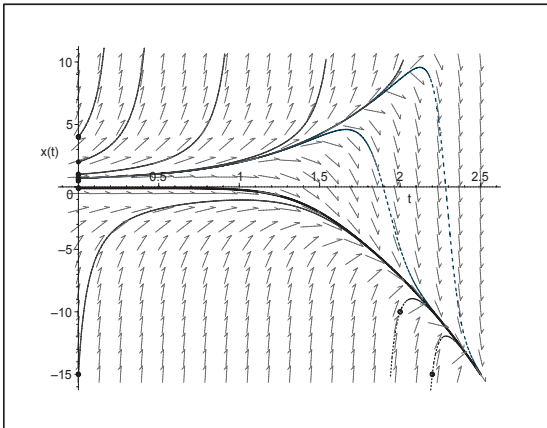


Abb. 7.81

Datei *nsteif36.ps*,
Richtungsfeld zur gDGI
des AWP (7.43),
Näherungslösungen mit den AB
 $x(0) = -0.1, -15, 4, 2, 1, 0.7,$
 $0.688, 0.687991, 0.687,$
 $x(2) = -10, x(2.2) = -15,$
Kommando `DEplot`

Nun lösen wir das AWP numerisch unter Verwendung von `dsolve` mit den Methoden

- `rkf45` (Standard, Fehlberg fourth-fifth order Runge-Kutta method, `abserr`= 10^{-7}),
- `gear` (Gear single-step extrapolation method, `abserr`= 10^{-7}),
- `dverk78` (seventh-eighth order continuous Runge-Kutta method, `abserr`= 10^{-8}),
- `rosenbrock` (implicit Rosenbrock third-fourth order Runge-Kutta, `abserr`= 10^{-6} , mit Option `stiff=true`).

Wir wählen zwecks Vergleichbarkeit überall die absolute Fehlertoleranz `abserr=1e-7`. Die Ergebnisse sind in Tabellenform, sodass man damit die grafische Auswertung vornehmen kann. Die Näherungslösungen sind allesamt so genau, dass sie sich grafisch von der exakten Lösung nicht unterscheiden lassen.

```
> T:=2.5; N:=25; h:=T/N;
> dsol1:=dsolve({dgl1,init},x(t),numeric, # method=rkf45 ist Standard
               abserr=1E-7,output=array([seq(i*h,i=0..N)]));
  ls1:=eval(dsol1[2,1]): # Tabelle mit [t,x(t)]
  nr:=rowdim(ls1);      # N+1
  lls1:=[seq([ls1[i,1],ls1[i,2]],i=1..nr)]:
  # analog
  # method=gear, method=dverk78, stiff=true (rosenbrock anstelle von rkf45)

> lexg:=[-0.1,seq(evalf(xx(i*h*0.5)),i=1..2*N)]:
  llexg:=[seq([i*h*0.5,lexg[i+1]],i=0..2*N)]:
> p1:=plot([llexg,lls1,lls2,lls3,lls4],color=[black,blue,red,green,magenta],
           linestyle=[1,2,3,4,4],thickness=2):
  display(p1,labels=['t',''],
         title='x(t) und Naehierungen aus rkf45, gear, dverk78, rosenbrock(stiff)');
```

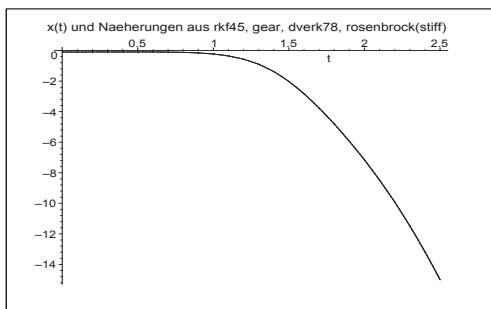


Abb. 7.82

Datei *nsteif37.ps*,
Lösung $x(t)$, $t \in [0, 2.5]$, zu (7.43)
sowie Näherungslösungen mit `rkf45`,
`gear`, `dverk78`, `rosenbrock(stiff)`,
Kommando `dsolve/numeric`

Günstiger ist ein numerischer Vergleich. Man sieht, dass das Gear-Verfahren am besten abschneidet und das speziell für steife Probleme empfohlene Rosenbrock-Verfahren schlechter und nicht viel besser als RKF45 ist.

```
> fprintf(default, ' t      x(t)-x(rkf45)      x(t)-x(gear)      x(t)-x(dverk78)
                  x(t)-x(stiff)      \n'):
  fprintf(default, '
                                     stiff=rosenbrock\n'):
  fprintf(default, '-----
                                     -----\n'):

  for i from 1 to nr do
    fprintf(default, '%3.1f %17.14f %17.14f %17.14f %17.14f \n', (i-1)*h,
              llex[i][2]-lls1[i][2], llex[i][2]-lls2[i][2],
              llex[i][2]-lls3[i][2], llex[i][2]-lls4[i][2]):
  end do:
```

t	x(t)-x(rkf45)	x(t)-x(gear)	x(t)-x(dverk78)	x(t)-x(stiff) stiff=rosenbrock
0.0	-0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
0.1	0.00000016905736	-0.000000000000000	0.000000000000000	-0.00000000127323
0.2	-0.00000005186162	-0.000000000000003	0.000000000000003	-0.00000004177038
0.3	0.00000117412020	-0.000000000000004	0.000000000000004	-0.00000005791164
0.4	-0.00000031864491	-0.000000000000178	0.000000000000309	-0.00000005724140
0.5	0.00000053629647	-0.000000000001295	0.00000000044261	-0.00000002839484
0.6	0.00000031546491	-0.00000000001265	0.00000000003058	-0.00000004882850
0.7	-0.00000033802165	-0.00000000001234	0.00000000059221	-0.00000005114452
0.8	-0.00000041867334	-0.00000000001208	0.00000000042353	0.00000000676051
0.9	-0.00000041703858	-0.00000000000589	0.00000000287195	0.00000000335879
1.0	-0.00000036916536	-0.00000000000595	0.00000000287422	-0.00000000593579
1.1	-0.00000030448658	0.00000000035281	0.00000000495726	-0.00000008728480
1.2	-0.00000032452329	0.00000000032602	0.00000000382640	0.00000000878593
1.3	-0.00000059348770	0.00000000047937	0.00000000437093	0.00000007083672
1.4	-0.00000080533875	0.00000000043230	0.00000001016220	0.00000044902201
1.5	-0.00000082734155	0.00000000061307	0.00000000959554	-0.00000014967193
1.6	-0.00000103610247	0.00000000068615	0.00000000648726	0.00000030306288
1.7	-0.00000140284812	0.00000000061330	0.00000000363566	-0.00000082054182
1.8	-0.00000154597091	0.00000000041456	0.00000000188713	-0.00000132398911
1.9	-0.00000232672866	0.00000000052781	0.00000000146561	-0.00000003359037
2.0	-0.00000209312064	0.00000000161115	0.00000000066557	0.00000005381828
2.1	-0.00000230283758	0.00000000045419	0.00000000031600	0.00000030005956
2.2	-0.00000321323206	0.00000000049394	0.00000000024306	0.00002388869983
2.3	-0.00000391350690	0.00000000025028	0.00000000010889	-0.00001252822380
2.4	-0.00000501823290	0.00000000009073	0.00000000013771	-0.00000103059298
2.5	-0.00000567194561	-0.00000000001657	0.00000000008328	0.00000307665675

In den Abschnitten 5.1 und 5.2 haben wir die RKV unter Verwendung einer vorgegebenen festen Schrittweite in MATLAB implementiert. Sie sollen hier zur Demonstration des Einflusses der Steifigkeit auf die Näherungslösung herangezogen werden. Es reicht aus, im Integrationsintervall $[0, 2.5]$ zu rechnen, denn nahe dem rechten Rand erkennt man schon das Problem des divergenten Verhaltens.

```

t0 = 0;
T = 2.5;
x0 = -0.1;
N = 10;           % interessant: 15(-1)10

[te,xe] = euler1('f15',t0,T,x0,N);      % PZV
[th,xh] = heun1('f15',t0,T,x0,N);      % HV
A = [0 0 0; 1/2 0 0; 1/3 2/3 0];
b = [1/3 1/3 1/3]';
c = [0 1/2 1]';
[t3,x3] = rk1(A,b,c,'f15',t0,T,x0,N);  % (3,3)-RKV
[t4,x4] = krkv1('f15',t0,T,x0,N);      % KRKV

```

Wir nehmen numerische und grafische Auswertungen der Ergebnisse im Vergleich mit der exakten Lösung vor. So ermitteln wir $x(t_N)$ und für die vier ESV PZV, HV, (3,3)-RKV, KRKV jeweils η_N , sowie die Fehler $\max_{i=0,\dots,N} |\eta_i - x(t_i)|$ und $\max_{i=0,\dots,N} |\eta_i - x(t_i)|/|x(t_i)|$.

	PZV	HV	(3,3)-RKV	KRKV	$x(t)$
N=15 -----					
Endwerte bei T	-14.811580	-28.802543	-15.012601	-18.020216	-15.003426
absolute Fehler	0.191845	13.799117	0.009174	3.016790	
relative Fehler	0.012786	0.919731	0.000611	0.201073	
N=14 -----					
Endwerte bei T	-14.343845	-27.730138	-15.047898	-18.474127	-15.003426
absolute Fehler	0.659580	12.726712	0.044471	3.470700	
relative Fehler	0.043962	0.848253	0.002964	0.231327	
N=13 -----					
Endwerte bei T	-13.883086	-18.628411	-15.126373	-15.078151	-15.003426
absolute Fehler	1.120339	3.624985	0.122947	0.074724	
relative Fehler	0.074672	0.241610	0.008194	0.004980	
N=12 -----					
Endwerte bei T	-14.265456	9.009299	-15.289875	6.339675	-15.003426
absolute Fehler	0.737969	24.012725	0.286449	21.343101	
relative Fehler	0.049186	1.600482	0.019092	1.422548	
N=11 -----					
Endwerte bei T	-16.288058	73.975292	-15.603578	99.877508	-15.003426
absolute Fehler	1.284631	88.978718	0.600152	114.880934	
relative Fehler	0.085622	5.930559	0.040000	7.656979	
N=10 -----					
Endwerte bei T	-18.953311	206.675574	-16.142984	449.731710	-15.003426
absolute Fehler	3.949884	221.679000	1.139558	464.735136	
relative Fehler	0.263265	14.775224	0.075953	30.975266	

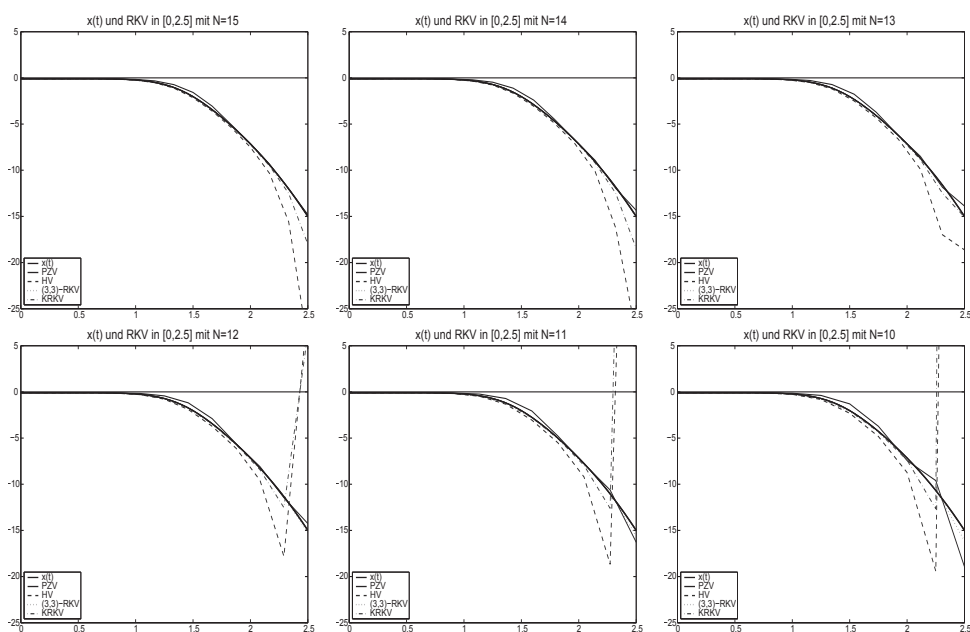


Abb. 7.83 Datei *steif_01..06.ps*, AWP (7.43), Lösung $x(t)$, $t \in [0, 2.5]$, und Näherungslösungen η_i mit den Verfahren PZV, HV, (3,3)-RKV, KRKV bei einer Teilintervallanzahl $N = 15(-)10$

Das (3,3)-RKV zeigt sich im Vergleich mit den anderen ESV am wenigsten empfindlich auf die Steifheit.

Zur Kontrolle wollen wir zum AWP (7.43) das A-stabile IEV

$$\eta_{i+1} = \eta_i + hf(t_{i+1}, \eta_{i+1}), \quad i = 0, 1, \dots, N-1, \quad \eta_0 = -0.1,$$

noch implementieren. Für $f(t, x) = x^2 - t^6$ lässt sich die rekursive Formel explizit nach η_{i+1} auflösen, denn sonst wäre eine iterative Lösung erforderlich. Wir erhalten

$$\eta_{i+1} = \frac{1}{2h} \pm \sqrt{\left(\frac{1}{2h}\right)^2 - \frac{1}{h}\eta_i + t_{i+1}^6}.$$

Von den beiden Näherungslösungen ist nur

$$\eta_{i+1} = \frac{1}{2h} - \sqrt{\left(\frac{1}{2h}\right)^2 - \frac{1}{h}\eta_i + t_{i+1}^6}.$$

relevant. Die andere erzeugt eine komplexwertige Folge.

```
> T:=2.5; N:=15; h:=T/N;

> eta1:=-0.1;
eta[0]:=eta1;
eta2:=-0.1;
eta2r:=Re(eta2);
eta2i:=Im(eta2);
fmax:=0;
fprintf(default, ' i          eta1_i          x(t_i)      |x(t_i)-eta1_i|
                eta2_i          \n'):
fprintf(default, '
                Re          Im          \n'):
fprintf(default, '-----
                \n'):
fprintf(default,
'%2g %12e %12e %12e %12e %12e \n', 0, eta1, eta1, 0, eta2r, eta2i):
for i from 1 to N do
eta1:=1/(2*h)-sqrt(1/(4*h^2)-eta1/h+(i*h)^6); # relevante Erstloesung
eta[i]:=eta1;
eta2:=1/(2*h)+sqrt(1/(4*h^2)-eta2/h+(i*h)^6); # komplexe Zweitloesung
eta2r:=Re(eta2);
eta2i:=Im(eta2);
ex1:=evalf(xx(i*h));
absf:=abs(ex1-eta1);
if absf>fmax then fmax:=absf; end if;
fprintf(default,
'%2g %12e %12e %12e %12e %12e \n', i, eta1, ex1, absf, eta2r, eta2i):
end do;
fprintf(default, '-----
                \n'):
fprintf(default, '
                max |x(t_i)-eta1_i| = %12e \n', fmax):
```


i	eta1_i	x(t_i)	x(t_i)-eta1_i	Re	eta2_i	Im
0	-1.000000e-01	-1.000000e-01	0.000000e+00	-1.000000e-01	0.000000e+00	
1	-9.839014e-02	-9.836116e-02	2.897180e-05	6.098390e+00	0.000000e+00	
2	-9.704901e-02	-9.683899e-02	2.100182e-04	3.000000e+00	5.252520e+00	
3	-9.805085e-02	-9.634091e-02	1.709936e-03	6.448650e+00	-4.569197e+00	
4	-1.106425e-01	-1.019784e-01	8.664102e-03	5.317792e+00	5.914074e+00	
5	-1.620805e-01	-1.313322e-01	3.074822e-02	6.121160e+00	-5.684496e+00	
6	-3.124738e-01	-2.292790e-01	8.319480e-02	5.881363e+00	5.918548e+00	
7	-6.601187e-01	-4.865788e-01	1.735399e-01	6.079286e+00	-5.766156e+00	
8	-1.310379e+00	-1.042616e+00	2.677635e-01	6.087538e+00	5.602674e+00	
9	-2.315346e+00	-2.026667e+00	2.886785e-01	6.252151e+00	-5.168278e+00	
10	-3.657743e+00	-3.443283e+00	2.144603e-01	6.516253e+00	4.409476e+00	
11	-5.301636e+00	-5.177583e+00	1.240533e-01	7.211690e+00	-3.140883e+00	
12	-7.237667e+00	-7.162542e+00	7.512485e-02	8.697782e+00	1.653740e+00	
13	-9.485242e+00	-9.428831e+00	5.641144e-02	1.078937e+01	-6.369221e-01	
14	-1.207632e+01	-1.202895e+01	4.737655e-02	1.328020e+01	1.858687e-01	
15	-1.504435e+01	-1.500343e+01	4.092349e-02	1.617047e+01	-4.233760e-02	

$$\max |x(t_i)-\eta_{1,i}| = 2.886785e-01$$

Rechnungen mit verschiedenen N ergeben die folgende Grafik mit erkennbarer Gutartigkeit des IEV. Das Integrationsintervall kann dabei beliebig groß sein. Allein eine gewünschte gute Genauigkeit wird die Wahl einer kleinen Schrittweite erfordern. Das aber bedeutet eine immens hohe Schrittzahl, nicht nur bei steifen Problemen.

T=100	N	x(t_N)-eta1_N	max x(t_i)-eta1_i
	100	1.495003e-04	4.326160e-01 bei i=1
	200	7.487530e-05	6.285409e-01 bei i=3
	500	2.998030e-05	3.355369e-01 bei i=7
	1000	1.499530e-05	1.878222e-01 bei i=15
	2000	7.499300e-06	1.007036e-01 bei i=29
	5000	3.000300e-06	4.198805e-02 bei i=74

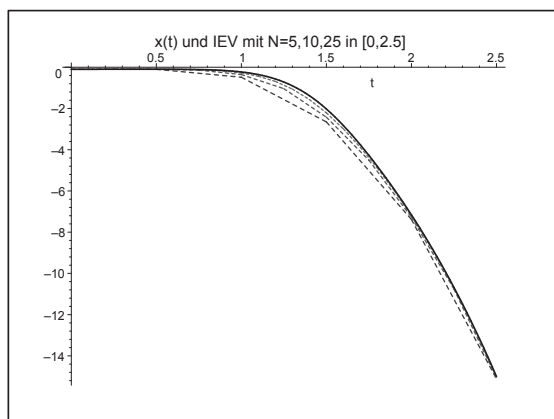


Abb. 7.84

Datei *nsteif38.ps*,

AWP (7.43),

Lösung $x(t)$, $t \in [0, 2.5]$, und

Näherungslösungen η_i

mittels IEV bei einer

Teilintervallanzahl $N = 5, 10, 25$

Zur Untersuchung der Steifheit und Schrittweitenbedingung nimmt man die Matrix A des SysgDGL (7.45). Ihre beiden konjugiert komplexen Eigenwerte sind

$$\lambda_{1,2} = \frac{-1 \pm i\sqrt{39999}}{200} = -0.005 \pm i0.9999874999218740.$$

Beide haben den gleichen Betrag Eins und die Schrittweitenbedingung zum PZV ist

$$h < \frac{1}{100}.$$

Analog ist die Lösung des SysgDGL (7.45) allgemein und mit den gegebenen AB.

```
> sys2:=diff(x(t),t)=y(t),diff(y(t),t)=-x(t)-y(t)/100:    # Digits=10
  fcns:=x(t),y(t):
  init:=x(0)=0,y(0)=1:
  loe2:=dsolve({sys2},{fcns});
  evalf(loe2);
> loe2a:=dsolve({sys2,init},{fcns});
  evalf(loe2a);

> # Loesungsfunktionen
xx:=unapply(rhs(op(1,loe2a)),t):    # Reihenfolge x,y beachten
evalf(xx(t)):
yy:=unapply(rhs(op(2,loe2a)),t):
evalf(yy(t)):
```

$$loe2 := \left\{ x(t) = e^{\left(-\frac{t}{200}\right)} \left({}_C1 \sin\left(\frac{\sqrt{39999}t}{200}\right) + {}_C2 \cos\left(\frac{\sqrt{39999}t}{200}\right) \right), \right.$$

$$y(t) = \frac{1}{200} e^{\left(-\frac{t}{200}\right)} \left(-{}_C1 \sin\left(\frac{\sqrt{39999}t}{200}\right) + {}_C1 \cos\left(\frac{\sqrt{39999}t}{200}\right) \sqrt{39999} \right. \\ \left. - {}_C2 \cos\left(\frac{\sqrt{39999}t}{200}\right) - {}_C2 \sin\left(\frac{\sqrt{39999}t}{200}\right) \sqrt{39999} \right) \left. \right\}$$

$$\{ x(t) = e^{(-0.005000000000t)} ({}_C1 \sin(0.9999875000t) + {}_C2 \cos(0.9999875000t)),$$

$$y(t) = 0.005000000000 e^{(-0.005000000000t)} (-1. {}_C1 \sin(0.9999875000t) + 199.9975000 \\ {}_C1 \cos(0.9999875000t) - 1. {}_C2 \cos(0.9999875000t) - 199.9975000 {}_C2 \sin(0.9999875000t)) \}$$

$$loe2a := \left\{ x(t) = \frac{200}{39999} \sqrt{39999} e^{\left(-\frac{t}{200}\right)} \sin\left(\frac{\sqrt{39999}t}{200}\right), \right.$$

$$y(t) = \frac{1}{200} e^{\left(-\frac{t}{200}\right)} \left(-\frac{200}{39999} \sin\left(\frac{\sqrt{39999}t}{200}\right) \sqrt{39999} + 200 \cos\left(\frac{\sqrt{39999}t}{200}\right) \right) \left. \right\}$$

$$\left. \begin{aligned} x(t) &= 1.000012500 e^{(-0.005000000000 t)} \sin(0.9999875000 t), \\ y(t) &= 0.005000000000 e^{(-0.005000000000 t)} (-1.000012500 \sin(0.9999875000 t) + \\ &\quad 200. \cos(0.9999875000 t)) \end{aligned} \right\}$$

Zum AWP (7.45) betrachten wir mittels `DEplot` das Richtungsfeld mit numerischen Lösungen sowie die exakten Lösungen $x(t)$, $y(t) = x'(t)$ in vorgegebenen Bereichen.

Der `DEplot`-Parameter `stepsize`, der zur Auswertung der Näherungslösung an Zwischenpunkten und zur grafischen Ausgabe gedacht ist, dient damit indirekt der Verwendung von hinreichend kleinen Schrittweiten. Das ist wichtig bei steifen SysgDGI, wenn als Standardmethode `method=rkf45` oder andere ESV genutzt werden.

Das KRKV hat nämlich eine Schrittweitenbedingung, im reellen Fall $h < 2.785/|\lambda|_{max}$. Man kann auch die vom PZV $h < 2/|\lambda|_{max}$ verwenden. Wenn `stepsize` fehlt bzw. zu groß ist, ist das als Methode genutzte ESV instabil und es droht die Gefahr eines Zahlenüberlaufs und von Fehlern, z.B. `Floating Point Overflow. Please shorten axes`.

```
> p11:=DEplot({sys2}, [fcns], t=0..4*Pi, [[x(0)=0,y(0)=1], [x(0)=.5,y(0)=-.5]],
  stepsize=0.01, linecolor=[black,blue]), labels=['x', 'x']):
p12:=pointplot([[0,1], [0.5,-0.5]], symbol=circle):
display(p11,p12);

> p1:=plot([xx(t),yy(t)], t=0..10, color=[black,blue], thickness=2):
p2:=plot([1,1.0000125*exp(-0.005*t), -1], t=0..10, color=black,
  thickness=1, linestyle=[1,2,1]):
p3:=textplot([[3.1,0.8, 'x(t)'], [1.3,-0.8, 'y(t)=x'(t)']]):
display(p1,p2,p3);

> p1:=plot([xx(t),yy(t)], t=0..50, color=[black,blue], thickness=2):
p2:=plot([1,1.0000125*exp(-0.005*t), -1], t=0..50, color=black,
  thickness=1, linestyle=[1,2,1]):
p3:=textplot([[3.4,0.85, 'x(t)'], [1.5,-0.8, 'y(t)']]):
display(p1,p2,p3);
```

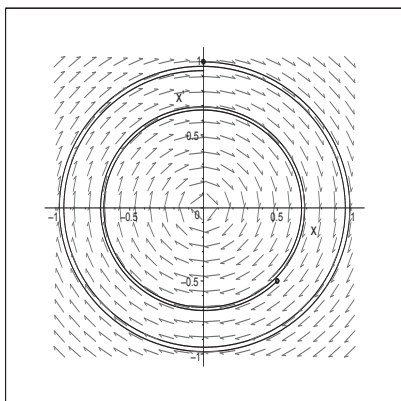


Abb. 7.86

Datei `nsteif41.ps`,

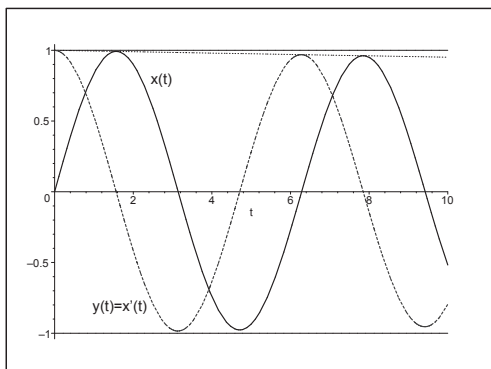
Richtungsfeld zum AWP (7.45)

mit zwei numerischen Lösungen

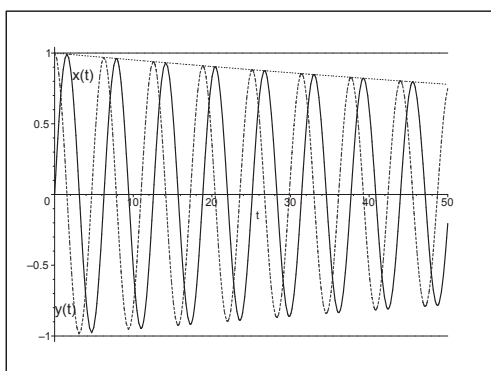
zu den AB $(x(0), y(0)) = (0, 1)$,

$(x(0), y(0)) = (0.5, -0.5)$,

Kommando `DEplot`

**Abb. 7.87**

Datei *nsteif42.ps*,
AWP (7.45) mit Lösungen
 $x(t), y(t), t \in [0, 10]$,
dazu die begrenzenden Geraden ± 1 und
zum Vergleich die Dämpfungsfunktion
 $1.0000125 e^{-0.005t}$

**Abb. 7.88**

Datei *nsteif43.ps*,
AWP (7.45) mit Lösungen
 $x(t), y(t), t \in [0, 50]$

Das PZV zur numerischen Lösung des AWP hat die Gestalt

$$\eta_{i+1} = A_h \eta_i, \quad \eta_i = (\eta_{1i}, \eta_{2i})^T, \quad \eta_0 = (0, 1)^T,$$

mit der Übergangs- oder Verstärkungsmatrix

$$A_h = I + hA = \begin{pmatrix} 1 & h \\ -h & 1 - \frac{h}{100} \end{pmatrix}.$$

Zwecks Stabilität müssen die beiden konjugiert komplexen Eigenwerte

$$\begin{aligned} \lambda_{1,2}(A_h) &= 1 - \frac{h}{200} \pm i h \frac{\sqrt{39999}}{200} \\ &= 1 - 0.005 h \pm i 0.9999874999218740 h \end{aligned}$$

betragsmäßig kleiner Eins sein. Ihr Betrag ist

$$|\lambda_{1,2}(A_h)| = \frac{1}{10} \sqrt{100 - h + 100h^2}.$$

Wir zeichnen den Betrag als Funktion von kleinen Werten $h \geq 0$.

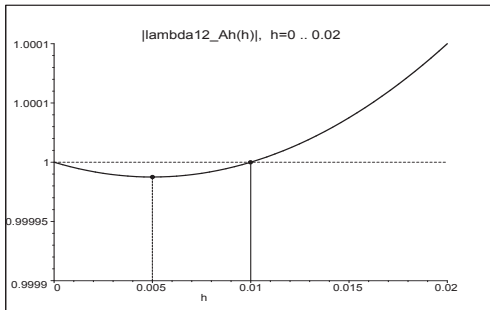


Abb. 7.89
 Datei *nsteif44.ps*,
 Betrag der EW der Übergangsmatrix
 des PZV zum AWP (7.45)
 als Funktion von $h \geq 0$

Die obere Schranke der Schrittweite ist $\frac{1}{100}$.

Für $0 < h < \frac{1}{100}$ sind die Beträge der Eigenwerte als Wachstumsfaktoren kleiner als Eins, für $h = \frac{1}{100}$ genau Eins. Bei $h = \frac{1}{200}$ ist der Wachstumsfaktor mit $\frac{\sqrt{39999}}{200} = 0.999987500$ am kleinsten und die Kontraktion am größten. Für $h \geq \frac{1}{100}$ ist das PZV instabil.

Der Test im Intervall $[0,50]$ mit den Schrittzahlen $N = 1000, 5000, 10000, 50000$ bestätigt die Stabilitätsbedingung.

Darüber hinaus erhalten wir aus dem Vergleich der Lösungskomponenten η_{1i} bzw. η_{2i} des PZV mit den exakten Lösungen $x(t)$ bzw. $y(t)$ die Erkenntnis, dass die Näherungsrechnung mit der Schrittweite $h = 0.005$ ($N = 10000$) der kleinsten Kontraktion weniger genau ist als eine mit einer Schrittweite $h < 0.005$ bei schlechterer Kontraktion.

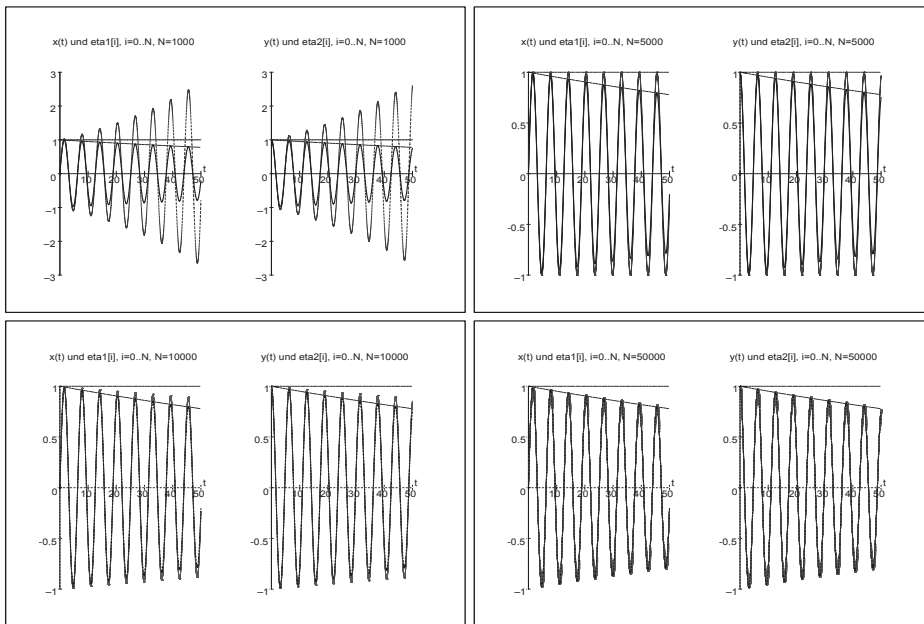


Abb. 7.90 Dateien *nsteif45..48.ps*, AWP (7.45) bei $t \in [0, 50]$,
 Näherungslösungen η_{1i}, η_{2i} mit PZV bei $h = 0.05, 0.01, 0.005, 0.001$
 im Vergleich mit den exakten Lösungskomponenten $x(t), y(t)$

7.3.1 Systeme mit ausgeprägter Steifheit

Es geht dabei um ein lineares SysgDGL $x' = Ax$, wo die Eigenwerte der Matrix A negative Realteile haben und die Steifheit durch sehr unterschiedliche Beträge der Eigenwerte und damit mehr oder weniger stark abnehmendes Verhalten der Lösungskomponenten verursacht wird. In der Praxis bezeichnet man SysgDGL mit $\frac{\max |\lambda(A)|}{\min |\lambda(A)|} \geq 10^4$ als steif. Von großer Bedeutung ist der Eigenwert mit dem betragsgrößten negativen Realteil.

Die Schrittweitenbeschränkung (7.13) ist bei $\lambda(A) = s + i\omega$, $s \ll 0$,

$$h < \frac{2|\Re\lambda|}{|\lambda|^2} = -\frac{2s}{s^2 + \omega^2} \approx -\frac{2}{s}$$

und kann damit $h \ll 1$ nach sich ziehen.

Beispiel 7.7

(a) Gegeben sei als AWP die homogene autonome lineare gDGL 2. Ordnung

$$x'' + 100x' + x = 0, \quad x(0) = 0, \quad x'(0) = 1, \quad t \in [0, T]. \quad (7.46)$$

Mit $x' = y$ erhält man die Darstellung als homogenes lineares SysgDGL 1. Ordnung

$$\begin{aligned} x'(t) &= y(t), & x(0) &= 0, \\ y'(t) &= -x(t) - 100y(t), & y(0) &= 1, \\ \mathbf{x}'(t) &= A\mathbf{x}(t), & A &= \begin{pmatrix} 0 & 1 \\ -1 & -100 \end{pmatrix}, & \mathbf{x}(0) &= (0, 1)^T. \end{aligned} \quad (7.47)$$

Wir bestimmen die exakte Lösung des AWP (7.46).

```
> dgl3:=diff(x(t),t,t)+100*diff(x(t),t)+x(t): # Digits=16
init:=x(0)=0,D(x)(0)=1:
loe3:=dsolve({dgl3,init},x(t));
evalf(loe3);
xx:=unapply(rhs(loe3),t): # Loesungsfunktion
```

$$loe3 := \frac{1}{714} \sqrt{51} e^{(-50 + 7\sqrt{51})t} - \frac{1}{714} \sqrt{51} e^{-(50 + 7\sqrt{51})t}$$

$$x(t) = 0.01000200060020007 (e^{-0.01000100020005t} - e^{-99.98999899979995t})$$

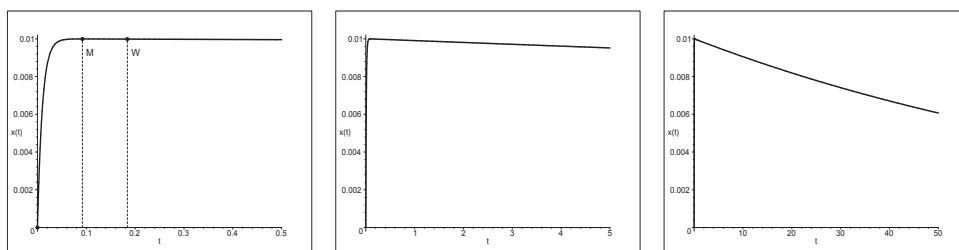


Abb. 7.91 Dateien *nsteif57..59.ps*, AWP (7.46), Lösung $x(t)$, $t \in [0, T]$, $T = 0.5, 5, 50$

Die Lösungskurve $x(t)$ hat einen vergleichsweise steilen Anfangsanstieg. An der Stelle $t_M = 0.092119$ ist ihr Maximum M mit $x(t_M) = 0.009991$ sowie bei $t_W = 0.184239$ ein Wendepunkt W mit $x(t_W) = 0.009983$ und $x'(t_W) = -0.0000998$.

Wollen wir die Lösungsfunktion $x(t)$, ihre Ableitungen sowie Phasenkurven betrachten, so ist es notwendig, geeignete Darstellungsbereiche zu wählen. Aus einer gemeinsamen Grafik von $x(t)$, $x'(t)$, $x''(t)$ ist wegen $x''(0) = -100$ und $|x(t)| \ll 1$, $|x'(t)| \leq 1$ nicht viel von den Funktionen $x(t)$, $x'(t)$ zu erkennen.

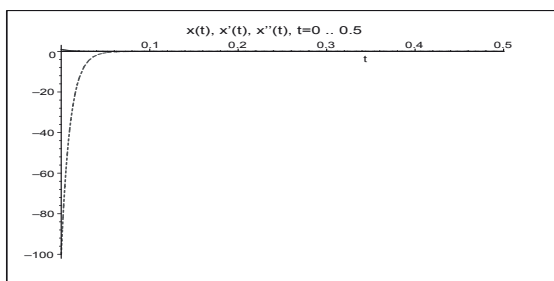


Abb. 7.92

Datei *nsteif60.ps*,
AWP (7.46),
Lösung $x(t)$, $t \in [0, 0.5]$ mit
den ersten beiden Ableitungen

Ähnlich gelingt es, die problematischen Teile der Phasenkurve (x, x') durch die Wahl eines günstigen Zeitbereichs zu erfassen. Wir wollen uns daher systematisch in die Problemzone der “unteren rechten Ecke“ hineinzoomen.

```
> opt1:=color=blue,thickness=3:
> plot([xx(t),xx1(t),t=0..0.5], opt1,view=[0..0.01,-0.1..1]);
plot([xx(t),xx1(t),t=0..5], opt1,view=[0..0.01,-0.1..1]);
plot([xx(t),xx1(t),t=0..50], opt1,view=[0..0.01,-0.1..1]);
plot([xx(t),xx1(t),t=0.06..20],opt1,view=[0.008..0.010,-0.0005..0.0020]);
```

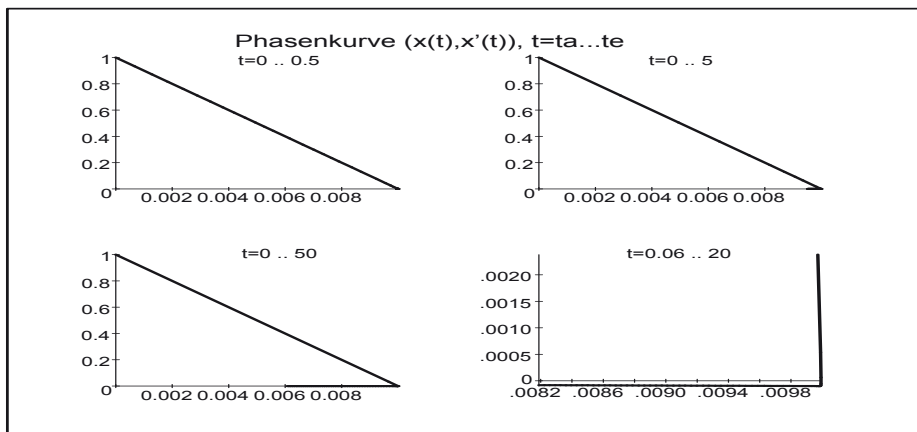


Abb. 7.93 Datei *nsteif61.ps*, Phasenkurve ($x(t), x'(t)$) in Zeitintervallen $[t_a, t_e]$

Damit ist zu erwarten, dass anhand der letzten beiden Grafen die Darstellung des Richtungsfeldes im Eckbereich wegen des “Knicks“ äußerst schwierig sein wird.

Zur Untersuchung der Steifheit und Schrittweitenbedingung nimmt man die Matrix A des SysgDGI (7.47). Ihre beiden negativen reellen Eigenwerte sind

$$\lambda_{1,2} = -50 \pm 7\sqrt{51} = -0.010\,001\,000\,200\,05, -99.989\,998\,999\,799\,95.$$

Ihre Beträge sind sehr verschieden und die Schrittweitenbedingung zum PZV ergibt sich aus dem betragsgrößten Wert und ist

$$h < \frac{2}{|\lambda_2|} = 0.020\,002\,000\,400\,100\,02, \text{ also grob } h \leq 0.02.$$

Analog ist die Lösung des SysgDGI (7.47) allgemein und mit den gegebenen AB.

```
> Digits:=16:
> sys3:=diff(x(t),t)=y(t),diff(y(t),t)=-x(t)-y(t)*100:
  fcns:=x(t),y(t):
  init:=x(0)=0,y(0)=1:
  loe3:=dsolve({sys3},{fcns});
  evalf(loe3);

> loe3a:=dsolve({sys3,init},{fcns});
  evalf(loe3a);

> # Loesungsfunktionen
xx:=unapply(rhs(op(1,loe3a)),t): # Reihenfolge x,y beachten
evalf(xx(t)):
yy:=unapply(rhs(op(2,loe3a)),t):
evalf(yy(t)):
```

$$\begin{aligned} \text{loe3} &:= \{x(t) = {}_C1 e^{((-50 + 7\sqrt{51})t)} + {}_C2 e^{-(50 + 7\sqrt{51})t}, \\ y(t) &= {}_C1 (-50 + 7\sqrt{51}) e^{((-50 + 7\sqrt{51})t)} + {}_C2 (-50 - 7\sqrt{51}) e^{-(50 + 7\sqrt{51})t}\} \\ \{x(t) &= {}_C1 e^{(-0.01000100020005t)} + {}_C2 e^{(-99.98999899979995t)}, \\ y(t) &= -0.01000100020005 {}_C1 e^{(-0.01000100020005t)} - \\ &\quad 99.98999899979995 {}_C2 e^{(-99.98999899979995t)}\} \\ \text{loe3a} &:= \{x(t) = \frac{1}{714}\sqrt{51} e^{((-50 + 7\sqrt{51})t)} - \frac{1}{714}\sqrt{51} e^{-(50 + 7\sqrt{51})t}, \\ y(t) &= \frac{1}{714}\sqrt{51} (-50 + 7\sqrt{51}) e^{((-50 + 7\sqrt{51})t)} - \frac{1}{714}\sqrt{51} (-50 - 7\sqrt{51}) e^{-(50 + 7\sqrt{51})t}\} \\ \{x(t) &= 0.01000200060020007 (e^{(-0.01000100020005t)} - e^{(-99.98999899979995t)}), \\ y(t) &= -0.0001000300100035011 e^{(-0.01000100020005t)} + \\ &\quad 1.000100030010004 e^{(-99.98999899979995t)}\} \end{aligned}$$

Wir berechnen das Richtungsfeld und dazu Phasenkurven des AWP mittels DEplot.

```
> DEplot({sys3}, [fcns], t=0..20, x=0..0.01, y=-0.1..1, [[x(0)=0, y(0)=1]],
         stepsize=0.01, dirgrid=[20,20], linecolor=black, labels=['x', 'x']);
```

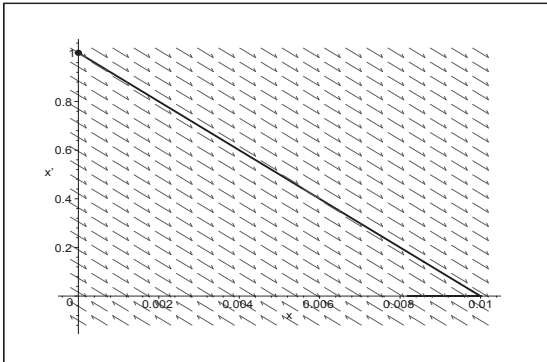


Abb. 7.94

Datei *nsteif62.ps*,
AWP (7.47),
Richtungsfeld (x, x') und
Phasenkurve $(x(t), x'(t))$, $t \in [0, 20]$,
Kommando DEplot

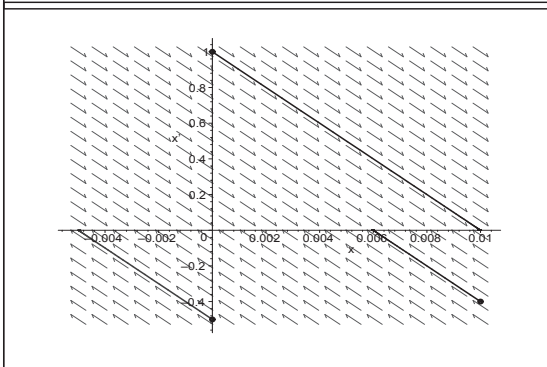


Abb. 7.95

Datei *nsteif63.ps*,
AWP (7.47) mit AB $(x(0), y(0)) =$
 $(0, 1), (0, -0.5), (0.01, -0.4)$,
Richtungsfeld (x, x') und
Phasenkurven $(x(t), x'(t))$, $t \in [0, 1]$,
Kommando DEplot

Man sieht, dass die Linienelemente nahe der x -Achse von beiden Seiten direkt aufeinander zugehen. Am “Knick“ der Phasenkurve ist es natürlich schwierig, ein solches Element als Tangente anzubringen.

Deshalb versuchen wir mittels `dfieldplot` diese Schwierigkeiten in der Grafik zumindest noch andeutungsweise zu beheben, indem wir uns immer mehr in die Problemzone hineinbegeben. Zum Schluss zeichnen wir zu den Linienelementen ein kleines Stück der Phasenkurve.

```
> opt1:=arrows=LARGE, labels=['x', 'x']:
> dfieldplot([sys3], [fcns], t=0..20, x=0..0.01, y=-0.1..1.0,
            dirgrid=[21,23], labels=['x', 'x']):
> dfieldplot([sys3], [fcns], t=0.06..20, x=0.008..0.010, y=-0.0005..0.0020,
            dirgrid=[21,16], opt1):
> dfieldplot([sys3], [fcns], t=0.06..20, x=0.008..0.011, y=-0.0002..0.00035,
            dirgrid=[31,23], opt1):
> dfieldplot([sys3], [fcns], t=0.06..20, x=0.0096..0.0102, y=-0.00012..0.00004,
            dirgrid=[31,17], opt1):
```

```

> dfieldplot([sys3],[fcns],t=0.06..20,x=0.0098..0.0102,
             y=-0.00011..-0.00008,dirgrid=[31,19],opt1):
> p1e:=dfieldplot([sys3],[fcns],t=0.06..20,x=0.0098..0.0102,
                 y=-0.000101..-0.000097,dirgrid=[37,21],opt1):
p2e:=plot([xx(t),yy(t),t=0.128..2],color=black,thickness=4,
          view=[0.0098..0.0102,-0.000101..-0.000097]):
display(p1e,p2e);

```

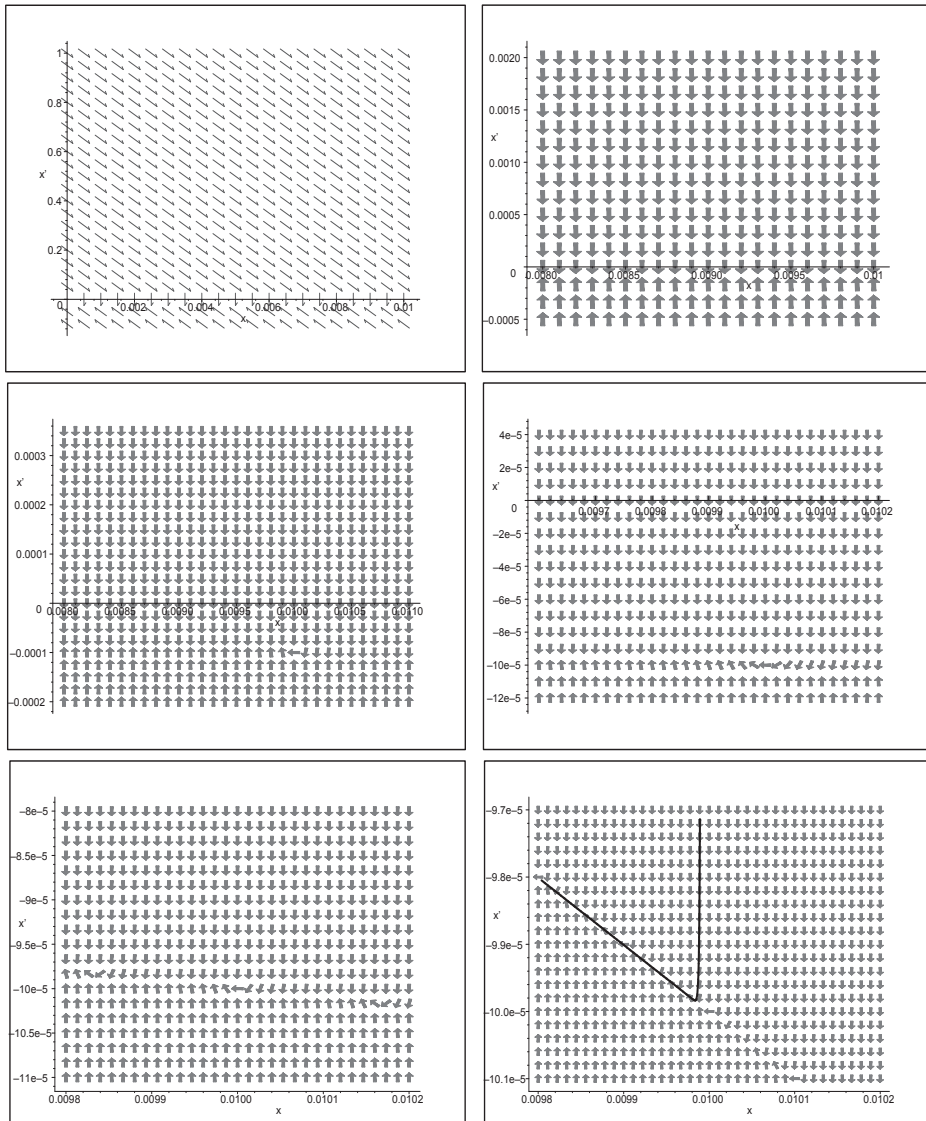


Abb. 7.96 Dateien *nsteif53.ps*, *nsteif53a.e.ps*, AWP (7.47), Richtungsfelder in ausgewählten Bereichen mit `dfieldplot`

Nimmt man das Verhalten der Lösungskomponenten $x(t)$ und $y(t)$ vom AWP (7.47) formal voneinander unabhängig, so kann man zwei typische Phasen (Abschnitte) ihrer Entwicklung beobachten. Dort, wo in kurzer Zeit viel passiert, das ist die transiente Phase. Ansonsten spricht man von der asymptotischen Phase.

Da die Funktion $x(t)$ im sehr kurzen Zeitintervall $[0, t_M] = [0, 0.092\ 119]$ ihren größten Anstieg hat und die Funktion $y(t)$ dort schnell von Eins auf Null fällt, kann man die ersten Intervallhälfte $[0, 0.05]$ als transiente Phase deklarieren.

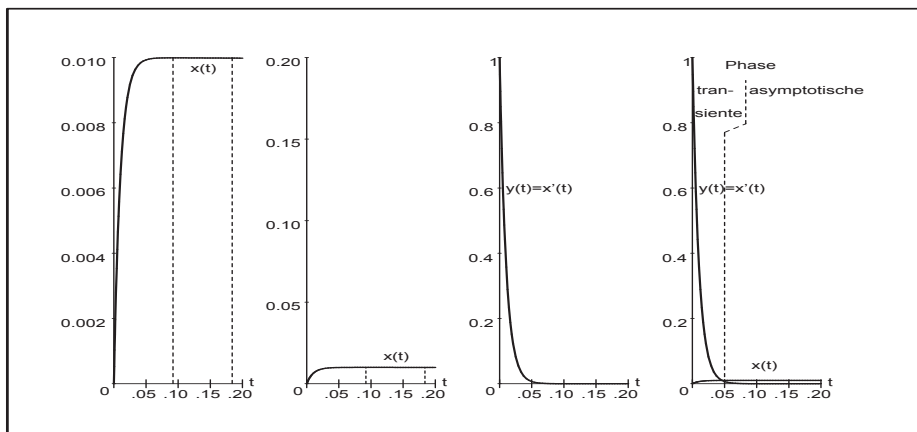


Abb. 7.97 Datei *nsteif54.ps*, AWP (7.47), Lösungen $x(t)$, $y(t)$ in $[0, 0.2]$ mit der Charakterisierung ihrer transienten und asymptotischen Phase

In solchen Fällen extremer Unterschiede im Verhalten der Lösungskomponenten eines SysgDGl wird ihre gemeinsame grafische Darstellung immer ein Problem sein, was die folgenden Figuren noch einmal unterstreichen sollen.

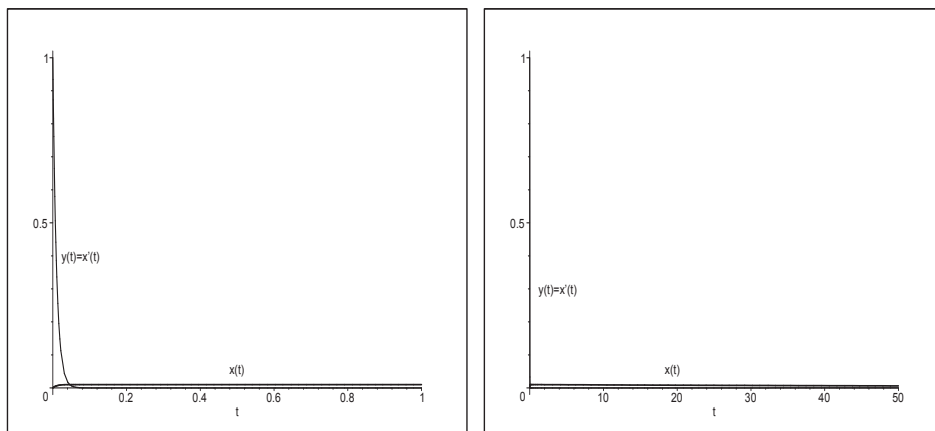


Abb. 7.98 Dateien *nsteif55..56.ps*, AWP (7.47), Lösungsverhalten von $x(t)$, $y(t)$ in $[0, 1]$ bzw. $[0, 50]$

Abschließend wollen wir sowohl das PZV als auch das KRKV im Intervall $[0, T]$ mit fester Schrittweite $h = T/N$ testen.

Als Schrittweitenbedingung zur Stabilität des PZV haben wir $h \leq 0.02$ erhalten. Das sind bei einem Integrationsintervall $[0, T] = [0, 1]$ also $N = 50$ Schritte. Es ist zu erwarten, dass für $h \lesssim 0.02$ die Näherungslösung zwar stabil ist. Aber sie wird sich oszillierend um die exakte Lösung dieser nur sehr langsam annähern. Erst bei $h \leq 0.01$ können monotone Eigenschaften und eine gute Genauigkeit erwartet werden.

Das KRKV besitzt ein etwas größeres Stabilitätsintervall (vergl. Abschnitt 7.1). Anstelle der Bedingung $h < 2/|\lambda|_{max}$ haben wir $h < 2.785293/|\lambda|_{max} = 0.027855$, also bleiben wir bei $T = 1$ und $N = 36$ gerade noch mit $h = 0.027778$ im Bereich der absoluten Stabilität. Mit $N = 35$ ist die Schrittweite h schon zu groß.

Dazu erfolgen einige Rechnungen und Grafiken mit MATLAB unter Verwendung von Prozeduren für die ESV aus den Abschnitten 4.2 und 5.2.

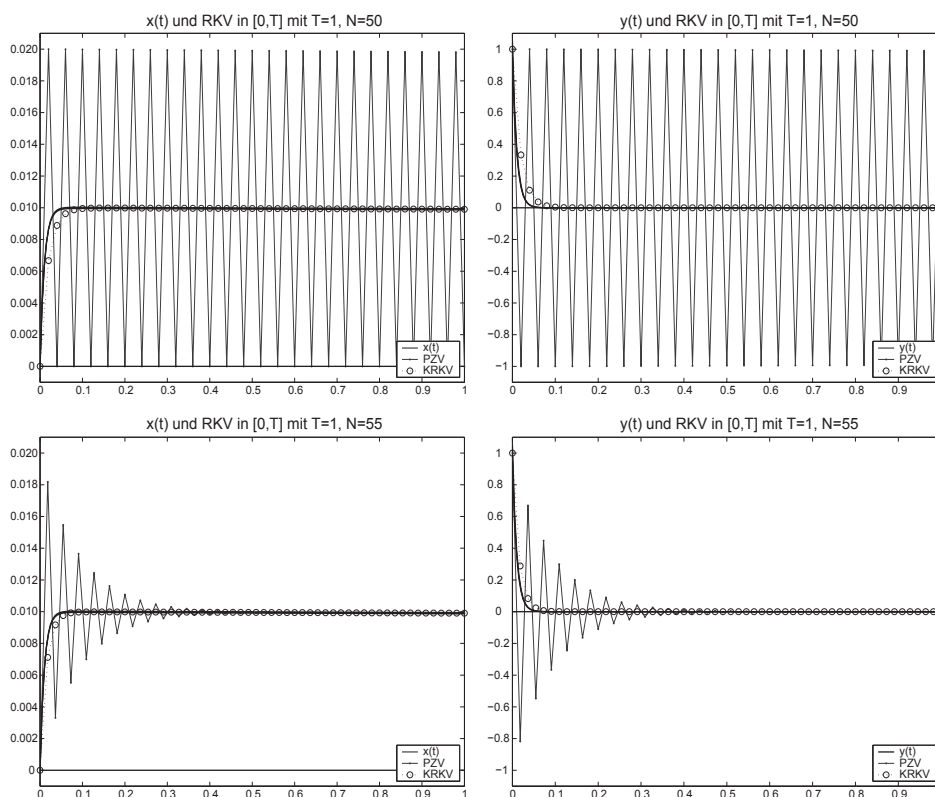


Abb. 7.99 Dateien *steif_10..13.ps*, AWP (7.47),

Näherungslösungen η_i , $i = 0, 1, \dots, N$, aus PZV und KRKV in $[0, 1]$
bei $N = 50, 55$ im Vergleich mit Lösungen $x(t)$, $y(t)$

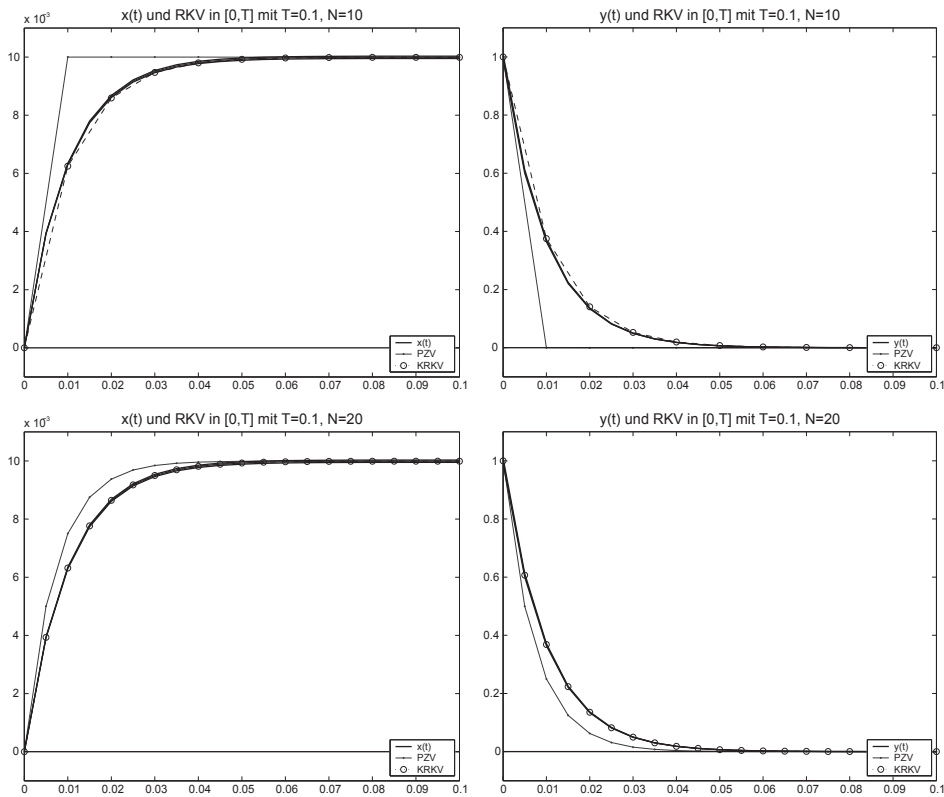


Abb. 7.100 Dateien *stief_14..17.ps*, AWP (7.47),
 Näherungslösungen η_i , $i = 0, 1, \dots, N$, aus PZV und KRKV in $[0, 0.1]$
 bei $N = 10, 20$ im Vergleich mit den Lösungen $x(t), y(t)$

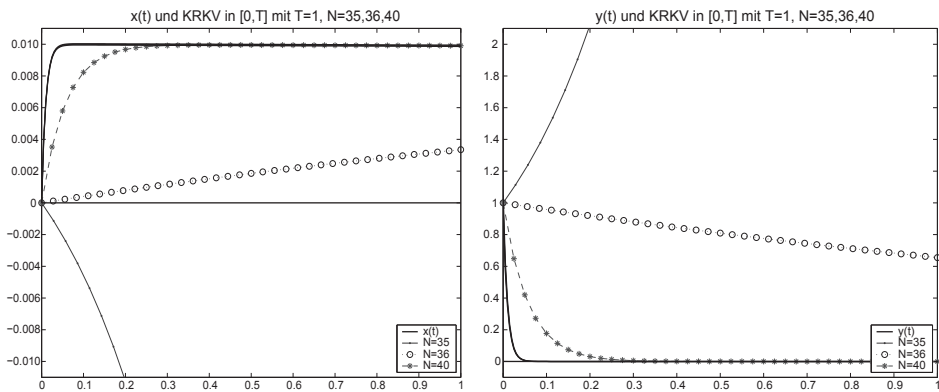


Abb. 7.101 Dateien *stief_18..19.ps*, AWP (7.47),
 Näherungslösungen η_i , $i = 0, 1, \dots, N$, aus KRKV in $[0, 1]$
 bei $N = 35, 36, 40$ im Vergleich mit den Lösungen $x(t), y(t)$

(b) Wir betrachten nun ein lineares SysgDGL mit mehr Steifheit als das vorherige. Das AWP für $\mathbf{x} = (x, y)^T$ aus [97] lautet

$$\mathbf{x}'(t) = A \mathbf{x}(t), \quad \mathbf{x}(0) = (1, 1)^T, \quad t \geq 0, \quad \text{wobei } A = \begin{pmatrix} -29\,998 & -39\,996 \\ 29\,997/2 & 19\,997 \end{pmatrix}. \quad (7.48)$$

Die allgemeine Lösung ist

$$x(t) = C_1 e^{-t} + C_2 e^{-10000t}, \quad y(t) = -\frac{3}{4}C_1 e^{-t} - \frac{1}{2}C_2 e^{-10000t},$$

die des AWP mit den gegebenen AB

$$x(t) = -6e^{-t} + 7e^{-10000t}, \quad y(t) = \frac{9}{2}e^{-t} - \frac{7}{2}e^{-10000t}.$$

Die Eigenwerte der Systemmatrix A sind $\lambda_{1,2} = -1, -10\,000$, was auch auf ihre schlechte Kondition hinweist.

Damit haben wir zum PZV die Schrittweitenbedingung $h < 2/|\lambda|_{max} = 0.0002$.

Der Eigenwert $-10\,000$ und damit die Funktion $e^{-10000t}$ verursachen die scharfe Schrittweitenbeschränkung. Auch für andere explizite ESV ist die Bedingung ähnlich.

Die Lösungsanteile e^{-t} und $e^{-10000t}$ verhalten sich für kleine $t > 0$ deutlich anders. Insbesondere tendiert $e^{-10000t}$ für wachsendes t sehr schnell gegen Null. Bei $t = 0.0005$ haben wir schon den Wert 0.006737 . Somit könnte die Stelle als Übergang zwischen der transienten und asymptotischen Phase der Lösungen $x(t), y(t)$ angenommen werden.

Leider kann man nicht generell bei einer näherungsweisen Lösung eines AWP die Schrittweitenbedingung durch "Ausblenden des restriktiven Anteils" entschärfen, weil man nicht die "zuständigen" Stellen im SysgDGL kennt.

Nur bei $A(\alpha)$ -Stabilität der Verfahrens und Schrittweitensteuerung bemerkt man, dass in Abhängigkeit von einer vorgegebenen Fehlertoleranz eine kleine Anfangsschrittweite nach Verlassen der transienten Lösungsphase sich stetig vergrößern wird.

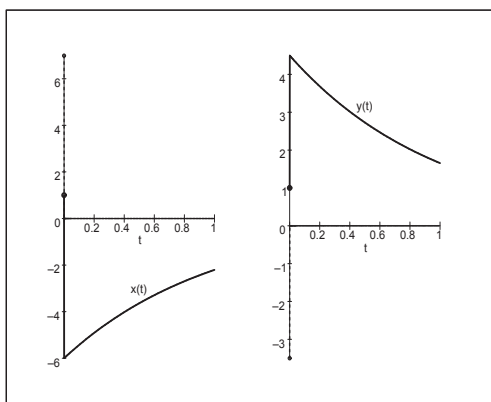


Abb. 7.102

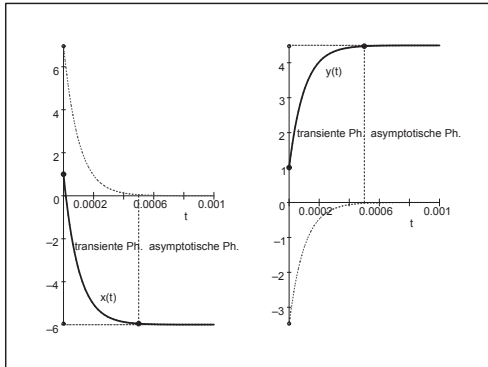
Datei *nsteif64.ps*,

AWP (7.48) mit AB $(x(0), y(0)) = (1, 1)$,
Lösungen $x(t), y(t)$ in $[0, 1]$,

dazu die Lösungsanteile

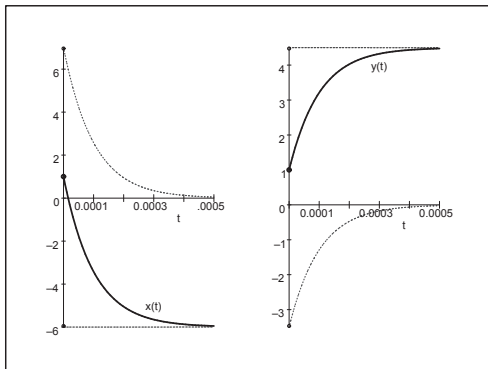
$-6e^{-t}$ (von $x(t)$ "zugedeckt"), $7e^{-10000t}$,
 $\frac{9}{2}e^{-t}$ (von $y(t)$ "zugedeckt"), $-\frac{7}{2}e^{-10000t}$

Einige Lösungsanteile zu $x(t), y(t)$ lassen sich nur bei besserer grafischer Auflösung erkennen.

**Abb. 7.103**Datei *nsteif65.ps*,AWP (7.48) mit AB $(x(0), y(0)) = (1, 1)$,Lösungen $x(t)$, $y(t)$ in $[0, 0.001]$,

dazu die Lösungsanteile

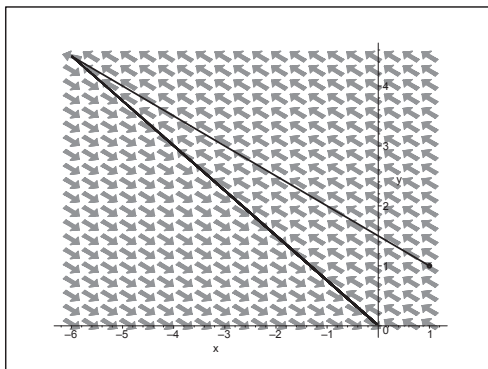
$$-6e^{-t}, 7e^{-10000t} \text{ bzw. } \frac{9}{2}e^{-t}, -\frac{7}{2}e^{-10000t}$$

**Abb. 7.104**Datei *nsteif66.ps*,AWP (7.48) mit AB $(x(0), y(0)) = (1, 1)$,Lösungen $x(t)$, $y(t)$ in $[0, 0.0005]$,

dazu die Lösungsanteile

$$-6e^{-t}, 7e^{-10000t} \text{ bzw. } \frac{9}{2}e^{-t}, -\frac{7}{2}e^{-10000t}$$

Das Zeichnen des Richtungsfeldes mit dem Kommando `DEplot` sowie der Näherungskurve zum AWP (7.48) verlangt bei ESV wie `method=classical[rk4]` eine hinreichend kleine Schrittweite h . Mit `stepsize=0.0002` liegen wir im zulässigen Bereich, bei `stepsize=0.0003` folgt schon eine Fehlermeldung.

**Abb. 7.105**Datei *nsteif67.ps*,

AWP (7.48),

Richtungsfeld (x, y) undPhasenkurve $(x(t), y(t))$, $t \in [0, 5]$ mit AB $(x(0), y(0)) = (1, 1)$,Kommando `DEplot`

Wir haben dieselbe Situation der Linienelemente und den ‘‘Knickeffekt‘‘ wie im Beispiel zuvor (vergl. Abb. 7.94), wollen uns jedoch nicht weiter in diesen Bereich hineinzoomen.

Zu ausgewählten Näherungsverfahren machen wir Rechnungen.

Einerseits nehmen wir Verfahren wie BDF(n) (vergl. Tab. 6.7) mit fester Schrittweite und BDF(5,4) mit variabler SWS, zum anderen testen wir einige MATLAB-Funktionen zur Lösung von SysgDGL.

1. `ode23`, `ode45`
Explizite RKV zur Lösung nicht steifer gDGL,
2. `ode113`
AM-AB-Prädiktor-Korrektor-Verfahren zur Lösung nicht steifer gDGL,
3. `ode23t` (trapezoidal rule)
BDF-Methoden für moderat steife gDGL,
4. `ode23s`, `ode23tb` (low order methods), `ode15s` (variable order method)
Implizite RKV, BDF-Methoden, NDF-Methoden (Numerical Differentiation Formulas), Rosenbrock-Verfahren für steife gDGL.

Die MATLAB-Versionen arbeiten mit ausgefeilter Schrittweiten- und Ordnungssteuerung.

Wir belassen es bei ihrer standardmäßigen absoluten Fehlertoleranz `AbsTol=1e-6`.

Für `ode15s` rechnen wir alle Varianten der Genauigkeitsordnung `MaxOrder=1,2,3,4,5`.

Als Ausgabe haben wir die Integrationspunkte t_i (Achtung: in MATLAB ist t_1 der linke Intervallgrenze), ihre Anzahl $N1 = N + 1$, die Näherungslösung η_{1i} , η_{2i} sowie die benutzten Schrittweiten am Anfang und Ende (vorletzte Schrittweite) des Lösungsintervalls $[0, T]$, bei $N/2$ und an der Phasengrenze $t = 0.0005$. Da die exakte Lösung $x(t)$, $y(t)$ zur Verfügung steht, ermitteln wir an den Integrationspunkten die Größen

$$aerr_x = \max_{i=0,1,\dots,N} |\eta_{1i} - x(t_i)|, \quad aerr_y = \max_{i=0,1,\dots,N} |\eta_{2i} - y(t_i)|$$

und die zugehörigen Stellen, wo diese Maxima angenommen werden.

Beispiel für die Implementierung in MATLAB

```
opt = odeset('AbsTol',1e-6,'MaxOrder',3) % Order 1,2,3,4,5
[t,x] = ode15s(@f5s,[0 1],[1 1],opt);
N1 = max(size(t)) % N+1
ha = t(2)-t(1) % Schrittweiten
Nm = (N1-mod(N1,2))/2;
hm = t(Nm)-t(Nm-1)
he = t(N1-1)-t(N1-2)
for i = 1:N1
    [xex(i),yex(i)] = f16exv(t(i)); % exakte Loesung
end;
[ae_x,ix] = max(abs(x(:,1))-xex(1:N1)) % Fehler und Stellen
[ae_y,iy] = max(abs(x(:,2))-yex(1:N1))
aerr_xN = abs(x(1,N1)-xex(N1))
aerr_yN = abs(x(2,N1)-yex(N1))
```

BDF(5,4)			ode23t		
Toleranz $\varepsilon = 10^{-4}$ Startschrittweite 10^{-8} Genauigkeitsordnung 5,4			Toleranz $\varepsilon = 10^{-6}$ Startschrittweite 10^{-6} Genauigkeitsordnung 2,3		
t -Werte	Schrittweite h	Schrittzahl $N1$	t -Werte	Schrittweite h	Schrittzahl $N1$
1.6e-4	4.0e-6	78	1.6e-4	1.9e-5	17
4.4e-4	5.7e-6	140	4.3e-4	4.6e-5	26
1.0e-3	1.6e-5	200	1.1e-3	1.7e-4	33
0.04	0.016	243	0.048	0.034	39
0.2	0.036	250	0.24	0.10	41
0.5	0.082	254	0.54	0.10	44
1	0.15	258	1	0.10	50
globaler Fehler $\mathcal{O}(10^{-6})$			globaler Fehler $\mathcal{O}(10^{-3})$		

Tab. 7.6 AWP (7.48) in $[0,1]$, Schrittweitenverhalten von BDF-Methoden mit variabler Schrittweiten- und Ordnungssteuerung

Methode	Schritt-zahl $N1$	Schrittweite h_i bei				$aerr_x(i)$	$aerr_y(i)$
		$i = 1$	$i \approx N1/2$	$t \approx 0.0005(i)$	$i = N1$		
ode23	3992	1.142e-6	2.512e-4	1.9e-4 (14)	2.512e-4	0.017 065 (14)	0.008 532 (14)
ode45	12085	7.177e-7	9.569e-5	3.5e-5 (30)	2.774e-5	0.006 242 (552)	0.003 121 (552)
ode113	6134	1.129e-7	1.945e-4	5.8e-5 (20)	2.481e-4	0.046 319 (3220)	0.023 159 (3220)
ode23t	50	1.182e-6	3.718e-5	6.0e-5 (28)	0.100	0.003 688 (21)	0.001 844 (21)
ode23s	33	4.276e-6	7.229e-5	9.1e-5 (16)	0.100	0.004 897 (12)	0.002 448 (12)
ode23tb	36	3.737e-6	5.048e-5	8.9e-5 (19)	0.100	0.003 527 (14)	0.001 763 (14)
ode15s	48	1.352e-6	5.013e-5	5.0e-5 (22)	0.100	0.005 701 (11)	0.003 979 (39)
Ord.=5	o. V.						
Ord.=4	o. V.						
Ord.=3	o. V.						
Ord.=2	55	1.352e-6	3.287e-5	4.9e-5 (30)	0.100	0.007 655 (46)	0.005 741 (46)
Ord.=1	123	1.352e-6	6.330e-6	4.5e-5 (87)	0.047	0.031 405 (121)	0.023 553 (121)

Tab. 7.7 AWP (7.48) in $[0,1]$, MATLAB-Funktionen zur numerischen Lösung mit verschiedenen Ergebnisgrößen (o. V. = ohne Veränderung zu ode15s)

Die expliziten ESV und MSV brauchen für ein steifes Problem wesentlich mehr Schritte, denn sie müssen die Schrittweitenbedingung über das ganze Integrationsintervall einhalten. So können sich ihre Schrittweiten im Laufe der Integration nicht extrem vergrößern, auch nicht in der asymptotischen Phase $[0.0005, 1]$.

Die anderen Methoden für steife Probleme haben ähnliches Lösungsverhalten und Fehlergrößen. Mit ihren verbesserten Stabilitätseigenschaften oder A-Stabilität ist die Fehlertoleranz für die Wahl von h wichtig. Im transienten Integrationsbereich $[0, 0.0005]$ muss zwecks guter Genauigkeit h trotzdem hinreichend klein sein. Dort treten auch die maximalen Fehler $aerr_x, aerr_y$ auf. I. Allg. ist es schwierig einzuschätzen, welche Methode für welche gDGl sich genau als die beste erweist.

Wir vergleichen die ersten BDF(n)-Verfahren der Genauigkeitsordnung $\mathcal{O}(h^n)$ bei Rechnung mit fester Schrittweite h .

Da das AWP linear ist, können die r -stufigen impliziten Formeln (6.42)

$$\alpha_r \eta_{i+1} + \alpha_{r-1} \eta_i + \dots + \alpha_0 \eta_{i-r+1} = h \beta_r f_{i+1}, \quad i=r-1, r, \dots, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ geg.}$$

nach η_{i+1} aufgelöst werden. Man erhält

$$\begin{aligned} (\alpha_r I - h \beta_r A) \eta_{i+1} &= -\alpha_{r-1} \eta_i - \dots - \alpha_0 \eta_{i-r+1} \\ \eta_{i+1} &= M(-\alpha_{r-1} \eta_i - \dots - \alpha_0 \eta_{i-r+1}), \quad M = (\alpha_r I - h \beta_r A)^{-1}. \end{aligned}$$

Die einmalige Berechnung von M nehmen wir hier in Kauf.

Wir rechnen also BDF(1,2,3) bei $i = r-1, r, \dots, N-1$, $r = 1, 2, 3$, gemäß

$$\begin{aligned} \eta_{i+1} &= M_1 \eta_i, \quad M_1 = (I - hA)^{-1}, \quad \eta_0 \text{ gegeben,} \\ \eta_{i+1} &= M_2(4\eta_i - \eta_{i-1}), \quad M_2 = (3I - 2hA)^{-1}, \quad \eta_0, \eta_1 \text{ geg.,} \\ \eta_{i+1} &= M_3(18\eta_i - 9\eta_{i-1} + 2\eta_{i-2}), \quad M_3 = (11I - 6hA)^{-1}, \quad \eta_0, \eta_1, \eta_2 \text{ geg.,} \end{aligned}$$

mit den zugehörigen Startvektoren $\eta_0 = (x(0), y(0))^T$ (AB), $\eta_1 = (x(h), y(h))^T$, $\eta_2 = (x(2h), y(2h))^T$, da die exakten Lösungen vorliegen.

Das Ergebnis ist eine Tabelle mit den maximalen absoluten Fehlern $amax_x$, $amax_y$ sowie den Fehlern

$$aerr_{xN} = |\eta_{1N} - x(t_N)|, \quad aerr_{yN} = |\eta_{2N} - y(t_N)|.$$

Dazu notieren wir die Stellen (Index) $i \in \{0, 1, \dots, N\}$, wo $amax_x$, $amax_y$ auftreten.

N	h	BDF(1)			
		$aerr_x(i)$	$aerr_y(i)$	$aerr_{xN}$	$aerr_{yN}$
10	0.1	0.105 983 089 (10)	0.079 487 317 (10)	0.105 983 089	0.079 487 317
100	0.01	0.069 011 873 (1)	0.034 432 172 (1)	0.010 990 626	0.008 242 970
1 000	0.001	0.636 042 841 (1)	0.318 020 672 (1)	0.001 103 178	0.000 827 384
2 000	0.000 5	1.119 500 288 (1)	0.559 749 956 (1)	0.000 551 704	0.000 413 778
4 000	0.000 25	1.425 404 822 (1)	0.712 702 364 (1)	0.000 275 880	0.000 206 910
5 000	0.000 2	1.385 986 230 (1)	0.692 993 085 (1)	0.000 220 709	0.000 165 531
10 000	0.000 1	0.924 843 881 (1)	0.462 421 933 (1)	0.000 110 359	0.000 082 769
20 000	0.000 05	0.535 955 007 (2)	0.267 977 500 (2)	0.000 055 180	0.000 041 385
100 000	0.000 01	0.123 646 934 (10)	0.061 823 466 (10)	0.000 011 036	0.000 008 277
200 000	0.000 005	0.063 070 290 (20)	0.031 535 144 (20)	0.000 005 518	0.000 004 138

Tab. 7.8 AWP (7.48) in $[0,1]$, Fehler zu BDF(1) $\eta_{i+1} = M_1 \eta_i$ bei fester Schrittweite h

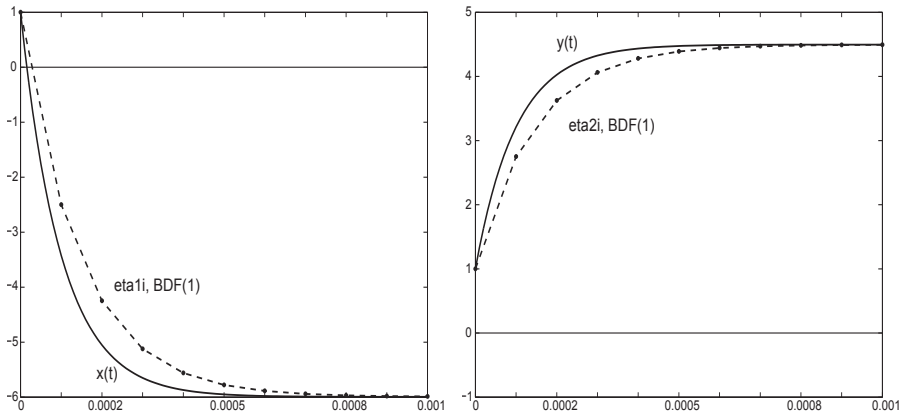


Abb. 7.106 Dateien *bdf1_3, 4.ps*, AWP (7.48) in $[0, 1]$, BDF(1) mit $h = 0.0001$, Vergleich mit exakten Lösungen $x(t)$ bzw. $y(t)$ im Bereich $t \in [0, 0.001]$

		BDF(2)			
N	h	$aerr_x(i)$	$aerr_y(i)$	$aerr_{xN}$	$aerr_{yN}$
10	0.1	0.006 716 697 (10)	0.005 037 523 (10)	0.006 716 697	0.005 037 523
100	0.01	0.034 481 450 (2)	0.017 240 398 (2)	0.000 073 017	0.000 054 762
1 000	0.001	0.304 292 569 (2)	0.152 146 284 (2)	0.000 000 735	0.000 000 551
2 000	0.0005	0.524 266 836 (2)	0.262 133 418 (3)	0.000 000 183	0.000 000 137
4 000	0.00025	0.634 868 133 (2)	0.317 434 066 (2)	0.000 000 045	0.000 000 034
5 000	0.0002	0.586 868 339 (2)	0.293 434 169 (2)	0.000 000 029	0.000 000 022
10 000	0.0001	0.335 440 799 (3)	0.167 720 399 (3)	0.000 000 007 3	0.000 000 005 5
20 000	0.00005	0.136 989 673 (4)	0.068 494 836 (4)	0.000 000 001 8	0.000 000 001 3
100 000	0.00001	0.007 943 518 (12)	0.003 971 759 (12)	0.000 000 000 062	0.000 000 000 046
200 000	0.000005	0.002 065 770 (22)	0.001 032 885 (22)	0.000 000 000 008	0.000 000 000 003

Tab. 7.9 AWP (7.48) in $[0, 1]$, Fehler zu BDF(2) $\eta_{i+1} = M_2(4\eta_i - \eta_{i-1})$ bei festem h

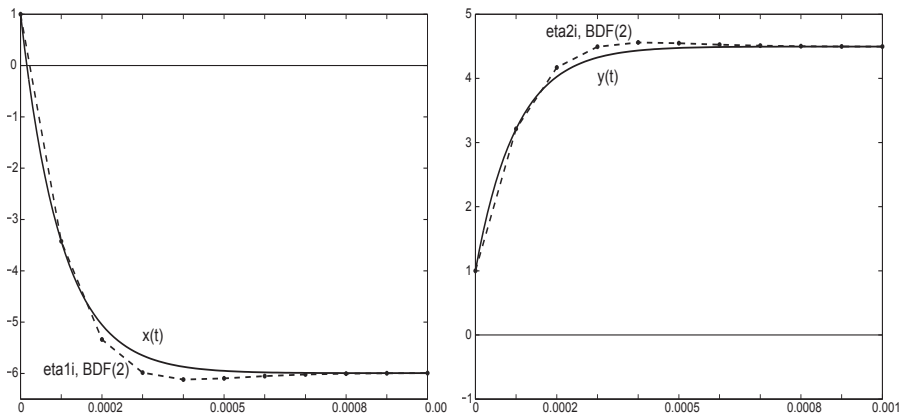


Abb. 7.107 Dateien *bdf2_3, 4.ps*, AWP (7.48) in $[0, 1]$, BDF(2) mit $h = 0.0001$, Vergleich mit exakten Lösungen $x(t)$ bzw. $y(t)$ im Bereich $t \in [0, 0.001]$

N	h	BDF(3)			
		$aerr_x(i)$	$aerr_y(i)$	$aerr_{xN}$	$aerr_{yN}$
10	0.1	0.002 264 184 (3)	0.001 115 872 (3)	0.000 467 926	0.000 350 944
100	0.01	0.022 913 248 (3)	0.011 456 622 (3)	0.000 000 544	0.000 000 408
1 000	0.001	0.197 142 817 (3)	0.098 571 408 (3)	0.000 000 000 55	0.000 000 000 41
2 000	0.000 5	0.331 247 364 (3)	0.165 623 682 (3)	0.000 000 000 055	0.000 000 000 041
4 000	0.000 25	0.368 344 809 (3)	0.184 172 404 (3)	0.000 000 000 009 2	0.000 000 000 006 9
5 000	0.000 2	0.320 981 241 (3)	0.160 490 620 (3)	0.000 000 000 019	0.000 000 000 014
10 000	0.000 1	0.163 750 232 (4)	0.081 875 116 (4)	0.000 000 000 019	0.000 000 000 014
20 000	0.000 05	0.042 329 278 (5)	0.021 164 639 (5)	0.000 000 000 041	0.000 000 000 030
100 000	0.000 01	0.000 565 717 (12)	0.000 282 858 (12)	0.000 000 000 11	0.000 000 000 084
200 000	0.000 005	0.000 075 413 (23)	0.000 037 706 (23)	0.000 000 000 000 79	0.000 000 000 000 59

Tab. 7.10 AWP (7.48) in $[0,1]$, Fehler zu BDF(3) $\eta_{i+1} = M_3(18\eta_i - 9\eta_{i-1} + 2\eta_{i-2})$ bei festem h

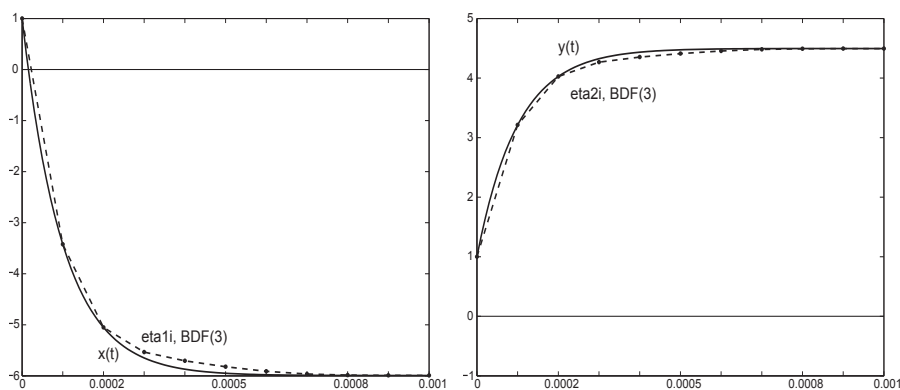


Abb. 7.108 Dateien *bdf3_3, 4.ps*, AWP (7.48) in $[0, 1]$, BDF(3) mit $h = 0.0001$, Vergleich mit exakten Lösungen $x(t)$ bzw. $y(t)$ im Bereich $t \in [0, 0.001]$

Was kann man beobachten?

Bei konstanter Schrittweite h haben die Näherungslösungen anfangs, also in der “gefährlichen“ transienten Phase, ihre Genauigkeitsprobleme. Dort tritt der maximale Fehler auf. Aber mit wachsendem t werden sie immer genauer, sodass am Ende der zweiten Phase, egal ob man nun $T = 0.2, 0.5, 1, 2, \dots$ nimmt, nur noch sehr kleine Abweichungen zu den exakten Lösungen auftreten.

Mit Verkleinerung von h , d.h. Vergrößerung der Schrittzahl N , wächst i. Allg. die Genauigkeit. Ausnahmen bilden jedoch solche Situationen, wo durch eine zu grobe Schrittweite $h > 0.0005$ überhaupt keine Stellen in der transienten Phase $[0, 0.0005]$ erfasst werden, also darüber hinweg integriert wird. Wenn man aber mit $h < 0.0005$ dann Stellen in diesem Bereich hat, so sollte man mit $h \ll 0.0005$ rechnen, um zumindest die Genauigkeit $h^{1..2}$ zu retten. Ein typisches Beispiel in allen drei Tabellen ist die Schrittweite $h = 0.00025$, wo der größte maximale Fehler auftritt. Die Steifheit macht dabei die Genauigkeit h^n des BDF(n) zunichte.

Die Berechnung der (2, 2)-Übergangsmatrix

$$M = (\alpha_r I - h\beta_r A)^{-1}$$

sollte eigentlich kein besonderes Problem darstellen.

Aber wir haben eingangs schon auf die schlechte Kondition der Systemmatrix A hingewiesen. Sie beträgt $\text{cond}(A) \approx 312\,439$. Die Kondition von M verbessert sich natürlich mit kleiner werdendem h . Für $M_1(h) = I - hA$ ist sie z. B. bei $h_s = 10^{-s}$, $s = 1, 2, 3, 4$, entsprechend $\text{cond}(M_1(h_s)) \approx 28\,377, 3\,064, 285, 17$.

Erstaunlich ist, wie schwer sich MATLAB mit der Berechnung von A^{-1} bzw. M tut. Betrachten wir die Ergebnisse der folgenden Anweisungen.

```
% Test zur Systemmatrix
A = [-29998 -39996; 29997/2 19997]
condA = cond(A)
invA = inv(A)
A*invA
det(A)
detA = A(1,1)*A(2,2)-A(2,1)*A(1,2)

invAs = [A(2,2) -A(1,2); -A(2,1) A(1,1)]/detA
A*invAs
invAs1 = 1/detA*[A(2,2) -A(1,2); -A(2,1) A(1,1)]
A*invAs1

% -----
% Ergebnisse
A      = -29998.0000000000    -39996.0000000000
         14998.5000000000     19997.0000000000

condA =  3.124395031180386e+005

invA   =  1.99970000001133    3.99960000002266
         -1.49985000000850    -2.99980000001699

ans    =  0.99999999999272          0
         0.00000000000728    1.00000000000728

ans    =  9.99999999943349e+003
detA   =  10000

invAs  =  1.99970000000000    3.99960000000000
         -1.49985000000000    -2.99980000000000

ans    =  1      0
         0      1

invAs1 =  1.99970000000000    3.99960000000000
         -1.49985000000000    -2.99980000000000

ans    =  1.00000000000728          0
         -0.00000000000364    1.00000000000000
```

Es ist unter Berücksichtigung der schlechten Kondition von A schon verwunderlich, dass die Befehle `inv` und `det` so unbefriedigende Ergebnisse erzielen. Im Gegensatz dazu liefert die einfache Cramersche Regel in der gegebenen GKA *double* sehr ordentliche Resultate.

Dazu kommt noch der Unterschied, ob man bei der Kontrolle der Übereinstimmung von AA^{-1} mit der Einheitsmatrix die Inverse gemäß $[A(2,2), -A(1,2); -A(2,1), A(1,1)] / \det A$ oder $1/\det A * [A(2,2), -A(1,2); -A(2,1), A(1,1)]$ bildet. An den Inversen ist der Unterschied noch nicht zu erkennen, aber die Produkte AA^{-1} sind verschieden.

Denselben Test kann man mit $M_1(h)$ durchführen. Dabei stellt man Ähnliches fest für $h = 0.1, 0.01, 0.001$, nicht mehr bei $h = 0.0001$. Deshalb haben wir bei den Rechnungen mit BDF(1,2,3) die Schrittweite $h = 0.0001$ genommen.

Mit dem BDF(2) erzielen wir eine gute Genauigkeit. Es soll nun gemeinsam mit dem BDF(1) zu einem Verfahren mit einer dualen SWS (Halbierung/ohne Veränderung/Verdopplung) verwendet werden (vergl. Abschnitt 5.4). Wir nehmen die beiden einfachsten BDF-Methoden, weil der Algorithmus der Steuerung bei nur drei aufeinander folgenden Werten $\eta_{i-1}, \eta_i, \eta_{i+1}$ mit $h_i = t_{i+1} - t_i$ gut überschaubar ist.

Die Anfangsschrittweite $h_0 = t_1 - t_0$ und die AB η_0, η_1 liegen vor.

Wir haben dann mit BDF(1,2) zwei Näherungen

$$\tilde{\eta}_{i+1} = M_1\eta_i, \quad \eta_{i+1} = M_2(4\eta_i - \eta_{i-1}), \quad i \geq 1.$$

Wir testen die Fehlergröße $est = |\eta_{i+1} - \tilde{\eta}_{i+1}|$. Liegt sie unterhalb einer vorgegebenen Toleranz ε , wird der Wert η_{i+1} als Näherungslösung akzeptiert. Die Schrittweite bleibt wie bisher, wenn der Fehler est nicht zu klein ausfällt. Bei entsprechend kleinem Fehler wird die Schrittweite verdoppelt und mit dem Tripel $\eta_{i-1}, \eta_{i+1}, 2h_i$ weiter gearbeitet.

Ist jedoch est zu groß, dann halbiert man die Schrittweite, berechnet mit dieser die Zwischengröße $\hat{\eta}_{i-1/2} = M_1\eta_{i-1}$ gemäß BDF(1) und macht mit dem Tripel $\hat{\eta}_{i-1/2}, \eta_i, \frac{h_i}{2}$ einen neuen Versuch zur Berechnung von $\eta_{i+1/2}$. Wir legen hier keinen Wert auf eine ausgefeilte Strategie und werden sie auch nicht in alle Richtungen austesten. Es können jedoch Probleme wegen fester Vorgabe von h_0 und der Zwischengröße $\hat{\eta}_{i-1/2}$ auftreten.

Als untere Toleranzgrenze ist wegen der Genauigkeitsordnung 1 des BDF(1) der Wert $\frac{1}{4}\varepsilon$ gewählt, gegebenenfalls ist er etwas kleiner zu nehmen, denn der Toleranzbereich $[\frac{1}{4}\varepsilon, \varepsilon]$ sollte die Gefahr einer ständigen Halbierung/Verdopplung weitgehend ausschließen. Zur gegebenen Toleranz ε wird die Anfangsschrittweite gerade so gewählt, dass sie möglichst groß ist und für einige anfängliche Schritte akzeptiert wird.

Als Näherungslösung werden die Werte η_{i+1} von BDF(2) gespeichert. Der maximale Fehler hat die Ordnung $\mathcal{O}(10^{-7})$, die bei einer Schrittweite von $h = 5e-8$ bis 0.00819 akzeptabel ist. Wegen der guten Übereinstimmung mit der exakten Lösung (vergl. Abb. 7.102) verzichten wir auf ihre Darstellung. Dazu haben wir einige Fehlergrößen sowie die Folgen der Integrationspunkte und variablen Schrittweiten, die wir grafisch zeigen.

Die Vorgehensweise ist in den MATLAB-Anweisungen zu erkennen und kommentiert.

```
% BDF(1,2) mit dualer Schrittweitensteuerung
eps = 1e-6
t0 = 0
T = 1
x0 = [1 1]',           % eta(0)
h0 = 5e-8             % 5e-8 guenstig ohne Anfangs-Halbierung/Verdopplung
if t0+h0>T, h0 = T-t0; end;
[xh,yh] = f16exv(t0+h0); % eta(1)
x1 = [xh,yh]'

clear t h x xex yex
h(1) = h0;
t(1) = t0;
x(:,1) = x0;
t(2) = t0+h0;
x(:,2) = x1;

xim1 = x0;
xi = x1;
hi = h0;
ti = t0+h0;
anz = 0;                % Anzahl der Halbierungen = Verlust

for i = 2:100000        % obere Schranke hinreichend gross waehlen
    while 1>0,
        bd1 = inv(II-hi*A)*xi;           % BDF(1)
        bd2 = inv(3*II-2*hi*A)*(4*xi-xim1); % BDF(2)
        est = abs(bd2-bd1);
        if est<=eps
            break;
        else
            hi = 0.5*hi;
            xim1 = inv(II-hi*A)*xim1;
            anz = anz+1;
        end;
    end;
    h(i) = hi;
    ti = ti+hi;
    xhi = xi;
    xi = bd2;           % eta(i+1) von BDF(2)
    t(i+1) = ti;       % letzter Integrationspunkt >= T
    x(:,i+1) = bd2;
    if ti>=T
        break
    elseif est<0.25*eps
        hi = 2*hi;     % bei Verdopplung von h bleibt eta(i-1)
    else
        xim1 = xhi;    % wenn h unveraendert, dann eta(i)
    end;
end;
end;
```



```

h(i+1) = 2*hi; % letzte "empfohlene" Schrittweite
N = i+1 % Anzahl der Schritte
anz % Anzahl der Halbierungen -> Verlust
t(4831) % nahe der Phasengrenze
h(4831)

% exakte Loesung auf Gitter des BDF auswerten
for i = 1:N, [xex(i),yex(i)] = f16exv(t(i)); end;

% Fehler und Stellen
[aerr_x,ix] = max(abs(x(1,:)-xex(1:N)))
[aerr_y,iy] = max(abs(x(2,:)-yex(1:N)))
aerr_xN = abs(x(1,N)-xex(N))
aerr_yN = abs(x(2,N)-yex(N))

figure(13)
semilogy(h)
xlabel('i');
text(N/2,4e-7,'h(i)');
axis([0 8000 1e-8 1e-3]);
print bdf12_1.ps -dpsc

figure(14)
plot(t)
xlabel('i');
text(N/2,0.07,'t(i)');
axis([0 8000 0 1]);
print bdf12_2a.ps -dpsc

figure(15)
semilogy(t)
xlabel('i');
text(N/2,8e-4,'t(i)');
axis([0 8000 1e-8 1]);
print bdf12_2.ps -dpsc

figure(16)
semilogy(t,h)
xlabel('t');
text(0.5,2e-4,'h(t)');
axis([0 1 1e-8 1e-3]); % [0,0.001], [0.001,0.002], [0.002,1]
print bdf12_3.ps -dpsc

% Grafik: Naehungsloesung, exakte Loesung
figure(17)
plot(t,x(1,:),'b.:',t,xex,'k-');
title(' in [0,1]x[-6,1], Loesung x(t) mit AB x(0)=1 sowie BDF(1,2)');
xlabel('t');
text(0.5,-3,'x(t)');
text(0.5,-5,'BDF(1,2)');
axis([0 1 -6 1]);
% analog mit plot(t,x(2,:),'b.:',t,yex,'k-');

```

Einige Ergebnisse

```

N = 7354
anz = 0,           d.h. keine Halbierungen vorgenommen
t(4831) = 0.00050015, d.h. nahe der Phasengrenze 0.0005
h(4831) = 4e-7

[aerr_x,ix] = [3.896197728892048e-007, 5141], d.h. i=5140
[aerr_y,iy] = [1.948095817994044e-007, 5141]
aerr_xN = 2.343957934414220e-007
aerr_yN = 1.757968439708435e-007

```

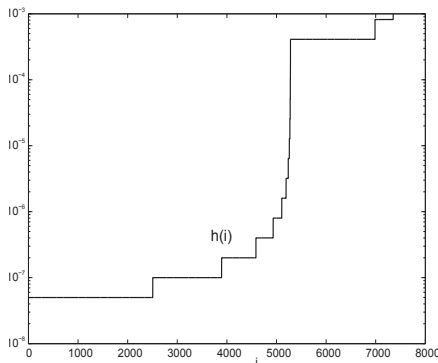


Abb. 7.109

Datei *bdf12_1.ps*

Zum AWP (7.48) in $[0, 1]$,

Verhalten der Schrittweite $h(i)$, $i = 1, \dots, N$,

bei Anfangsschrittweite $h_0 = 5e-8$,

MATLAB-Kommando `semilogy(h)`

Die Schrittweiten $h(i)$ nehmen stetig zu (\geq), wobei in der transienten Phase $[0, 0.0005]$ nur wenige Verdopplungen stattfinden und bei $t = 0.0005$ ist $h_i = 4e-7$. Erst für $t > 0.0005$ sind stärkere Zuwächse bei gleicher Genauigkeit möglich.

Betrachtet man die Punkte $t(i)$, $i = 1, \dots, N$, im Integrationsbereich $[0, 1]$, so fallen die meisten Werte in die kurze transiente Phase, ca. 5000. Ihr deutliches Anwachsen kommt anschließend.

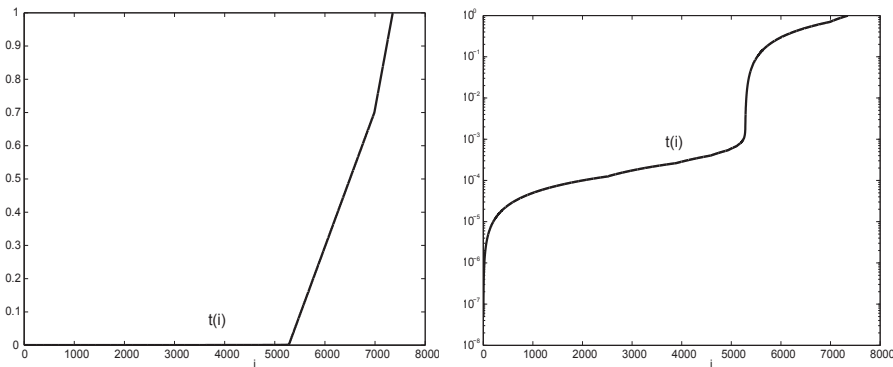


Abb. 7.110 Dateien *bdf12_2a*, *bdf12_2.ps*, zum AWP (7.48) in $[0, 1]$,

Verhalten der Integrationspunkte $t(i)$, $i = 1, \dots, N$, bei Anfangsschrittweite $h_0 = 5e-8$, MATLAB-Kommandos `plot(t)` und `semilogy(t)`

Für den Plot der Funktion $h(t)$ ist es sinnvoll, ihn intervallweise vorzunehmen.

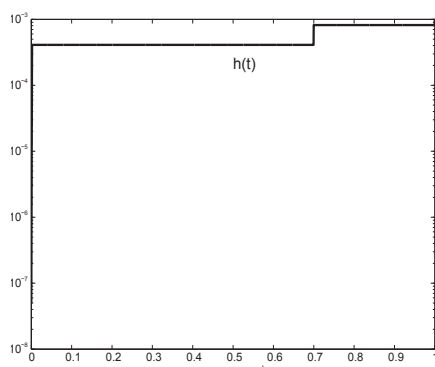


Abb. 7.111

Datei *bdf12_3.ps*

Zum AWP (7.48) in $[0, 1]$,

Verhalten der Schrittweite $h(t)$, $t \in [0, 1]$,

bei Anfangsschrittweite $h_0 = 5e-8$,

MATLAB-Kommando `semilogy(t,h)`

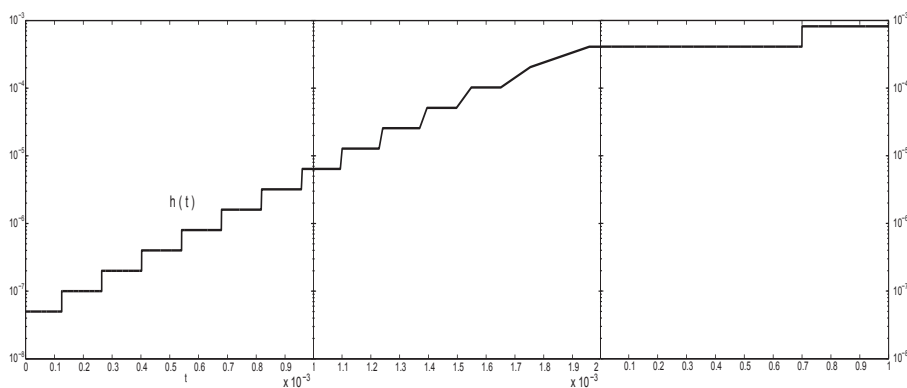


Abb. 7.112 Dateien *bdf12_3a, b, c.ps*, zum AWP (7.48) in $[0, 1]$,
Schrittweite $h(t)$, $t \in [0, 1]$, intervallweise zusammengesetzt
mit $[0, 0.001] \cup [0.001, 0.002] \cup [0.002, 1]$

Rechnen wir das gegebene Beispiel mit BDF(1,2) bei $h_0 = 10^{-7}$ oder größer, so handeln wir uns plötzlich eine Endlos-Schleife ein. Gleich in der Anfangsphase wird festgestellt, dass mit dieser Schrittweite der Fehler $|\eta_{i+1} - \tilde{\eta}_{i+1}|$ zu groß ist. Also wird η_{i+1} nicht akzeptiert, die Schrittweite halbiert, mit BDF(1) auf der Basis von η_{i-1} ein Zwischenwert $\hat{\eta}_{i-1/2}$ erzeugt und gemeinsam mit η_i ein neuer Versuch gestartet. Man merkt jedoch, dass $|\eta_{i+1/2} - \tilde{\eta}_{i+1/2}|$ nicht viel kleiner wird, auch nicht beim nächsten Mal nach erneuter Halbierung. Somit bleibt der Algorithmus in der Halbierungsphase (Schleife `while 1>0`) stecken. Ausschlaggebend dafür ist, dass $\hat{\eta}_{i-1/2}$ keine Größe aus dem ‘Konzept’ von BDF(2) ist.

Wir modifizieren also unsere Methode BDF(1,2) ein wenig.

Die duale SWS sowie die Bereitstellung von zwei genauen AB η_0, η_1 aufgrund der Kenntnis der exakten Lösung behalten wir bei.

Worin bestehen die Modifikationen?

1. In der Anfangsphase wird die Zulässigkeit der Anfangsschrittweite h_0 geprüft.
Wenn $|\eta_2 - \tilde{\eta}_2| > \varepsilon$ ist, dann wird h_0 halbiert. Das neue η_1 wird aus der Berechnung der exakten Funktionen genommen. Damit steht am Ende η_2 fest.
2. Die Anzahl der Halbierungsschritte, die aufwandsmäßig als Verlust einzustufen ist, wird getrennt für die Anfangsphase und den weiteren Verlauf erfasst.
3. Wir arbeiten aktuell mit den Größen $\eta_{i-1}, \eta_i, \eta_{i+1}$ und werden zusätzlich η_{i-2} mitführen, um das bisherige Problem “ $\hat{\eta}_{i-1/2}$ “ weitestgehend zu vermeiden.
4. Im Toleranzintervall $[\rho\varepsilon, \varepsilon]$, $0 < \rho < 1$, wählen wir wiederum $\rho = \frac{1}{4}$ wegen der Genauigkeitsordnung $\mathcal{O}(h)$ von BDF(1) und $\text{BDF}(2) \approx (x(t), y(t))$.
Man sagt dann, dass $|\text{BDF}(2) - \text{BDF}(1)|$ ein Schätzer für $|(x(t), y(t)) - \text{BDF}(1)|$ ist.
Wählt man ρ zu groß, dann ist die Gefahr einer ständigen Verdopplung/Halbierung.
Wählt man ρ sehr klein, dann werden bei kleinen Fehlern keine Verdopplungen der Schrittweite vorgeschlagen und das Verfahren kommt etwas langsamer voran.
5. Der Normalfall sind 4 aufeinander folgende Größen $\eta_{i-2}, \eta_{i-1}, \eta_i, \eta_{i+1}$ (Quadrupel) an äquidistanten Stellen. Ihr Abstand ist die aktuelle Schrittweite. η_{i+1} entsteht aus dem BDF(2). Das Quadrupel wird einfach “weiter geschoben“.
6. Die Schrittweitenverdopplung ist mit einem Indikator verknüpft. Er soll garantieren, dass ein solcher Vorschlag als Schritt unbedingt realisiert wird.
7. Bei $|\eta_{i+1} - \tilde{\eta}_{i+1}| \leq \rho\varepsilon < \varepsilon$ wird der Schritt zunächst akzeptiert und eine Schrittweitenverdopplung vorgeschlagen. Dazu wird der Indikator **verd=1** gesetzt.
Im neuen Schritt werden beiden Näherungen BDF(1) und BDF(2) berechnet und unabhängig vom Fehler das Ergebnis von BDF(2) akzeptiert. Der Indikator wird zurückgesetzt (**verd=0**). Eine sofort anschließende Verdopplung ist nicht möglich und das Quadrupel wird aktualisiert (Umspeicherung).
8. Falls $|\eta_{i+1} - \tilde{\eta}_{i+1}| \geq \varepsilon$ und der Indikator **verd=0** sind, dann wird η_{i+1} aus BDF(2) verworfen. Die neue Zwischengröße $\hat{\eta}_{i-1/2}$ entsteht mittels quadratischer Interpolation aus dem Tripel $\eta_{i-2}, \eta_{i-1}, \eta_i$, deren Stellen äquidistant mit Abstand h_i sind. Es gilt (Nachweis auch mit Taylor-Reihenentwicklung)

$$\hat{\eta}_{i-1/2} = \frac{1}{8}(3\eta_i + 6\eta_{i-1} - \eta_{i-2}).$$

Mit der halbierten Schrittweite und dem neuen Tripel $\eta_{i-1}, \hat{\eta}_{i-1/2}, \eta_i$ versucht man es weiter.

Die Modifikationen im Algorithmus sind in den MATLAB-Anweisungen zu erkennen.

```

% Modifikation von BDF(1,2) mit dualer Schrittweitensteuerung
eps = 1e-6
t0 = 0
T = 1
x0 = [1 1]' % eta(0)
h0 = 5e-7 % einige Anfangs-Halbierungen notwendig
if t0+h0>T, h0 = T-t0; end;
[xh,yh] = f16exv(t0+h0); % eta(1)
x1 = [xh,yh]'

clear t h x xex yex
t(1) = t0;
x(:,1) = x0;
xim1 = x0;
xi = x1;
anz1 = 0; % Anzahl der Halbierungen = Verlust, anfangs
while 1>0,
    bd1 = inv(II-h0*A)*xi; % BDF(1)
    bd2 = inv(3*II-2*h0*A)*(4*xi-xim1); % BDF(2)
    est = abs(bd2-bd1);
    if est<=eps
        break;
    else
        h0 = 0.5*h0;
        [xh,yh] = f16exv(t0+h0); % eta(1)
        xi = [xh,yh]';
        anz1 = anz1+1;
    end;
end;
h(1) = h0;
t(2) = t0+h0;
x(:,2) = xi;
h(2) = h0;
t(3) = t0+2*h0;
x(:,3) = bd2;

xim2 = x0;
xim1 = xi;
xi = bd2;
hi = h0;
ti = t0+2*h0;

anz2 = 0; % Anzahl der Halbierungen = Verlust, spaeter
verd = 0; % Indikator fuer Verdopplung
for i = 3:100000 % obere Schranke hinreichend gross waehlen
    while 1>0,
        bd1 = inv(II-hi*A)*xi; % BDF(1)
        bd2 = inv(3*II-2*hi*A)*(4*xi-xim1); % BDF(2)
        est = abs(bd2-bd1);
        if (est<=eps) | (verd==1)
            if verd==1, verd = 0; est = eps; end;
            break;
        end;
    end;
end;

```

```

else
    xim2 = xim1;
    % Interpolation und Auswertung
    xim1 = (3*xi+6*xim1-xim2)/8;
    hi = 0.5*hi;
    anz2 = anz2+1;
end;
end;
h(i) = hi;
ti = ti+hi;
xhi = xi;
xi = bd2; % eta(i+1) von BDF(2)
t(i+1) = ti; % letzter Integrationspunkt >= T
x(:,i+1) = bd2;
if ti>=T
    break
elseif est<0.25*eps
    hi = 2*hi; % bei Verdopplung von h bleibt eta(i-1)
    verd = 1;
else
    xim2 = xim1;
    xim1 = xhi; % wenn h unverändert, dann eta(i)
end;
end;
h(i+1) = 2*hi;
N = i+1
anz1,anz2
t(4990)
h(4990)

% exakte Loesung auf Gitter des BDF auswerten
for i = 1:N, [xex(i),yex(i)] = f16exv(t(i)); end;

% Fehler und Stellen
[aerr_x,ix] = max(abs(x(1,:)-xex(1:N)))
[aerr_y,iy] = max(abs(x(2,:)-yex(1:N)))
aerr_xN = abs(x(1,N)-xex(N))
aerr_yN = abs(x(2,N)-yex(N))

Einige Ergebnisse

% 1. Rechnung
h0 = 5e-8
N = 7355
anz1 = 0, anz2 = 0, d.h. keine Halbierungen vorgenommen
t(4831) = 0.0005002, d.h. nahe der Phasengrenze 0.0005
h(4831) = 4e-7

[aerr_x,ix] = [3.896197728892048e-007, 5141], d.h. i=5140
[aerr_y,iy] = [1.948095817994044e-007, 5141]
aerr_xN = 2.343557556905296e-007
aerr_yN = 1.757668164348303e-007

```

```

% 2. Rechnung
h0 = 5e-7 % analog 1e-6, 1e-5, ...
N = 7393
anz1 = 4, anz2 = 0, d.h. Halbierungen am Anfang vorgenommen
t(4990) = 0.00049990625, d.h. nahe der Phasengrenze 0.0005
h(4990) = 4e-7
[aeerr_x,ix] = [3.892179423203857e-007, 5427], d.h. i=5426
[aeerr_y,iy] = [1.946107186512336e-007, 5141]
aeerr_xN = 1.924794363894478e-007
aeerr_yN = 1.443595769590189e-007

```

Die grafischen Darstellungen der Ergebnisse sind denen in den Abbildungen 7.109 – 7.112 sehr ähnlich.

7.3.2 Steife Systeme und Linearisierung

Wir wollen nun eine weitere Lösungsstrategie mit Schrittweitenwahl untersuchen. Dabei stützen wir uns auf die Kenntnis der Eigenwerte und exakten Lösungskomponenten des SysgDGL. Darauf aufbauend entwickeln wir dann die Linearisierung.

Beispiel 7.8 Wir betrachten das lineare SysgDGL für $\mathbf{x} = (x_1, x_2, x_3)^T$

$$\mathbf{x}' = A\mathbf{x}, \quad \mathbf{x}(0) = (4, 13, 1)^T, \quad (7.49)$$

mit

$$A = \begin{pmatrix} -0.5 & 32.6 & 35.7 \\ 0 & -48.0 & 9.0 \\ 0 & 9.0 & -72.0 \end{pmatrix}. \quad (7.50)$$

Die Systemmatrix A besitzt die drei reellen Eigenwerte $\lambda_1 = -75$, $\lambda_2 = -45$ und $\lambda_3 = -\frac{1}{2}$. Die allgemeine Lösung

$$\begin{aligned} x(t) &= C_1 e^{-t/2} - C_2 e^{-45t} + C_3 e^{-75t}, \\ y(t) &= C_2 e^{-45t} + C_3 e^{-75t}, \\ z(t) &= \frac{1}{3} C_2 e^{-45t} - 3 C_3 e^{-75t}, \end{aligned} \quad (7.51)$$

sowie die exakte Lösung mit der AB

$$x(t) = 15e^{-t/2} - 12e^{-45t} + e^{-75t}, \quad y(t) = 12e^{-45t} + e^{-75t}, \quad z(t) = 4e^{-45t} - 3e^{-75t} \quad (7.52)$$

setzen sich zusammen aus Anteilen, welche wie $e^{\lambda_i t}$ gegen Null abklingen.

Es handelt sich dabei um ein SysgDGL, wo die Steifheit durch betragsunterschiedliche negative Eigenwerte der Systemmatrix A verursacht wird. Sie ist aber moderat.

Die Schrittweitenbeschränkung (7.13) ist bei $\lambda(A) < 0$ für das PZV

$$h < \frac{2}{\max_i |\lambda_i|} = \frac{2}{|\lambda_1|} = \frac{2}{75} = 0.026667$$

und kann auch für weitere ESV herangezogen werden.

Gemäß der Bemerkung 4.3 sollte man die etwas schärfere Schrittweitenbedingung $h < 1/\max_i |\lambda_i| = 1/75 = 0.013333$ verwenden, um evtl. oszillierendes Verhalten der Näherungslösung zu vermeiden.

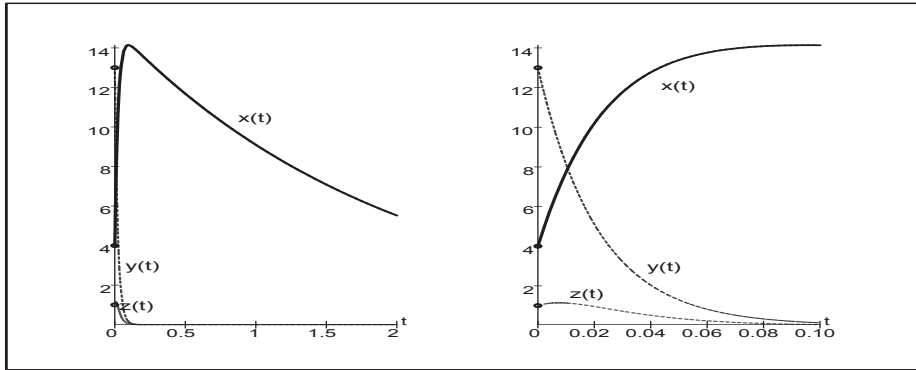


Abb. 7.113 Datei *nsteif70.ps*, Lösungen $x(t), y(t), z(t)$ zum AWP (4.46) in $[0, T]$

Das gegebene AWP soll nun mit dem KRKV gelöst werden.

Dieses hat nach Beispiel 7.4 (c) das Stabilitätsintervall $(-2.785293, 0)$. Hierdurch und durch den betragsgrößten negativen Eigenwert λ_3 wird die Stabilitätsobergrenze für die Schrittweite h festgelegt durch

$$h_{max} = \frac{2.785293}{|\lambda_3|} = 0.037137. \quad (7.53)$$

Diese Grenze muss im Folgenden immer berücksichtigt werden.

Wir fordern nun des Weiteren, dass

$$|F(h\lambda_i) - e^{h\lambda_i}| < \varepsilon = 10^{-5}, \quad i = 1, 2, 3, \quad (7.54)$$

$$F(h\lambda) = 1 + h\lambda + \frac{1}{2}(h\lambda)^2 + \frac{1}{6}(h\lambda)^3 + \frac{1}{24}(h\lambda)^4,$$

(vergl. Definition 7.2).

Ein Maß für die Genauigkeit zur Schrittweite h_{max} ist $|F(h_{max}\lambda_1) - e^{h_{max}\lambda_1}| = 0.938$. Soll dieser Wert auf etwa 10^{-5} verringert werden, so ist eine viel kleinere Schrittweite nötig. Dies führt nach numerischer Auswertung der linken Seite von (7.54) mit λ_1 auf die Wahl $h_1 = 0.0035$. Nach 44 Schritten mit h_1 ist man bei $t = 0.1540$, die wie $e^{\lambda_1 t}$ abklingenden Anteile sind um den Faktor $e^{0.1540\lambda_3} = 9.636 \cdot 10^{-6} < \varepsilon$ reduziert und werden bei der Schrittweitenwahl nicht mehr berücksichtigt. Die Reduktion der beiden anderen Anteile ist nur $e^{0.1540\lambda_2} = 9.780 \cdot 10^{-4}$ und $e^{0.1540\lambda_3} = 0.925$.

Die nächste Schrittwahl h_2 ergibt sich aus der Forderung (7.54) für λ_2 und führt auf $h_2 = 0.0058$. Nach weiteren 18 Zeitschritten können auch die wie $e^{\lambda_2 t}$ abklingenden Komponenten vernachlässigt werden. Es gilt

$$t = 0.0035 \cdot 44 + 0.0058 \cdot 18 = 0.2584, \quad e^{0.2584\lambda_2} = 8.91 \cdot 10^{-6}, \quad e^{0.2584\lambda_3} = 0.878.$$

Die Forderung (7.54) für $i = 3$ führt auf den Zeitschritt $h_3 = 0.525$, denn

$$|F(h_3\lambda_3) - e^{h_3\lambda_3}| = 0.994 \cdot 10^{-5}.$$

Da dieser Zeitschritt nicht mit der Stabilitätsforderung (7.53) im Einklang steht, könnten die wie $e^{\lambda_3 t}$ abklingenden Komponenten zu Instabilitäten führen. Somit wird als letzter Zeitschritt z.B. $h_3 = 0.037 < h_{max}$ festgelegt.

Die beschriebene Strategie mit Schrittweitenwahl ist aber nur aufgrund der Kenntnis der Eigenwerte und exakten Lösungskomponenten möglich.

Wie können die vorangegangenen Überlegungen auf allgemeinere, also insbesondere inhomogene nichtlineare Systeme erweitert werden? Dies soll an einem Beispiel demonstriert werden.

Beispiel 7.9 Die folgenden Gleichungen beschreiben ein SysgDGL, wie es typischerweise bei der Berechnung der chemischen Reaktion dreier Substanzen auftreten kann.

$$\mathbf{x}' = \begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{pmatrix} -0.1x_1 + 100x_2x_3 \\ 0.1x_1 - 100x_2x_3 - 500x_2^2 \\ 500x_2^2 - 0.5x_3 \end{pmatrix} \quad (7.55)$$

mit den AB $\mathbf{x}(0) = (4, 2, 0.5)^T$.

Um herauszufinden, mit welcher Schrittweite bei der numerischen Integration begonnen werden soll, untersuchen wir die Linearisierung des Systems um den Anfangsvektor $\mathbf{x}(0)$. Hierzu setzen wir $\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}(0)$ und vernachlässigen Terme, welche quadratisch in den Komponenten von $\Delta\mathbf{x}$ sind. Dies führt mit $\mathbf{x}(t) = \Delta\mathbf{x}(t) + \mathbf{x}(0)$ und $\mathbf{x}'(t) = \Delta\mathbf{x}'(t)$ über

$$\begin{aligned} \Delta\mathbf{x}' &= \begin{pmatrix} -0.1(\Delta x_1 + x_1(0)) + 100(\Delta x_2 + x_2(0))(\Delta x_3 + x_3(0)) \\ 0.1(\Delta x_1 + x_1(0)) - 100(\Delta x_2 + x_2(0))(\Delta x_3 + x_3(0)) - 500(\Delta x_2 + x_2(0))^2 \\ 500(\Delta x_2 + x_2(0))^2 - 0.5(\Delta x_3 + x_3(0)) \end{pmatrix} \\ &= \begin{pmatrix} -0.1\Delta x_1 + 100x_3(0)\Delta x_2 + 100x_2(0)\Delta x_3 \\ 0.1\Delta x_1 - (100x_3(0) + 1000x_2(0))\Delta x_2 - 100x_2(0)\Delta x_3 \\ 1000x_2(0)\Delta x_2 - 0.5\Delta x_3 \end{pmatrix} \\ &\quad + \begin{pmatrix} -0.1x_1(0) + 100x_2(0)x_3(0) & +100\Delta x_2 \Delta x_3 \\ 0.1x_1(0) - 100x_2(0)x_3(0) - 500x_2^2(0) & +100\Delta x_2 \Delta x_3 - 500(\Delta x_2)^2 \\ 500x_2^2(0) - 0.5x_3(0) & +500(\Delta x_2)^2 \end{pmatrix} \\ &= A \Delta\mathbf{x} + \begin{pmatrix} 99.6 \\ -2099.6 \\ 1999.75 \end{pmatrix} + \begin{pmatrix} 100\Delta x_2 \Delta x_3 \\ 100\Delta x_2 \Delta x_3 - 500(\Delta x_2)^2 \\ 500(\Delta x_2)^2 \end{pmatrix} \end{aligned}$$

auf das linearisierte System

$$\tilde{\mathbf{x}}' = A \tilde{\mathbf{x}} + \begin{pmatrix} 99.6 \\ -2099.6 \\ 1999.75 \end{pmatrix}, \quad \tilde{\mathbf{x}}(0) = \mathbf{0}, \quad (7.56)$$

mit

$$A = \begin{pmatrix} -0.1 & 100x_3(0) & 100x_2(0) \\ 0.1 & -100x_3(0) - 1000x_2(0) & -100x_2(0) \\ 0 & 1000x_2(0) & -0.5 \end{pmatrix} = \begin{pmatrix} -0.1 & 50 & 200 \\ 0.1 & -2050 & -200 \\ 0 & 2000 & -0.5 \end{pmatrix}.$$

Die Eigenwerte von A sind $\lambda_3 = -0.000\,249$, $\lambda_2 = -219.064\,614$ und $\lambda_1 = -1\,831.535\,136$. Für das KRKV ergibt sich hieraus eine Stabilitätsgrenze $h_{max} = \frac{2.785\,293}{1\,831.535\,136} = 0.001\,520$. Ein Maß für die Genauigkeit bei dieser Schrittweite ist $|F(h_{max}\lambda_1) - e^{h_{max}\lambda_1}| = 0.938$. Soll dieser Wert auf etwa 10^{-5} verringert werden, so ist eine Schrittweite $h = 0.000\,143$ nötig. Wegen der Nichtlinearität des SysdGdI muss eine Linearisierung zur Schrittweitenbestimmung jeweils nach einigen Schritten wiederholt werden.

Das AWP (7.55) ist analytisch nicht lösbar. Die numerische Lösung kann in Maple mittels `dsolve` und bei einer Methode für steife gDGI, z.B. `gear`, erfolgen.

```
> T:=0.5;      # 0.05
N:=100;
hh:=T/N;
loe6b:=dsolve({sys6,init},{fcns},numeric,method=gear,
              output=array([seq(hh*i,i=0..N)]));

> ko:=eval(loe6b[1,1]);
lo:=eval(loe6b[2,1]);
nr:=rowdim(lo);
> lisx:=seq([lo[i,1],lo[i,2]],i=1..nr):
lisy:=seq([lo[i,1],lo[i,3]],i=1..nr):
lisz:=seq([lo[i,1],lo[i,4]],i=1..nr):

> p1:=plot([lisx,lisy,lisz],color=[black,blue,red],linestyle=[1,2,3],
           thickness=3,labels=['t','']):
p2:=pointplot([[0,4],[0,2],[0,0.5]],symbol=circle,view=[0..T,0..5]):
p3:=textplot([[0.25,4.8,'x(t)'],[0.25,0.3,'y(t)'],[0.25,1.9,'z(t)']]):
display(p1,p2,p3);
```

Das KRKV (`method=classical[rk4]`) ist nicht zu empfehlen. Probleme können entstehen: entweder ist das Integrationsintervall zu groß, die Schrittweite `stepsize` zu groß, die Anzahl der Funktionsauswertungen `maxfun` zu klein eingestellt u.ä. Man muss erst an einigen Parametern geeignet "drehen", damit eine numerische Lösung erzeugt wird.

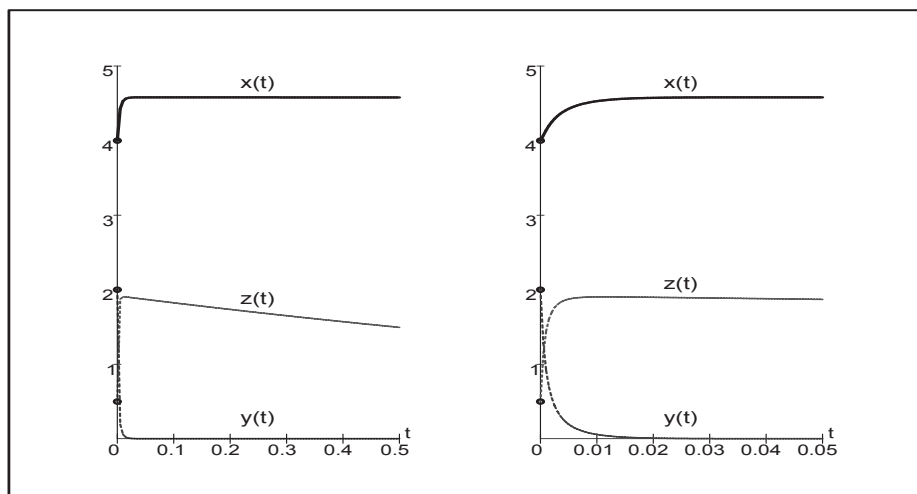


Abb. 7.114 Datei *nsteif71.ps*, numerische Lösungen $x(t), y(t), z(t)$ zum AWP (7.55) in $[0, T]$ mittels `dsolve(..., numeric, method=gear, ...)`

Die transiente Phase der Steifheit ist ungefähr das Anfangsintervall $[0, 0.0015]$.

Das linearisierte AWP (7.56) können wir exakt lösen. Die Lösungskomponenten sind

$$\begin{aligned}\tilde{x}(t) &= -504 + 505.244\,834\,126\,340\,918 e^{-0.000249\,236\,998\,989\,403\,52\,t} \\ &\quad -1.352\,106\,278\,157\,4651 e^{-219.064614\,714\,585\,44\,t} \\ &\quad +0.107\,272\,151\,816\,5464 e^{-1831.535\,136\,048\,4155\,t}, \\ \tilde{y}(t) &= -1 + 0.000\,062\,962\,860\,952\,611\,34 e^{-0.000249\,236\,998\,989\,403\,52\,t} \\ &\quad -0.166\,316\,332\,038\,2160 e^{-219.064614\,714\,585\,44\,t} \\ &\quad +1.166\,253\,369\,177\,2634 e^{-1831.535\,136\,048\,4155\,t}, \\ \tilde{z}(t) &= -0.5 + 0.251\,977\,047\,816\,8695 e^{-0.000249\,236\,998\,989\,403\,52\,t} \\ &\quad +1.521\,896\,234\,259\,1613 e^{-219.064614\,714\,585\,44\,t} \\ &\quad -1.273\,873\,282\,076\,0308 e^{-1831.535\,136\,048\,4155\,t}.\end{aligned}$$

Sie erfüllen die AB auf eine Genauigkeit von $\mathcal{O}(10^{-15})$. Nach Einsetzen in die gDGI sind die Koeffizienten vor den Termen $e^{\lambda_i t}$ von der Ordnung $\mathcal{O}(10^{-12})$.

Wir überprüfen, wie lange die Lösung $\tilde{\mathbf{x}}(t)$ der linearisierten Version und damit die Funktion $\tilde{\mathbf{x}}(t) + \mathbf{x}(0)$ als gute Approximation der Lösung $\mathbf{x}(t)$ verwendet werden kann. Eine tendenzielle Übereinstimmung der jeweils rechten Bilder in den Abb. 7.114 und 7.115 ist im kleinen Intervall $[0, 0.001]$ zu erkennen. Weiter rechts sind die Unterschiede beträchtlich.

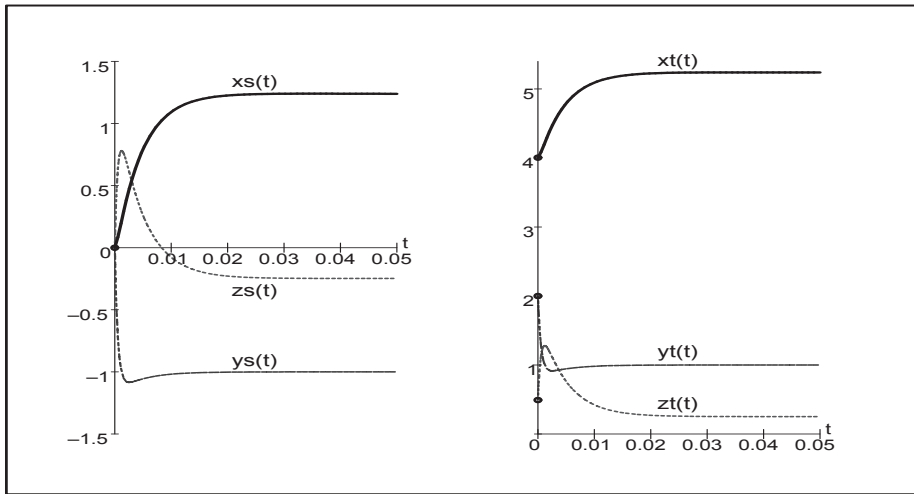


Abb. 7.115 Datei *nsteif72.ps*, exakte Lösung $\mathbf{x}\mathbf{s}(t) = \tilde{\mathbf{x}}(t) = (\tilde{x}(t), \tilde{y}(t), \tilde{z}(t))^T$ zum AWP (7.56) in $[0, 0.05]$ sowie transformierte Funktion $\mathbf{x}\mathbf{t}(t) = \tilde{\mathbf{x}}(t) + \mathbf{x}(0)$

Genauso ist ein numerischer Vergleich zwischen $\mathbf{x}(t)$ aus der *dsolve*-Rechnung als sehr gute Näherung zu $\mathbf{x}(t)$ und der transformierten Funktion $\mathbf{x}\mathbf{t}(t) = \tilde{\mathbf{x}}(t) + \mathbf{x}(0)$ mit $\tilde{\mathbf{x}}(t)$ aus dem linearisierten AWP (7.56) für ausgewählte Stellen möglich. Wir machen das zu den Werten $t_i = i/10\,000$, $i = 0, 1, \dots, 15$.

$t(i)$	$x(t(i))$	$xt(t(i))$	$x(t(i))$	$xt(t(i))$	$x(t(i))$	$xt(t(i))$
0.0000	4.000000	4.000000	2.000000	2.000000	0.500000	0.500000
0.0001	4.011213	4.011332	1.807922	1.808420	0.680835	0.680218
0.0002	4.024220	4.025034	1.646064	1.649428	0.829649	0.825471
0.0003	4.038240	4.040617	1.507897	1.517558	0.953751	0.941713
0.0004	4.052766	4.057675	1.388641	1.408261	1.058430	1.033903
0.0005	4.067465	4.075873	1.284717	1.317747	1.147601	1.106165
0.0006	4.082112	4.094934	1.193389	1.242862	1.224221	1.161932
0.0007	4.096563	4.114628	1.112537	1.180981	1.290559	1.204060
0.0008	4.110721	4.134766	1.040486	1.129917	1.348386	1.234925
0.0009	4.124525	4.155194	0.975900	1.087849	1.399100	1.256503
0.0010	4.137937	4.175784	0.917699	1.053262	1.443818	1.270437
0.0011	4.150937	4.196429	0.865000	1.024895	1.483444	1.278095
0.0012	4.163516	4.217047	0.817076	1.001697	1.518714	1.280612
0.0013	4.175673	4.237566	0.773320	0.982794	1.550236	1.278931
0.0014	4.187411	4.257930	0.733225	0.967458	1.578514	1.273839
0.0015	4.198740	4.278095	0.696362	0.955084	1.603969	1.265985

An der Stelle $t \approx 0.008$ hat man die Abweichungen schon in der ersten Nachkommastelle. Generell ist es problematisch, in der transienten Phase eines steifen SysgDGI zu linearisieren.

Kapitel 8

Implizite Einschrittverfahren

8.1 Motivation für implizite Schrittverfahren

Die Verbesserung der Genauigkeit der Näherungslösungen in Schrittverfahren ist abhängig von der Schrittweite h und/oder der Schritt-/Stufenzahl s . Kleineres h bzw. größeres s führen aber oft zu einem wachsenden Aufwand und sind nur beschränkt einsetzbar. Dass man manchmal nicht den gewünschten Erfolg erzielt, hängt mit den der gDGL innewohnenden Eigenschaften und denen des Diskretisierungsverfahrens zusammen. Deshalb sind wir schon in mehreren Abschnitten den impliziten Schrittverfahren begegnet, insbesondere dort, wo es um Stabilitätsbetrachtungen von Verfahren und ihren diesbezüglich günstigen Eigenschaften geht.

Implizites EV im Abschnitt 7.1

$$\eta_{i+1} = \eta_i + hf(t_{i+1}, \eta_{i+1}). \quad (8.1)$$

Zweistufiges implizites RKV im Abschnitt 7.1

$$\begin{aligned} k_1 &= f\left(t_i + \frac{3-\sqrt{3}}{6}h, \eta_i + \frac{1}{4}hk_1 + \frac{3-2\sqrt{3}}{12}hk_2\right), \\ k_2 &= f\left(t_i + \frac{3+\sqrt{3}}{6}h, \eta_i + \frac{3+2\sqrt{3}}{12}hk_1 + \frac{1}{4}hk_2\right), \\ \eta_{i+1} &= \eta_i + \frac{h}{2}(k_1 + k_2). \end{aligned} \quad (8.2)$$

Implizites s -Schritt-Adams-Moulton-Verfahren mit $s = \max(r, 1) \geq 1$ im Abschnitt 6.1.2

$$\eta_i = \eta_{i-1} + h(\beta_0 f_i + \beta_1 f_{i-1} + \dots + \beta_r f_{i-r}). \quad (8.3)$$

Im Abschnitt 7.3.1 Bsp. 7.7 (b) wurde ausführlich ein steifes SysgDGL betrachtet, auf das zunächst noch einmal eingegangen wird.

Das AWP für $\mathbf{x} = (x, y)^T$ ist [97]

$$\mathbf{x}'(t) = A \mathbf{x}(t), \quad \mathbf{x}(0) = (1, 1)^T, \quad t \in [0, 1], \quad \text{wobei } A = \begin{pmatrix} -29\,998 & -39\,996 \\ 29\,997/2 & 19\,997 \end{pmatrix}. \quad (8.4)$$

Die allgemeine Lösung lautet

$$x(t) = C_1 e^{-t} + C_2 e^{-10000t}, \quad y(t) = -\frac{3}{4}C_1 e^{-t} - \frac{1}{2}C_2 e^{-10000t}, \quad (8.5)$$

die des AWP mit den gegebenen AB

$$x(t) = -6e^{-t} + 7e^{-10000t}, \quad y(t) = \frac{9}{2}e^{-t} - \frac{7}{2}e^{-10000t}. \quad (8.6)$$

Die Eigenwerte der Systemmatrix A sind $\lambda_{1,2} = -1, -10\,000$. Damit müssen explizite Verfahren mit einer Schrittweitenbeschränkung der Größenordnung $h = 10^{-4} \dots 10^{-3}$ rechnen. Die Steifheit des SysgDGI erkennt man an den betragsgroßen partiellen Ableitungen der rechten Seite nach x, y , d.h. wenn $\|f_x(t, \mathbf{x})\| \gg \|\mathbf{x}\|$ ist.

Das SysgDGI (8.4) besitzt eine transiente und asymptotische Phase der Lösung mit der Phasengrenze bei ungefähr $t = 0.0005$. Bis dahin verkleinert sich der Lösungsanteil $e^{-10000t}$ auf die Größe 0.00674, ein Wert, der mit der anderen Lösungskomponente erst bei $t = 5$ erreicht wird.

Bei $t = 0.0000154166$ hat die Lösung $x(t)$ den Nulldurchgang. Mit der Darstellung $x(t) = e^{-10000t} - 6(e^{-t} - e^{-10000t})$ erkennt man den zunächst fallenden Verlauf der Funktion in den negativen Bereich hinein und dann die Tendenz gegen Null von unten.

$$x(t) \begin{cases} \geq 0, & x \in [0, 0.0000154166], \\ < 0, & x \in (0.0000154166, 1], \end{cases} \quad x'(t) \begin{cases} \leq 0, & x \in [0, 0.000936542], \\ > 0, & x \in (0.000936542, 1]. \end{cases}$$

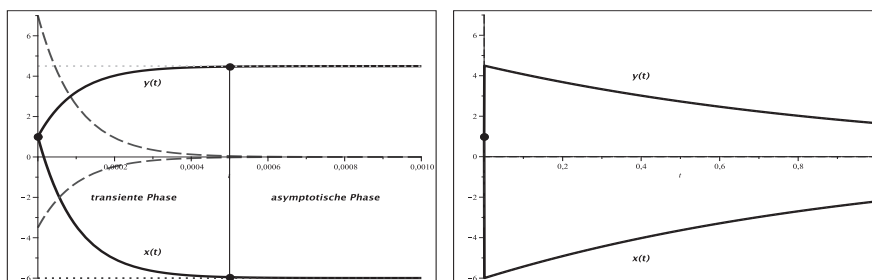


Abb. 8.1 Dateien *nsteif82..3.ps*, AWP (8.4),

Lösungen $x(t) = -6e^{-t} + 7e^{-10000t}$, $y(t) = \frac{9}{2}e^{-t} - \frac{7}{2}e^{-10000t}$,
 dazu die Lösungsanteile $-6e^{-t}$, $7e^{-10000t}$, $\frac{9}{2}e^{-t}$, $-\frac{7}{2}e^{-10000t}$

Beschreibung des Lösungsverhaltens

Transiente Phase	Asymptotische Phase
Meistens weniger interessant, extrem starke Schwankung der Lösung, Schrittweite h sehr klein wählen, viele Funktionsauswertungen.	Interessantes Lösungsintervall, moderater Verlauf der Lösung, h könnte man eigentlich wieder vergrößern, aber die Steifheit des SysgDGI verhindert das.

Die Integration mit dem eingebetteten (5,6)-RKV von H.J. Verner (vergl. Abschnitt 5.4.2) mit automatischer Schrittweitensteuerung in $[0, 1]$ bei der Toleranz $\varepsilon = 10^{-8}$ zeigt den hohen Aufwand an Funktionswertberechnungen. Ursache ist auch die "geringe" Stabilität des Verfahrens.

	Anzahl der Schritte bis t	aktueller Wert t	aktuelle Schrittweite h
Phase 1	7	1.01e-4	5.03e-5
	15	7.75e-4	1.33e-4
Phase 2	25	3.89e-3	4.82e-4
	50	1.41e-2	4.19e-4
	100	3.47e-2	4.25e-4
	200	7.53e-2	4.39e-4
	300	1.17e-1	4.13e-4
	400	1.57e-1	4.26e-4
	...		
2000	1	$\approx 4e-4$	

Tab. 8.1
Eingebettetes (5, 6)–RKV
mit SWS,
Aufwand an Schritten
und Schrittweiten

Die SWS verlässt sich darauf, dass das Problem eigentlich nicht steif ist. Wenn es jedoch steif ist, muss es seine Unkenntnis mit einem hohen Aufwand “bezahlen”. Also braucht man noch andere Verfahrensklassen und die Untersuchung ihrer Stabilität.

8.2 Das implizite Euler-Verfahren

Die Formulierung und die Eigenschaften von impliziten ESV lassen sich anschaulich am Euler-Verfahren demonstrieren. Dabei ist gut zu erkennen, welche Aspekte im Vergleich zum expliziten PZV hinzukommen.

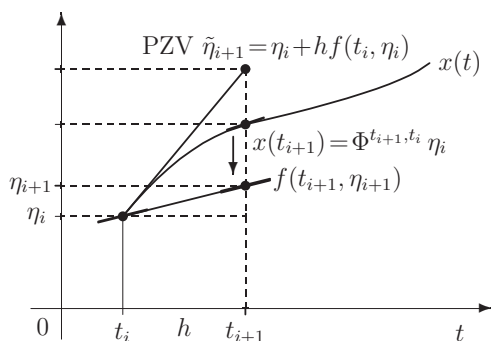


Abb. 8.2 Datei *iev1.pic*,
Geometrische Illustration des
IEV $\eta_{i+1} = \eta_i + h f(t_{i+1}, \eta_{i+1})$
zum AWP $x'(t) = f(t, x(t))$

Mittels des Rückwärtsdifferenzenquotienten gelangt man auf das IEV bei konstanter Schrittweite $h = t_{i+1} - t_i = \frac{T-t_0}{N}$ (Vektorschreibweise)

$$\frac{\mathbf{x}(t_{i+1}) - \mathbf{x}(t_i)}{h} \approx \mathbf{f}(t_{i+1}, \mathbf{x}(t_{i+1})), \quad \eta_i \approx \mathbf{x}(t_i), \quad (8.7)$$

$$\eta_{i+1} = \eta_i + h \mathbf{f}(t_{i+1}, \eta_{i+1}), \quad i = 0, 1, \dots, N-1, \quad \eta_0 = \mathbf{x}(t_0).$$

Die Lösung kann in jedem Schritt iterativ erfolgen, wobei die Konvergenz der inneren Iteration zu klären ist. Möglich wäre auch eine Linearisierung mit Betrachtung eines linearen Gleichungssystems. Schließlich stellt sich noch die Frage des Diskretisierungsfehlers bzw. der Genauigkeit des Verfahrens.

Mittels der Trapezregel der numerischen Integration erhält man unmittelbar eine zweites implizites ESV, nämlich das implizite Heun-Verfahren (IHV), der Gestalt

$$\eta_{i+1} = \eta_i + \frac{h}{2} (\mathbf{f}(t_i, \eta_i) + \mathbf{f}(t_{i+1}, \eta_{i+1})), \quad i = 0, 1, \dots, N-1, \quad \eta_0 = \mathbf{x}(t_0). \quad (8.8)$$

Beispiel 8.1

(1) Gegeben sei auf $I = [t_0, T] = [0, 1]$ das AWP

$$x'(t) = 2\sqrt{x(t)}, \quad x(0) = x_0 = 0. \quad (8.9)$$

Die allgemeine Lösung ist gegeben durch

$$x(t) = \begin{cases} (t-s)^2 & \text{für } t \geq s \geq 0, \\ 0 & \text{sonst,} \end{cases} \quad (8.10)$$

Sie enthält auch die Nulllösung, die spezielle Lösung $x(t) = t^2$, und das AWP ist wegen des frei wählbaren Parameters s nicht eindeutig lösbar.

Das PZV und andere explizite ESV liefern als Näherung des AWP die Nulllösung.

Welche Approximationen können mit den impliziten ESV (8.7) und (8.8) erzeugt werden?

(a) IEV

$$\begin{aligned} \eta_{i+1} &= \eta_i + 2h\sqrt{\eta_{i+1}}, \quad i = 0, 1, \dots, N-1, \quad \eta_0 = 0, \\ \eta_{i+1}^{(k+1)} &= \eta_i + 2h\sqrt{\eta_{i+1}^{(k)}}, \quad k = 0, 1, \dots, \quad \eta_{i+1}^{(0)} = \begin{cases} \eta_i, \\ c \in (0, c_0). \end{cases} \end{aligned}$$

Die Konvergenzuntersuchung basiert auf der Fixpunktgleichung $z = g(z) = a + 2h\sqrt{z}$, $a \geq 0$, mit der Ableitung $g'(z) = \frac{h}{\sqrt{z}} > 0$. Die Kontraktionsbedingung $|g'(z)| < 1$ führt auf die Schrittweitenbeschränkung $h < \sqrt{z}$, die aber eher unproblematisch ist.

Für den Anfangswert $\eta_0 = x_0 = 0$ folgt $a = 0$ und in jedem Schritt mit dem Startwert $\eta_{i+1}^{(0)} = 0$ sofort auch $\eta_{i+1}^{(k+1)} = 0$. Damit bleibt man bei der Nulllösung.

Sollte jedoch $a > 0$ oder $\eta_{i+1}^{(0)} > 0$ sein, werden die Näherungen keine Nullfolge mehr bilden, sondern die Approximation einer Lösungsparabel darstellen.

Für $a = 1$ z.B. gelten für die Lösung z^* der Fixpunktgleichung $z = g(z)$

$$z^* = 1 + 2h(h + \sqrt{1+h^2}), \quad 0 < g'(z^*) = \frac{h}{h + \sqrt{1+h^2}} < 1,$$

sodass die Kovvergenz der inneren Iteration gesichert ist.

Interessanter ist es, wenn man die Formel des IEV als Nullstellenaufgabe formuliert und mit $z = \sqrt{\eta_{i+1}}$ die quadratische Gleichung $z^2 - 2hz - \eta_i = 0$ bzgl. z löst, d.h.

$$z_{1,2} = h \pm \sqrt{h^2 + \eta_i}.$$

Damit ergeben sich bei $\eta_0 = 0$ die Werte $z_2^2 = \eta_{i+1} = 0$ für alle i , während mit der Lösungskomponente z_1 Näherungswerte zur Lösung $x(t) = t^2$ bei $t = t_i = ih$ erzeugt werden.

Es gelten die Beziehungen

$$\begin{aligned}\eta_0 &= 0 && = x(0) = 0, \\ \eta_1 &= 4h^2 && > x(h) = h^2, \\ \eta_2 &= (6 + 2\sqrt{5})h^2 && > x(2h) = 4h^2, \quad \text{und} \quad \lim_{h \rightarrow 0} \eta_i = x(t_i). \\ \eta_3 &= (8 + 2(\sqrt{5} + \sqrt{7 + 2\sqrt{5}}))h^2 && > x(3h) = 9h^2, \\ \text{usw.}\end{aligned}$$

(b) IHV

Ähnliche Betrachtungen kann man auch für dieses Verfahren anstellen.

Wir notieren die Beziehungen.

$$\begin{aligned}\eta_{i+1} &= \eta_i + h(\sqrt{\eta_i} + \sqrt{\eta_{i+1}}), \quad i = 0, 1, \dots, N-1, \\ \eta_{i+1} - h\sqrt{\eta_{i+1}} - (\eta_i + h\sqrt{\eta_i}) &= 0, \\ z^2 - hz - (\eta_i + h\sqrt{\eta_i}) &= 0, \quad z = \sqrt{\eta_{i+1}}, \\ z_{1,2} &= \frac{h}{2} \pm \sqrt{\frac{h^2}{4} + \eta_i + h\sqrt{\eta_i}}, \\ z_2 &= \sqrt{\eta_{i+1}} = 0 \quad \text{bei} \quad \eta_0 = 0 \quad \text{für alle } i, \\ z_1 &= \frac{h}{2} + \sqrt{\frac{h^2}{4} + \eta_i + h\sqrt{\eta_i}}, \quad \eta_0 = 0, \\ \eta_{i+1} &= z_1^2, \quad \eta_0 = 0 = x(0), \\ \eta_1 &= h^2 = x(h), \\ \eta_2 &= (2h)^2 = x(2h), \\ \eta_i &= (ih)^2 = x(ih) \quad \text{für alle } i.\end{aligned}$$

Das IHV approximiert die Lösung sogar exakt (Nachweis der Übereinstimmung $\eta_i = x(ih)$ mittels vollständiger Induktion bzgl. i).

(2) SysgDgl mit konstanter Matrix

$$\mathbf{x}'(t) = A\mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0 = I, \quad \mathbf{x}(t) = e^{At}. \quad (8.11)$$

Die Diskretisierung in $[t_0, T] = [0, T]$ erfolgt bei konstanter und beschränkter Schrittweite $h = t_{i+1} - t_i = \frac{T-t_0}{N}$, $h \leq h_0$.

(a) PZV

$$\begin{aligned}\eta_{i+1} &= \eta_i + hA\eta_i, \quad \eta_0 = I, \\ &= (I + hA)\eta_i \approx e^{hA}\eta_i, \\ &= (I + hA)^{i+1}\eta_0 \approx e^{(i+1)hA} = e^{At_{i+1}}.\end{aligned}$$

Die Matrix $I + hA$ bezeichnet man auch als Übertragungs- oder Verstärkungsfunktion.

(a) IEV

$$\begin{aligned}
 \eta_{i+1} &= \eta_i + hA\eta_{i+1}, \quad \eta_0 = I, \\
 (I - hA)\eta_{i+1} &= \eta_i, \\
 \eta_{i+1} &= (I - hA)^{-1}\eta_i, \\
 &= (I + hA + h^2A^2 + h^3A^3 + \dots)\eta_i, \\
 &\approx (I + hA + \frac{(hA)^2}{2!} + \frac{(hA)^3}{3!} + \dots)\eta_i = e^{hA}\eta_i.
 \end{aligned}$$

Zu Testzwecken betrachtet man hier Matrizen mit reellen und negativen Eigenwerten, damit man eine abnehmende Lösung erhält.

Damit ist beim IEV die Stabilität für eine beliebige Schrittweite h garantiert. Allein die Genauigkeitsforderung wird die Beschränkung $h \leq h_0$ nach sich ziehen.

Die Lösung des linearen Gleichungssystems $(I - hA)\eta_{i+1} = \eta_i$ verlangt z.B. eine einmalige LU -Zerlegung der Matrix $I - hA$. Man kann aber auch ein iteratives Verfahren dazu verwenden, wobei man möglichst wenig Iterationen brauchen sollte.

Zur iterativen Lösung des allgemeinen IEV im skalaren Fall

$$\eta_{i+1} = \eta_i + h f(t_{i+1}, \eta_{i+1}), \quad i = 0, 1, \dots, N-1, \quad \eta_0 = x_0, \quad (8.12)$$

folgen nun detaillierte Betrachtungen.

Die innere Iteration im i -ten Schritt ist

$$\eta_{i+1}^{(k+1)} = \eta_i + h f(t_{i+1}, \eta_{i+1}^{(k)}), \quad k = 0, 1, 2, \dots, \quad \eta_{i+1}^{(0)} \text{ gegeben}, \quad (8.13)$$

wobei in vielen Fällen mit $\eta_{i+1}^{(0)} = \eta_i$, $\eta_{i+1}^{(0)} = \eta_i + h f(t_i, \eta_i)$ oder aus einem anderen expliziten ESV schon gute Startwerte definiert sind.

Konvergenzuntersuchungen

(a) Fixpunktgleichung, Fixpunktiteration

$$\begin{aligned}
 z &= y + h f(t, z) = g(z), \quad t, y, h \text{ Parameter}, \\
 z^{(k+1)} &= g(z^{(k)}).
 \end{aligned} \quad (8.14)$$

Konvergenzbedingungen

$$|g'(z)| < 1, \quad h \left| \frac{\partial f(t, z)}{\partial z} \right| < 1, \quad h \left| \frac{\partial f(t, x)}{\partial x} \right| < 1. \quad (8.15)$$

Für hinreichend kleines $h \leq h_0$ ist das in der Umgebung des Fixpunktes stets erreichbar. Aber die Ungleichung kann eine starke Forderung an h sein.

(b) Nullstellengleichung, Newtonsches Iterationsverfahren

$$\begin{aligned}
 \varphi(z) &= z - h f(t, z) - y = 0, \quad \varphi'(z) = 1 - h f_z(t, z), \\
 z^{(k+1)} &= z^{(k)} - \frac{\varphi(z^{(k)})}{\varphi'(z^{(k)})} = z^{(k)} - \frac{z^{(k)} - h f(t, z^{(k)}) - y}{1 - h f_z(t, z^{(k)})}.
 \end{aligned} \quad (8.16)$$

Falls der Startwert $z^{(0)}$ hinreichend gut ist, dann ist die Konvergenz des Newton-Verfahrens gesichert. Es ist ja zugleich ein Fixpunktverfahren mit $g(z) = z - \frac{\varphi(z)}{\varphi'(z)}$ mit $g'(z^*) = 0$ und hat quadratische Konvergenzordnung. Daraus sind Einschränkungen für h nicht zu befürchten.

(c) Nullstellengleichung, modifizierte Newtonsche Iterationsverfahren

$$\begin{aligned}\varphi(z) &= z - h f(t, z) - y = 0, \quad \varphi'(z) = 1 - h f_z(t, z), \\ z^{(k+1)} &= z^{(k)} - \frac{\varphi(z^{(k)})}{\varphi'(z^{(0)})} = z^{(k)} - \frac{z^{(k)} - h f(t, z^{(k)}) - y}{1 - h f_z(t, z^{(0)})},\end{aligned}\tag{8.17}$$

bzw. Sekantenverfahren

$$z^{(k+1)} = z^{(k)} - \frac{\varphi(z^{(k)})}{\frac{\varphi(z^{(k)}) - \varphi(z^{(k-1)})}{z^{(k)} - z^{(k-1)}}} = z^{(k)} - \frac{z^{(k)} - h f(t, z^{(k)}) - y}{1 - h \frac{f_z(t, z^{(k)}) - f_z(t, z^{(k-1)})}{z^{(k)} - z^{(k-1)}}}.$$

Die Ableitung muss dabei entweder nur einmal bestimmt werden (bei Systemen vorteilhaft) bzw. wird ersetzt durch den Differenzenquotienten (aber 2 Startwerte notwendig). Es bleibt aber zumindest die überlineare Konvergenz der Iterationsverfahren.

(d) Linearisierungen für autonome gDGL bzw. SysgDGL

Betrachten wir also $x' = f(x)$ und das IEV in der Gestalt

$$\eta_{i+1} = \eta_i + h f(\eta_{i+1}), \quad i = 0, 1, \dots, N-1, \quad \eta_0 = x(t_0).\tag{8.18}$$

Zur Linearisierung verwenden wir die Taylor-Reihenentwicklung und ihre Partialsumme.

$$\begin{aligned}\eta_{i+1} &= \eta_i + h (f(\eta_i) + (\eta_{i+1} - \eta_i) f_x(\eta_i) + \dots), \\ \eta_{i+1} &\doteq (1 - h f_x(\eta_i)) \eta_i + h f(\eta_i) + h \eta_{i+1} f_x(\eta_i), \\ (1 - h f_x(\eta_i)) \eta_{i+1} &= (1 - h f_x(\eta_i)) \eta_i + h f(\eta_i), \\ \eta_{i+1} &= \eta_i + \frac{h}{1 - h f_x(\eta_i)} f(\eta_i),\end{aligned}\tag{8.19}$$

$$(I - h \mathbf{f}_x(\eta_i)) (\eta_{i+1} - \eta_i) = h \mathbf{f}(\eta_i) \quad \text{bei SysgDGL.}$$

Man bezeichnet diese Formel auch als semiimplizite Euler-Diskretisierung oder linear-implizites Euler-Verfahren (LIEV).

Sie macht nur eine Berechnung an der Stelle t_{i+1} und entspricht dem ersten Schritt des Newton-Verfahrens (8.16) mit einem speziellen Startwert $z^{(0)}$.

$$\begin{aligned}\eta_{i+1}^{(k+1)} &= \eta_{i+1}^{(k)} - \frac{\eta_{i+1}^{(k)} - h f(\eta_{i+1}^{(k)}) - \eta_i}{1 - h f_x(\eta_{i+1}^{(k)})}, \quad k = 0, 1, \dots, \\ k = 0, \quad \eta_{i+1}^{(0)} &= \eta_i, \\ \eta_{i+1}^{(1)} &= \eta_{i+1}^{(0)} - \frac{\eta_{i+1}^{(0)} - h f(\eta_{i+1}^{(0)}) - \eta_i}{1 - h f_x(\eta_{i+1}^{(0)})} \\ &= \eta_i - \frac{\eta_i - h f(\eta_i) - \eta_i}{1 - h f_x(\eta_i)} \\ &= \eta_i + \frac{h}{1 - h f_x(\eta_i)} f(\eta_i).\end{aligned}\tag{8.20}$$

Die rechte Seite stimmt mit der der Linearisierung (8.19) überein.

Explizite und semiimplizite Verfahren lassen sich somit in einer gemeinsamen Darstellung zusammenfassen.

$$(1 - h A) \Delta \eta_i = h f(\eta_i), \quad \Delta \eta_i = \eta_{i+1} - \eta_i,$$

$$A = \begin{cases} 0, & \text{explizite Form,} \\ f_x(\eta_i), & \text{semiimplizite Form,} \\ \dots, & \text{sonst.} \end{cases} \quad (8.21)$$

Will man anschließend die Größe η_{i+1} weiter verbessern, muss man die nächsten Newton-Iterationen ausführen.

In der "vollen" Version bedeutet dies mit $\Delta \eta_{i+1}^{(k)} = \eta_{i+1}^{(k+1)} - \eta_{i+1}^{(k)}$

$$(1 - h A_{i+1}^{(k)}) \Delta \eta_{i+1}^{(k)} = -(\eta_{i+1}^{(k)} - h f(\eta_{i+1}^{(k)}) - \eta_i), \quad k = 0, 1, 2, \dots, \quad (8.22)$$

wobei $\eta_{i+1}^{(0)} = \eta_i$, $A_{i+1}^{(k)} = f_x(\eta_{i+1}^{(k)})$ sind. Damit ist wiederum die erste Iterierte $\eta_{i+1}^{(1)} = \eta_{i+1}$ aus dem semiimpliziten Verfahren (8.19).

Vereinfachte Versionen der Newton-Iteration sparen Berechnungen von $A_{i+1}^{(k)}$ ein.

So kann man entweder die stationäre Iterationsformel

$$(1 - h A) \Delta \eta_{i+1}^{(k)} = -(\eta_{i+1}^{(k)} - h f(\eta_{i+1}^{(k)}) - \eta_i), \quad k = 0, 1, 2, \dots, \quad (8.23)$$

wobei $\eta_{i+1}^{(0)} = \eta_i$, $A = A_{i+1}^{(0)} = f_x(\eta_i)$ sind, oder eine Stufenformel mit zwei Berechnungen von $A_{i+1}^{(k)}$ gemäß

$$(1 - h A) \Delta \eta_i = h f(\eta_i), \quad A = f_x(\eta_i), \quad \eta_{i+1}^{(0)} = \eta_i, \quad \eta_{i+1}^{(1)} = \eta_{i+1},$$

$$(1 - h A_{i+1}^{(1)}) \Delta \eta_{i+1}^{(k)} = -(\eta_{i+1}^{(k)} - h f(\eta_{i+1}^{(k)}) - \eta_i), \quad k = 1, 2, \dots, \quad A_{i+1}^{(1)} = f_x(\eta_{i+1}^{(1)}), \quad (8.24)$$

verwenden.

Zuweilen bietet sich die folgende Darstellung des Newton-Verfahrens und damit eine Zusammenstellung der genannten Fälle an.

Sei $\psi(z) = z - h f(t, z)$, sodass $\psi(z) - y = \varphi(z)$. Die Lösung von $\varphi(z) = 0$ ist $z^* = \eta_{i+1}$, d.h. es gilt dann $\psi(\eta_{i+1}) = \eta_i$.

Verschiedene Notationen des Newton-Verfahrens sind

$$\varphi'(\eta_{i+1}^{(k)}) \underbrace{(\eta_{i+1}^{(k+1)} - \eta_{i+1}^{(k)})}_{\text{Newton-Korrektur}} = \underbrace{-\varphi(\eta_{i+1}^{(k)})}_{\text{Newton-Defekt}} \approx 0,$$

$$\psi'(\eta_{i+1}^{(k)}) (\eta_{i+1}^{(k+1)} - \eta_{i+1}^{(k)}) = -(\psi(\eta_{i+1}^{(k)}) - \eta_i), \quad (8.25)$$

$$(1 - h f_x(\eta_{i+1}^{(k)})) (\eta_{i+1}^{(k+1)} - \eta_{i+1}^{(k)}) = -(\eta_{i+1}^{(k)} - h f(\eta_{i+1}^{(k)}) - \eta_i), \quad \eta_{i+1}^{(0)} = \eta_i,$$

$$(1 - h A) \Delta \eta_{i+1}^{(k)} = -(\eta_{i+1}^{(k)} - h f(\eta_{i+1}^{(k)}) - \eta_i),$$

wobei

$$A = \begin{cases} f_x(\eta_{i+1}^{(k)}), & \text{Newton-Verfahren (8.16),} \\ f_x(\eta_{i+1}^{(0)}), & \text{Version (8.23),} \\ f_x(\eta_{i+1}^{(0)}) & \text{bei } k = 0, \quad f_x(\eta_{i+1}^{(1)}) & \text{bei } k \geq 1, \quad \text{Version (8.24),} \\ f_x(\eta_{i+1}^{(m)}) & \text{bei jedem } m\text{-ten Schritt wird } f_x(\eta_{i+1}^{(k)}) & \text{neu berechnet,} \\ \dots, & \text{sonst.} \end{cases}$$

Falls in $x' = f(x)$ die rechte Seite eine lineare Funktion, d.h. eine Matrix ist ($x' = Ax + b$), dann hat man die Jacobi-Matrix $\mathbf{f}_x = A$ und braucht für alle Schritte nur ihre einmalige LU -Zerlegung von A .

Die beiden impliziten Verfahren IEV (8.7) und IHV (8.8) kann man wie bei den expliziten ESV auch mittels der Zuwächse k_i definieren.

$$\begin{aligned} \eta_{i+1} &= \eta_i + h f(t_{i+1}, \eta_{i+1}) \\ \text{bzw.} & \end{aligned} \tag{8.26}$$

$$\eta_{i+1} = \eta_i + h k_1, \quad k_1 = f(t_i + h, \eta_i + h k_1),$$

$$\begin{aligned} \eta_{i+1} &= \eta_i + \frac{h}{2} (f(t_i, \eta_i) + f(t_{i+1}, \eta_{i+1})) \\ \text{bzw.} & \end{aligned} \tag{8.27}$$

$$\eta_{i+1} = \eta_i + \frac{h}{2} (k_1 + k_2), \quad k_1 = f(t_i, \eta_i), \quad k_2 = f(t_i + h, \eta_i + \frac{h}{2}(k_1 + k_2)).$$

Die iterative Berechnung von η_{i+1} wird damit verlagert auf die Lösung einer Gleichung bzw. eines Systems für die Zuwächse k_i , natürlich genauso mit einem Iterationsverfahren. Das LIEV entsteht wiederum aus der abgebrochenen Taylor-Reihenentwicklung zum IEV gemäß

$$\begin{aligned} k_1 &= f(t_i + \alpha_1 h, \eta_i + h \beta_{11} k_1), \quad \alpha_1 = \beta_{11} = 1 \\ &= f(t_i + h, \eta_i + h k_1), \quad f(t, x) = f(x) \\ &= f(\eta_i + h k_1) \\ &= f(\eta_i) + h f_x(\eta_i) k_1 + \mathcal{O}(h^2), \\ k_1 &\doteq f(\eta_i) + h f_x(\eta_i) k_1, \\ (1 - h f_x(\eta_i)) k_1 &\doteq f(\eta_i), \\ k_1 &\doteq (1 - h f_x(\eta_i))^{-1} f(\eta_i), \\ \eta_{i+1} &= \eta_i + h (1 - h f_x(\eta_i))^{-1} f(\eta_i), \\ \frac{1}{h} (\eta_{i+1} - \eta_i) &= (1 - h f_x(\eta_i))^{-1} f(\eta_i). \end{aligned}$$

Das Weglassen der Glieder der Taylor-Reihe der Ordnung $\mathcal{O}(h^2)$ und kleiner deutet auf die Konsistenzordnung 1 des Verfahrens hin.

Präziser macht man die Bestimmung der Ordnung vom LIEV mittels des lokalen Diskretisierungsfehler (5.10), (5.35), indem man in das Diskretisierungsverfahren anstelle von η_i den exakten Lösungswert $x(t_i)$ einsetzt. Das ergibt mit $x' = f(x)$, $x'' = f_x(x)x' = f_x(x)f(x)$ usw. die Abschätzung

$$\begin{aligned}
 \tau_{i+1} &= \tau(t_i, \eta_i, h) = \frac{1}{h} (\Phi^{t_i+h, t_i} \eta_i - \eta_{i+1}) \\
 &= \frac{1}{h} (\Phi^{t_i+h, t_i} \eta_i - \eta_i) - \Psi(t_i, \eta_i, h) \\
 &= \frac{1}{h} (x_{i+1} - x_i) - \Psi(t_i, x_i, h), \quad x_i = x(t_i) \\
 &= \frac{1}{h} (x_{i+1} - x_i) - (1 - h f_x(x_i))^{-1} f(x_i) \\
 &= \frac{1}{h} (x_i + hx'_i + \frac{h^2}{2} x''(\xi) - x_i) - (1 - h f_x(x_i))^{-1} f(x_i) \\
 &= x'_i + h \int_0^1 (1 - \theta) x''(t_i + \theta h) d\theta - (1 - h f_x(x_i))^{-1} f(x_i) \\
 &= -(1 - h f_x(x_i))^{-1} [-(1 - h f_x(x_i)) x'_i + x'_i] + h \int_0^1 (1 - \theta) x''(t_i + \theta h) d\theta \\
 &= -h (1 - h f_x(x_i))^{-1} f_x(x_i) x'_i + h \int_0^1 (1 - \theta) x''(t_i + \theta h) d\theta, \\
 &\quad \text{sei } h \leq h_0, \quad |f| \leq M_0, \quad |f_x| \leq M_1, \quad |x''| \leq M_2, \\
 &\quad |(1 - h f_x(x_i))^{-1}| \leq \frac{1}{2} \quad (\text{Störungslemma}),
 \end{aligned}$$

$$|\tau_{i+1}| \leq Ch.$$

Stabilitätsuntersuchungen erfolgen mittels der Test-DGI $x' = \lambda x$, $\lambda < 0$.

Mit $f_x = \lambda$ hat das LIEV die einfache Gestalt

$$\eta_{i+1} = \frac{1}{1 - \lambda h} \eta_i,$$

sodass wie beim IEV die A-Stabilität folgt.

Damit ist auch die Konvergenzordnung des Verfahrens 1.

Den Aufwand bestimmt man nach der Anzahl der Funktionswertberechnungen. Dabei muss man auch die partielle(n) Ableitung(en) exakt oder näherungsweise bestimmen. Für SysDGI ist dann noch das Gleichungssystem zu lösen.

8.3 Parameterschemata

Ähnlich wie bei expliziten ESV kann man auch für implizite wie in Abschnitt 5.2.1 Parameterschemata aufstellen. Ausgangspunkt sind Notationen wie im Verfahren (8.2).

Der allgemeinste Ansatz für die Zuwächse ist

$$\begin{aligned}
 k_1 &= f(t_i + \alpha_1 h, \eta_i + h(\beta_{11}k_1 + \beta_{12}k_2 + \dots + \beta_{1r}k_r)), \\
 k_2 &= f(t_i + \alpha_2 h, \eta_i + h(\beta_{21}k_1 + \beta_{22}k_2 + \dots + \beta_{2r}k_r)), \\
 &\dots \\
 k_r &= f(t_i + \alpha_r h, \eta_i + h(\beta_{r1}k_1 + \beta_{r2}k_2 + \dots + \beta_{rr}k_r)).
 \end{aligned} \tag{8.28}$$

Damit folgt die allgemeine Form des Parameterschemas (Butcher-Schema) eines r -stufigen impliziten ESV der Ordnung p .

(r, p) -Schema

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \quad \begin{array}{c|c} \alpha & \beta \\ \hline & \gamma \end{array} \quad \begin{array}{c|cccccc} \alpha_1 & \beta_{11} & \beta_{12} & \beta_{13} & \dots & \beta_{1r} \\ \alpha_2 & \beta_{21} & \beta_{22} & \beta_{23} & \dots & \beta_{2r} \\ \alpha_3 & \beta_{31} & \beta_{32} & \beta_{33} & \dots & \beta_{3r} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \alpha_r & \beta_{r1} & \beta_{r2} & \beta_{r3} & \dots & \beta_{rr} \\ \hline 1 & \gamma_1 & \gamma_2 & \gamma_3 & \dots & \gamma_r \end{array} \quad (8.29)$$

So gibt es prinzipiell zwei Darstellungsformen des r -stufigen impliziten ESV. Zunächst kann man für einen Schritt jeweils eine direkte Darstellung ohne die Einbeziehung der Anstiege k_1, k_2, \dots notieren. Diese sei

$$\eta_{i+1} = G(t_i, \eta_i, \eta_{i+1}, h) = \eta_i + g(t_i, \eta_i, \eta_{i+1}, h) = \eta_i + h f(t_i, \eta_i, \eta_{i+1}, h). \quad (8.30)$$

Aus dem Parameterschema ergibt sich die zweite gebräuchlichere Form mit der Verwendung der Anstiege.

$$\begin{aligned} k_j &= f\left(t_i + \alpha_j h, \eta_i + h \sum_{l=1}^r \beta_{jl} k_l\right), \quad j = 1, 2, \dots, r, \\ \eta_{i+1} &= \eta_i + h \Psi(t_i, \eta_i, h) = \eta_i + h \sum_{j=1}^r \gamma_j k_j. \end{aligned} \quad (8.31)$$

Für einfache implizite ESV sind meist beide Versionen angebar und ineinander überführbar. Man hat aber auch ESV, wo nur eine der beiden Formeln existiert.

In den Darstellungen wird deutlich, dass entweder für die Bestimmung der Näherungswerte η_{i+1} bzw. der Anstiege k_1, k_2, \dots, k_r eine innere Iteration zu implementieren ist. Dabei sind geeignete Anfangsgrößen der Iteration zu wählen.

Definition 8.1 Ein (r, p) -RKV (8.31) heißt

- (a) *explizit*, falls $\alpha_1 = 0$, $\beta_{ij} = 0$ für $i \leq j$,
- (b) *diagonal-implizit (DIRKV)*, falls $\beta_{ij} = 0$ für $i \neq j$, aber mindestens ein $\beta_{ii} \neq 0$ für ein i ,
- (c) *voll-implizit oder einfach implizit (IRKV)*, falls mindestens ein $\beta_{ij} \neq 0$ für ein Paar i, j mit $i < j$,
- (c) *semi-implizit (SIRKV)*, falls $\beta_{ij} = 0$ für $i < j$, aber mindestens ein $\beta_{ii} \neq 0$ und mindestens ein $\beta_{ij} \neq 0$ für ein Paar i, j mit $i > j$, existiert.

Für spezielle Klassen bzw. besondere rechte Seiten der gDGL, z.B. $x' = f(x)$, reduziert sich das Parameterschema auf die Notation von zwei Vektoren.

Bei IRKV ist pro Schritt i. Allg. ein nichtlineares Gleichungssystem mit den r Unbekannten k_1, k_2, \dots, k_r iterativ zu lösen.

Für DIRKV und SIRKV lässt sich die innere Iteration etwas vereinfachen, da eine sukzessive Lösung der r aufeinander folgenden Gleichungen möglich ist.

$$\begin{aligned} k_1 - f(t_i + \alpha_1 h, \eta_i + h \beta_{11} k_1) &= 0 \rightarrow k_1, \\ k_2 - f(t_i + \alpha_2 h, \eta_i + h \beta_{21} k_1 + h \beta_{22} k_2) &= 0 \rightarrow k_2, \quad k_1 \text{ von vorher,} \\ &\dots \\ k_r - f\left(t_i + \alpha_r h, \eta_i + h \sum_{l=1}^{r-1} \beta_{rl} k_l + h \beta_{rr} k_r\right) &= 0 \rightarrow k_r, \quad k_1, k_2, \dots, k_{r-1} \text{ von vorher.} \end{aligned}$$

Unter Verwendung des Newton-Verfahrens für die j -te Gleichung

$$g(k_j) = k_j - f\left(t_i + \alpha_j h, \eta_i + h \sum_{l=1}^{j-1} \beta_{jl} k_l + h \beta_{jj} k_j\right) = 0$$

erhält man die Iterationsfolge

$$k_j^{(m+1)} = k_j^{(m)} - [g'(k_j^{(m)})]^{-1} g(k_j^{(m)}), \quad m = 0, 1, 2, \dots, \quad k_j^{(0)} \text{ gegeben.}$$

Bei SysgDGL der Dimension n hat man vektorielle Notationen für die Koeffizienten, Systeme mit $r \cdot n$ Unbekannten sowie die Jacobi-Matrix der rechten Seite \mathbf{f} , also der partiellen Ableitungen von $\mathbf{f}(t, \mathbf{x}) = (f_1(t, x_1, \dots, x_n), \dots, f_n(t, x_1, \dots, x_n))^T$ gemäß

$$\mathcal{J}(\mathbf{f}(\mathbf{x})) = \mathbf{f}'(\mathbf{x}) = \left(\frac{\partial f_i}{\partial x_j} \right) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}.$$

Ist \mathbf{f} eine lineare Funktion der Variablen \mathbf{x} , kann man die Nullstellengleichung $\mathbf{g}(k_j) = 0$ einfach nach dem Vektor k_j umstellen.

Ausgewählte Parameterschemata für implizite ESV

Man vergleiche die Darstellungen jeweils mit den expliziten ESV aus Abschnitt 5.2.1.

(a) (1,1)-IEV, Gear(1), Radau-II-A(1), Genauigkeit $\mathcal{O}(h)$

Direkte Formel

$$\eta_{i+1} = \eta_i + h f(t_{i+1}, \eta_{i+1}) \quad (8.32)$$

Kurzform

$$\begin{aligned} k_1 &= f(t_{i+1}, \eta_{i+1}) = f(t_i + h, \eta_i + h k_1) \\ \eta_{i+1} &= \eta_i + h k_1 \end{aligned} \quad \left| \begin{array}{c} 1 \\ \hline 1 \end{array} \right.$$

Langform

$$\begin{aligned} k_1 &= f(t_i, \eta_i) && \left| \begin{array}{ccc} 0 & 0 & 0 \\ \hline 1 & 0 & 1 \end{array} \right. \\ k_2 &= f(t_{i+1}, \eta_{i+1}) = f(t_{i+1}, \eta_i + h k_2) \\ \eta_{i+1} &= \eta_i + h k_2 && \left| \begin{array}{ccc} \hline & 0 & 1 \end{array} \right. \end{aligned}$$

Wachstumsfunktion $F(z) = \frac{1}{1-z}$, $\lim_{\Re z \rightarrow -\infty} F(z) = 0$, L-stabil

- (b) (1,2)-Gauß-Legendre-Verfahren, Gauß-Formel, implizite Mittelpunktsregel,
(1,2)-IRKV, implizites Collatz-Verfahren, Genauigkeit $\mathcal{O}(h^2)$

Direkte Formel

Die Idee dafür entnehmen wir dem expliziten Collatz-Verfahren (MPZV).

$$\begin{aligned} \eta_{i+1} &= \eta_i + h \underbrace{f\left(t_i + \frac{h}{2}, \frac{1}{2}(\eta_i + \eta_{i+1})\right)}_{=k_1, \text{ d.h. } \eta_{i+1} = \eta_i + h k_1} \\ &= \eta_i + h f\left(t_i + \frac{h}{2}, \frac{1}{2}(\eta_i + \eta_i + h k_1)\right) \\ &= \eta_i + h \underbrace{f\left(t_i + \frac{h}{2}, \eta_i + \frac{h}{2} k_1\right)}_{=k_1}. \end{aligned} \quad (8.33)$$

Damit ist auch die implizite Formel für den Anstieg k_1 bestätigt.

Kurzform

$$\begin{array}{l} k_1 = f\left(t_i + \frac{h}{2}, \eta_i + \frac{h}{2} k_1\right) \\ \eta_{i+1} = \eta_i + h k_1 \end{array} \quad \left. \begin{array}{l} \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{array} \right| \begin{array}{l} \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{array}$$

Langform

$$\begin{array}{l} k_1 = f(t_i, \eta_i) \\ k_2 = f\left(t_i + \frac{h}{2}, \eta_i + \frac{h}{2} k_2\right) \\ \eta_{i+1} = \eta_i + h k_2 \end{array} \quad \left. \begin{array}{l} 0 \\ \frac{1}{2} \\ 0 \end{array} \right| \begin{array}{l} 0 \\ 0 \\ 1 \end{array}$$

Wachstumsfunktion $F(z) = \frac{1+z/2}{1-z/2}$, $\lim_{\Re z \rightarrow -\infty} F(z) = -1$, A-stabil

- (c) (2,2)-IRKV, Sehnentrapezregel, Genauigkeit $\mathcal{O}(h^2)$

Direkte Formel

$$\begin{aligned} \eta_{i+1} &= \eta_i + \frac{h}{2} [f(t_i, \eta_i) + f(t_i + h, \eta_i + h f(t_i, \eta_i)) + 2\zeta_2 (\eta_{i+1} - \eta_i - h f(t_i, \eta_i))], \\ \zeta_{1,2} &\geq 0, \quad \zeta_1 + \zeta_2 = 1. \end{aligned} \quad (8.34)$$

Die Idee dafür korrespondiert durchaus mit dem expliziten HV, erfordert jedoch noch eine kurze Begründung. Man geht von der nachfolgenden impliziten Kurzform mit den Anstiegen k_1, k_2 aus und macht einige Umformungen.

$$\begin{aligned} \eta_{i+1} &= \eta_i + \frac{h}{2} \{k_1 + k_2\}, \\ k_1 &= f(t_i, \eta_i), \quad k_2 = f(t_i + h, \eta_i + h [\zeta_1 k_1 + \zeta_2 k_2]), \quad \zeta_1 + \zeta_2 = 1, \\ &= \eta_i + \frac{h}{2} \{k_1 + f(t_i + h, \eta_i + h [\zeta_1 k_1 + \zeta_2 k_2])\}, \quad k_2 = \frac{2}{h} (\eta_{i+1} - \eta_i) - k_1 \\ &= \eta_i + \frac{h}{2} \{k_1 + f(t_i + h, \eta_i + h [\zeta_1 k_1 + \zeta_2 (\frac{2}{h} (\eta_{i+1} - \eta_i) - k_1)])\} \\ &= \eta_i + \frac{h}{2} \{k_1 + f(t_i + h, \eta_i + h \zeta_1 k_1 + 2\zeta_2 (\eta_{i+1} - \eta_i) - h \zeta_2 k_1)\}, \quad \zeta_1 = 1 - \zeta_2 \\ &= \eta_i + \frac{h}{2} \{k_1 + f(t_i + h, \eta_i + h k_1 + 2\zeta_2 (\eta_{i+1} - \eta_i) - 2h \zeta_2 k_1)\} \\ &= \eta_i + \frac{h}{2} \{k_1 + f(t_i + h, \eta_i + h k_1 + 2\zeta_2 (\eta_{i+1} - \eta_i - h k_1))\} \\ &= \eta_i + \frac{h}{2} \{k_1 + f(t_i + h, \eta_i + h k_1 + 2h \zeta_2 (\frac{\eta_{i+1} - \eta_i}{h} - k_1))\}. \end{aligned}$$

$\underbrace{\hspace{15em}}_{=k_2 \text{ von der Kurzform}}$

Man hat zusätzlich die Beziehungen

$$k_2 = f(t_i + h, \eta_i + h(\zeta_1 k_1 + \zeta_2 k_2)) = f(t_i + h, \eta_i + h k_1 + \underbrace{h \zeta_2 (k_2 - k_1)}),$$

$$h \zeta_2 (k_2 - k_1) = 2h \zeta_2 \left(\frac{\eta_{i+1} - \eta_i}{h} - k_1 \right).$$

Kurzform

$$\begin{array}{l} k_1 = f(t_i, \eta_i) \\ k_2 = f(t_i + h, \eta_i + h(\zeta_1 k_1 + \zeta_2 k_2)) \\ \eta_{i+1} = \eta_i + \frac{h}{2}(k_1 + k_2) \end{array} \quad \left| \begin{array}{l} 0 \\ 1 \\ \hline \frac{1}{2} \end{array} \right| \begin{array}{l} 0 \\ \zeta_1 \\ \frac{1}{2} \end{array} \quad \begin{array}{l} 0 \\ \zeta_2 \\ \frac{1}{2} \end{array}$$

Betrachten wir einige Sonderfälle des (2,2)-IRKV.

(1) $\zeta_1 = 1, \zeta_2 = 0$: IRKV wird zum expliziten HV

(2) $\zeta_1 = \zeta_2 = \frac{1}{2}$: (2,2)-IRKV1, IHV

Direkte Formel

$$\eta_{i+1} = \eta_i + \frac{h}{2} [f(t_i, \eta_i) + f(t_i + h, \eta_{i+1})] \quad (8.35)$$

Kurzform

$$\begin{array}{l} k_1 = f(t_i, \eta_i) \\ k_2 = f(t_i + h, \eta_i + \frac{h}{2}(k_1 + k_2)) \\ \eta_{i+1} = \eta_i + \frac{h}{2}(k_1 + k_2) \end{array} \quad \left| \begin{array}{l} 0 \\ 1 \\ \hline \frac{1}{2} \end{array} \right| \begin{array}{l} 0 \\ \frac{1}{2} \\ \frac{1}{2} \end{array}$$

Wachstumsfunktion

$$F(z) = \frac{1 + z/2}{1 - z/2}, \quad \lim_{\Re z \rightarrow -\infty} F(z) = -1, \quad \text{A-stabil}$$

(3) $\zeta_1 = 0, \zeta_2 = 1$: (2,2)-IRKV

Direkte Formel

$$\begin{aligned} \eta_{i+1} &= \eta_i + \frac{h}{2} [f(t_i, \eta_i) + f(t_i + h, 2\eta_{i+1} - \eta_i - hf(t_i, \eta_i))] \\ &= \eta_i + \frac{h}{2} [f(t_i, \eta_i) + f(t_i + h, \eta_{i+1} + \underbrace{\eta_{i+1} - \eta_i - hf(t_i, \eta_i)}_{\approx 0})] \end{aligned} \quad (8.36)$$

Kurzform

$$\begin{array}{l} k_1 = f(t_i, \eta_i) \\ k_2 = f(t_i + h, \eta_i + h k_2) \\ \eta_{i+1} = \eta_i + \frac{h}{2}(k_1 + k_2) \end{array} \quad \left| \begin{array}{l} 0 \\ 1 \\ \hline \frac{1}{2} \end{array} \right| \begin{array}{l} 0 \\ 0 \\ \frac{1}{2} \end{array} \quad \begin{array}{l} 0 \\ 1 \\ \frac{1}{2} \end{array}$$

Wachstumsfunktion

$$F(z) = \frac{1 - z^2/2}{1 - z}, \quad \text{reelles Stabilitätsintervall } z \in (-1 - \sqrt{5}, 0)$$

(d) Die bisherigen IRKV lassen sich vergleichen bzw. einbetten in die parametrischen ESV mit einem oder mehreren Parametern.

Folgende drei Fälle von (r, p) -IRKV mit Parametern in ihrer direkten Notation sollen betrachtet werden.

θ -Verfahren

$$\eta_{i+1} = \eta_i + h [(1 - \theta) f(t_i, \eta_i) + \theta f(t_{i+1}, \eta_{i+1})], \quad \theta \geq 0 \quad (8.37)$$

One-leg-Verfahren, implizites Collatz-Verfahren

$$\eta_{i+1} = \eta_i + h f((1 - \theta)t_i + \theta t_{i+1}, (1 - \theta)\eta_i + \theta \eta_{i+1}), \quad \theta \geq 0 \quad (8.38)$$

$\theta_{1,2}$ -Verfahren

$$\eta_{i+1} = \eta_i + h [(1 - \theta_1) f(t_i, \eta_i) + \theta_1 f(t_i + \theta_2 h, (1 - \theta_2)\eta_i + \theta_2 \eta_{i+1})], \quad \theta_{1,2} \geq 0 \quad (8.39)$$

explizite ESV	
PZV	θ -Verfahren, $\theta = 0$
Collatz-V., MPZV	ähnlich zu One-leg-Verfahren, $\theta = \frac{1}{2}$
HV, VPZV	ähnlich zu θ -Verfahren, $\theta = \frac{1}{2}$ bzw. ähnlich zu $\theta_{1,2}$ -Verfahren, $\theta_1 = \frac{1}{2}$, $\theta_2 = 1$
implizite ESV	
IEV	θ -Verfahren, $\theta = 1$
(1,2)-IRKV	One-leg-Verfahren, $\theta = \frac{1}{2}$
(2,2)-IRKV1, IHV	θ -Verfahren, $\theta = \frac{1}{2}$ bzw. $\theta_{1,2}$ -Verfahren, $\theta_1 = \frac{1}{2}$, $\theta_2 = 1$
(2,2)-IRKV	ähnlich zu θ -Verfahren, $\theta = \frac{1}{2}$

Tab. 8.2 Implizite RKV im Vergleich mit den parametrischen ESV mit einem oder mehreren Parametern

Nachweis, dass das One-leg-Verfahren mit $\theta = \frac{1}{2}$ sowie (1,2)-IRKV übereinstimmen.

$$\begin{aligned} \eta_{i+1} &= \eta_i + h f\left(\frac{1}{2}t_i + \frac{1}{2}t_{i+1}, \frac{1}{2}\eta_i + \frac{1}{2}\eta_{i+1}\right) \\ &= \eta_i + h f\left(t_i + \frac{1}{2}(t_{i+1} - t_i), \frac{1}{2}\eta_i + \frac{1}{2}(\eta_i + hk_1)\right) \\ &= \eta_i + h f\left(t_i + \frac{h}{2}, \eta_i + \frac{h}{2}k_1\right) \\ &= \eta_i + h k_1. \end{aligned}$$

Nachweis, dass das θ -Verfahren mit $\theta = \frac{1}{2}$ sowie (2,2)-IRKV1 übereinstimmen.

$$\begin{aligned} \eta_{i+1} &= \eta_i + \frac{h}{2} [f(t_i, \eta_i) + f(t_i + h, \eta_{i+1})] \\ &= \eta_i + \frac{h}{2} [f(t_i, \eta_i) + f(t_i + h, \eta_i + \frac{h}{2}(k_1 + k_2))] \\ &= \eta_i + \frac{h}{2} [k_1 + k_2], \quad k_1 = f(t_i, \eta_i), \quad k_2 = f(t_i + h, \eta_i + \frac{h}{2}(k_1 + k_2)). \end{aligned}$$

(e) Die drei Fälle von (r, p) -IRKV mit Parametern (8.37) – (8.39) werden in ihren verschiedenen Formen notiert.

θ -Verfahren, Genauigkeit $\mathcal{O}(h^2)$ für $\theta = \frac{1}{2}$

Direkte Formel

$$\eta_{i+1} = \eta_i + h [(1 - \theta) f(t_i, \eta_i) + \theta f(t_{i+1}, \eta_{i+1})], \quad \theta \geq 0$$

Kurzform

$$\begin{array}{l} k_1 = f(t_{i+1}, \eta_{i+1}) = f(t_i + h, \eta_i + h[(1 - \theta)f(t_i, \eta_i) + \theta k_1]) \\ \eta_{i+1} = \eta_i + h [(1 - \theta)f(t_i, \eta_i) + \theta k_1] \end{array} \quad \frac{1 \mid 1 - \theta \quad \theta}{\mid 1 - \theta \quad \theta}$$

Langform

$$\begin{array}{l} k_1 = f(t_i, \eta_i) \\ k_2 = f(t_{i+1}, \eta_{i+1}) = f(t_i + h, \eta_i + h [(1 - \theta)k_1 + \theta k_2]) \\ \eta_{i+1} = \eta_i + h [(1 - \theta)k_1 + \theta k_2] \end{array} \quad \frac{0 \mid 0 \quad 0}{1 \mid 1 - \theta \quad \theta} \\ \frac{\mid 1 - \theta \quad \theta}{\mid 1 - \theta \quad \theta}$$

One-leg-Verfahren, implizites Collatz-Verfahren, Genauigkeit $\mathcal{O}(h^2)$ für $\theta = \frac{1}{2}$

Direkte Formel

$$\eta_{i+1} = \eta_i + h f((1 - \theta)t_i + \theta t_{i+1}, (1 - \theta)\eta_i + \theta \eta_{i+1}), \quad \theta \geq 0$$

Kurzform

$$\begin{array}{l} k_1 = f((1 - \theta)t_i + \theta t_{i+1}, \eta_i + \theta h k_1) = f(t_i + \theta h, \eta_i + \theta h k_1) \\ \eta_{i+1} = \eta_i + h k_1 \end{array} \quad \frac{\theta \mid \theta}{\mid 1}$$

$\theta_{1,2}$ -Verfahren, Genauigkeit mindestens $\mathcal{O}(h)$

Direkte Formel

$$\eta_{i+1} = \eta_i + h [(1 - \theta_1) f(t_i, \eta_i) + \theta_1 f(t_i + \theta_2 h, (1 - \theta_2)\eta_i + \theta_2 \eta_{i+1})], \quad \theta_{1,2} \geq 0$$

Langform

$$\begin{array}{l} k_1 = f(t_i, \eta_i) \\ k_2 = f(t_i + \theta_2 h, (1 - \theta_2)\eta_i + \theta_2 \eta_{i+1}) \\ \quad = f(t_i + \theta_2 h, (1 - \theta_2)\eta_i + \theta_2 (\eta_i + h [(1 - \theta_1) k_1 + \theta_1 k_2])) \\ \quad = f(t_i + \theta_2 h, \eta_i + h \theta_2 [(1 - \theta_1) k_1 + \theta_1 k_2]) \\ \quad = f(t_i + \theta_2 h, \eta_i + h [\theta_2 (1 - \theta_1) k_1 + \theta_2 \theta_1 k_2]) \\ \eta_{i+1} = \eta_i + h [(1 - \theta_1) k_1 + \theta_1 k_2] \end{array} \quad \frac{0 \mid 0 \quad 0}{\theta_2 \mid \theta_2 (1 - \theta_1) \quad \theta_2 \theta_1} \\ \frac{\mid 1 - \theta_1 \quad \theta_1}{\mid 1 - \theta_1 \quad \theta_1}$$

Achtung

Die Konsistenzbedingung $\theta_2 = \theta_2(1 - \theta_1) + \theta_2 \theta_1$ in der zweiten Zeile des Parameterschemas ist erfüllt. Bei $\theta_2 = 1$ erhält man mit $\theta_1 = \theta$ das bekannte θ -Verfahren.

(f) Zweistufiges implizites RKV, (2,3)-IRKV, Genauigkeit $\mathcal{O}(h^3)$

Kurzform

$$\begin{array}{l} k_1 = f\left(t_i + \frac{3+\sqrt{3}}{6}h, \eta_i + \frac{3+\sqrt{3}}{6}hk_1\right) \\ k_2 = f\left(t_i + \frac{3-\sqrt{3}}{6}h, \eta_i - \frac{\sqrt{3}}{3}hk_1 + \frac{3+\sqrt{3}}{6}hk_2\right) \\ \eta_{i+1} = \eta_i + \frac{h}{2}(k_1 + k_2) \end{array} \quad \begin{array}{c|cc} \frac{3+\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} & 0 \\ \frac{3-\sqrt{3}}{6} & -\frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad (8.40)$$

Mit $f(t, x) = \lambda x$ folgt für k_1 und k_2 nacheinander berechnet

$$k_1 = \frac{1}{1 - \frac{3+\sqrt{3}}{6}h\lambda} \lambda \eta_i, \quad k_2 = \frac{1 - \frac{\sqrt{1+\sqrt{3}}}{2}h\lambda}{\left(1 - \frac{3+\sqrt{3}}{6}h\lambda\right)^2} \lambda \eta_i.$$

Wachstumsfunktion

$$F(z) = \frac{1 - (2a - 1)z - ((1 + \sqrt{3})/4 + a/2 - a^2)z^2}{(1 - az)^2}, \quad a = \frac{3 + \sqrt{3}}{6}$$

$$\lim_{\Re z \rightarrow -\infty} F(z) = -\frac{(1 + \sqrt{3})/4 + a/2 - a^2}{a^2} = -0.7320508068, \quad \text{A-stabil}$$

(g) Zweistufiges implizites RKV, (2,4)-IRKV, Gauß-Verfahren, Genauigkeit $\mathcal{O}(h^4)$

Kurzform

$$\begin{array}{l} k_1 = f\left(t_i + \frac{3-\sqrt{3}}{6}h, \eta_i + \frac{1}{4}hk_1 + \frac{3-2\sqrt{3}}{12}hk_2\right) \\ k_2 = f\left(t_i + \frac{3+\sqrt{3}}{6}h, \eta_i + \frac{3+2\sqrt{3}}{12}hk_1 + \frac{1}{4}hk_2\right) \\ \eta_{i+1} = \eta_i + \frac{h}{2}(k_1 + k_2) \end{array} \quad \begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \frac{3+\sqrt{3}}{6} & \frac{3+2\sqrt{3}}{12} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad (8.41)$$

Bei $r = 2$ Stufen ist die maximal erreichbare Genauigkeitsordnung $p = 4$.

Mit $f(t, x) = \lambda x$ folgt ein lineares Gleichungssystem für k_1 und k_2 mit der Lösung

$$k_1 = \frac{1 - \frac{\sqrt{3}}{6}h\lambda}{1 - \frac{1}{2}h\lambda + \frac{1}{12}(h\lambda)^2} \lambda \eta_i, \quad k_2 = \frac{1 + \frac{\sqrt{3}}{6}h\lambda}{1 - \frac{1}{2}h\lambda + \frac{1}{12}(h\lambda)^2} \lambda \eta_i.$$

Wachstumsfunktion

$$F(z) = \frac{1 + \frac{z}{2} + \frac{z^2}{12}}{1 - \frac{z}{2} + \frac{z^2}{12}}$$

$$\lim_{\Re z \rightarrow -\infty} F(z) = 1, \quad \text{A-stabil}$$

(h) Zweistufiges implizites Radau-IA-Verfahren, Genauigkeit $\mathcal{O}(h^3)$

Kurzform

$$\begin{array}{l|ll} k_1 = f(t_i, \eta_i + \frac{1}{4}hk_1 - \frac{1}{4}hk_2) & 0 & \frac{1}{4} \quad -\frac{1}{4} \\ k_2 = f(t_i + \frac{2}{3}h, \eta_i + \frac{1}{4}hk_1 + \frac{5}{12}hk_2) & \frac{2}{3} & \frac{1}{4} \quad \frac{5}{12} \\ \eta_{i+1} = \eta_i + \frac{h}{4}(k_1 + 3k_2) & \frac{1}{4} & \frac{3}{4} \end{array} \quad (8.42)$$

Wachstumsfunktion

$$F(z) = \frac{1 + \frac{z}{3}}{1 - \frac{2z}{3} + \frac{z^2}{6}}$$

$$\lim_{\Re z \rightarrow -\infty} F(z) = 0, \text{ L-stabil}$$

(i) Vierstufiges implizites Lobatto-IIIC-IA-Verfahren, Genauigkeit $\mathcal{O}(h^6)$

Kurzform

$$\begin{array}{l} k_1 = f(t_i, \eta_i + \frac{h}{12}(k_1 - \sqrt{5}k_2 + \sqrt{5}k_3 - k_4)) \\ k_2 = f(t_i + \frac{1}{10}(5 - \sqrt{5})h, \eta_i + \frac{h}{12}(k_1 + 3k_2 + (2 - \frac{7\sqrt{5}}{5})k_3 + \frac{\sqrt{5}}{5}k_4)) \\ k_3 = f(t_i + \frac{1}{10}(5 + \sqrt{5})h, \eta_i + \frac{h}{12}(k_1 + (2 + \frac{7\sqrt{5}}{5})k_2 + 3k_3 - \frac{\sqrt{5}}{5}k_4)) \\ k_4 = f(t_i + h, \eta_i + \frac{h}{12}(k_1 + 5k_2 + 5k_3 + k_4)) \\ \eta_{i+1} = \eta_i + \frac{h}{12}(k_1 + 5k_2 + 5k_3 + k_4) \end{array} \quad (8.43)$$

$$\begin{array}{l|llll} 0 & \frac{1}{12} & -\frac{\sqrt{5}}{12} & \frac{\sqrt{5}}{12} & -\frac{1}{12} \\ \frac{1}{2} - \frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{1}{4} & \frac{1}{6} - \frac{7\sqrt{5}}{60} & \frac{\sqrt{5}}{60} \\ \frac{1}{2} + \frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{1}{6} + \frac{7\sqrt{5}}{60} & \frac{1}{4} & -\frac{\sqrt{5}}{60} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

Wachstumsfunktion

$$F(z) = \frac{1 + \frac{z}{3} + \frac{z^2}{30}}{1 - \frac{2z}{3} + \frac{z^2}{5} - \frac{z^3}{30} + \frac{z^4}{360}}$$

$$\lim_{\Re z \rightarrow -\infty} F(z) = 0, \text{ L-stabil}$$

Wegen ihrer L-Stabilität sind implizite RKV, insbesondere das IEV und die erwähnten Radau- und Lobatto-Verfahren, besonders für die Langzeitintegration gedämpfter Schwingungssysteme (dissipative SysgDGI) geeignet. Ansonsten erweisen sich jedoch meist die A-stabilen Gauß-Legendre-Verfahren überlegen.

8.4 Implementation ausgewählter impliziter RKV

Die Programmierung der IRKV erfolgt in MATLAB als Funktionen unter Verwendung einer der angegebenen Darstellungen als direkte Formel, Kurzform oder Langform. Der Rahmen für die Funktion ist einheitlich gestaltet. Der Abbruch der inneren Iteration wird in jedem Schritt durch eine Toleranz $\varepsilon > 0$ gemäß $|k_j^{(m+1)} - k_j^{(m)}| < \varepsilon$ bzw. $\|k^{(m+1)} - k^{(m)}\|_\infty < \varepsilon$ gesteuert. Für die Iteration werden geeignete Startwerte vorgegeben.

(a) IEV, θ -Verfahren mit $\theta = 1$, Genauigkeit $\mathcal{O}(h)$

Iteration der Kurzform

$$\begin{aligned} k_1^{(m+1)} &= f(t_i + h, \eta_i + h k_1^{(m)}), \quad m = 0, 1, 2, \dots, \quad k_1^{(0)} = f(t_i, \eta_i) \\ \eta_{i+1} &= \eta_i + h k_1^{(M)} \end{aligned}$$

```
% euler_impl1.m
function [t,x] = euler_impl1(f,t0,t1,x0,N,eps)
h = (t1-t0)/N;
t(1) = t0;
x(1) = x0;
for i = 1:N
    t(i+1) = t0+h*i;
    k1 = feval(f,t(i),x(i));
    while 1 % innere Iteration
        k1n = feval(f,t(i+1),x(i)+h*k1);
        if abs(k1n-k1)<eps, break; end
        k1 = k1n;
    end
    x(i+1) = x(i)+h*k1n;
end
end
```

(b) (1,2)-IRKV, One-leg-Verfahren mit $\theta = \frac{1}{2}$, Genauigkeit $\mathcal{O}(h^2)$

Iteration der Kurzform

$$\begin{aligned} k_1^{(m+1)} &= f\left(t_i + \frac{h}{2}, \eta_i + \frac{h}{2} k_1^{(m)}\right), \quad m = 0, 1, 2, \dots, \quad k_1^{(0)} = f(t_i, \eta_i) \\ \eta_{i+1} &= \eta_i + h k_1^{(M)} \end{aligned}$$

```
% gauss_impl1.m
function [t,x] = gauss_impl1(f,t0,t1,x0,N,eps)
h = (t1-t0)/N;
t(1) = t0;
x(1) = x0;
for i = 1:N
    t(i+1) = t0+h*i;
    k1 = feval(f,t(i),x(i));
    while 1 % innere Iteration
        k1n = feval(f,t(i)+h/2,x(i)+h/2*k1);
        if abs(k1n-k1)<eps, break; end
    end
end
```

```

    k1 = k1n;
end
x(i+1) = x(i)+h*k1n;
end
end

```

(c) θ -Verfahren, Genauigkeit $\mathcal{O}(h) \cdot \mathcal{O}(h^2)$ abhängig von θ

Iteration der Langform

$$\begin{aligned}
 k_1 &= f(t_i, \eta_i) \\
 k_2^{(m+1)} &= f(t_{i+1}, \eta_{i+1}) = f(t_i + h, \eta_i + h[(1 - \theta)k_1 + \theta k_2^{(m)}]), \quad m = 0, 1, 2, \dots \\
 k_2^{(0)} &= k_1 \\
 \eta_{i+1} &= \eta_i + h[(1 - \theta)k_1 + \theta k_2^{(M)}]
 \end{aligned}$$

Sonderfall: (2,2)-IRKV1 mit $\zeta_{1,2} = \frac{1}{2}$, θ -Verfahren mit $\theta = \frac{1}{2}$, Genauigkeit $\mathcal{O}(h^2)$

```

% heun_impl1.m
% Theta-Verfahren, 0<=Theta<=1, Standardversion Theta=1/2, (2,2)-IRKV1
function [t,x] = heun_impl1(f,t0,t1,x0,N,eps,Theta)
h = (t1-t0)/N;
t(1) = t0;
x(1) = x0;
for i = 1:N
    t(i+1) = t0+h*i;
    k1 = feval(f,t(i),x(i));
    k2 = k1; % auch moeglich, etwas besser
            % k2 = feval(f,t(i+1),x(i)+h*k1);
    while 1 % innere Iteration
        k2n = feval(f,t(i+1),x(i)+h*((1-Theta)*k1+Theta*k2));
        if abs(k2n-k2)<eps, break; end
        k2 = k2n;
    end
    x(i+1) = x(i)+h*((1-Theta)*k1+Theta*k2n);
end
end

```

(d) (2,2)-IRKV, $\zeta_1 = 0$, $\zeta_2 = 1$

Iteration der Kurzform

$$\begin{aligned}
 k_1 &= f(t_i, \eta_i) \\
 k_2^{(m+1)} &= f(t_i + h, \eta_i + h k_2^{(m)}), \quad m = 0, 1, 2, \dots, \quad k_2^{(0)} = k_1 \\
 \eta_{i+1} &= \eta_i + \frac{h}{2}(k_1 + k_2^{(M)})
 \end{aligned}$$

```

% heun_impl2.m
% Implizites Heun-Verfahren, (2,2)-IRKV
% Parameterkonstellation: 2 verschiedene zeta-Werte
function [t,x] = heun_impl2(f,t0,t1,x0,N,eps)

```



```

h = (t1-t0)/N;
t(1) = t0;
x(1) = x0;
for i = 1:N
    t(i+1) = t0+h*i;
    k1 = feval(f,t(i),x(i));
    k2 = k1;           % auch moeglich, etwas besser
                    % k2 = feval(f,t(i+1),x(i)+h*k1);
    while 1 % innere Iteration
        k2n = feval(f,t(i+1),x(i)+h*k2);
        if abs(k2n-k2)<eps, break; end
        k2 = k2n;
    end
    x(i+1) = x(i)+h/2*(k1+k2n);
end
end

```

(e) One-leg-Verfahren, implizites Collatz-Verfahren, Genauigkeit $\mathcal{O}(h^2)$ für $\theta = \frac{1}{2}$
Iteration der direkten Formel

$$\eta_{i+1}^{(m+1)} = \eta_i + h f((1-\theta)t_i + \theta t_{i+1}, (1-\theta)\eta_i + \theta \eta_{i+1}^{(m)}), \quad m = 0, 1, 2, \dots$$

$$\eta_{i+1}^{(0)} = \eta_i + f(t_i, \eta_i)$$

Sonderfall: (1,2)-IRKV, $\theta = \frac{1}{2}$, Genauigkeit $\mathcal{O}(h^2)$

```

% collatz_impl1.m
% One-leg-Verfahren, 0<=Theta<=1, implizites Collatz-Verfahren
% (1,2)-IRKV, Gauss-Legendre-Formel, Theta=1/2
function [t,x] = collatz_impl1(f,t0,t1,x0,N,eps,Theta)
h = (t1-t0)/N;
t(1) = t0;
x(1) = x0;
for i = 1:N
    t(i+1) = t0+h*i;
    z = x(i)+h*feval(f,t(i),x(i));
    while 1 % innere Iteration
        zn = x(i)+h*feval(f,(1-Theta)*t(i)+Theta*t(i+1),...
                        (1-Theta)*x(i)+Theta*z);
        if abs(zn-z)<eps, break; end
        z = zn;
    end
    x(i+1) = zn;
end
end

```

(f) Zweistufiges implizites RKV, (2,3)-IRKV, Genauigkeit $\mathcal{O}(h^3)$

Iteration der Kurzform

$$k_1^{(m+1)} = f\left(t_i + \frac{3+\sqrt{3}}{6}h, \eta_i + \frac{3+\sqrt{3}}{6}hk_1^{(m)}\right), \quad m = 0, 1, 2, \dots$$

$$k_2^{(m+1)} = f\left(t_i + \frac{3-\sqrt{3}}{6}h, \eta_i - \frac{\sqrt{3}}{3}hk_1^{(m)} + \frac{3+\sqrt{3}}{6}hk_2^{(m)}\right)$$

$$k_1^{(0)} = f(t_i + c_1h, \eta_i + c_1hf(t_i, \eta_i)), \quad k_2^{(0)} = f(t_i + c_2h, \eta_i + c_2hf(t_i, \eta_i))$$

$$\eta_{i+1} = \eta_i + \frac{h}{2}(k_1^{(M)} + k_2^{(M)})$$

```

% rkv23_impl.m
% (2,3)-IRKV, Genauigkeitsordnung 3
function [t,x] = rkv23_impl(f,t0,t1,x0,N,eps)
h = (t1-t0)/N;
t(1) = t0;
x(1) = x0;
c1 = (3+sqrt(3))/6;
c2 = (3-sqrt(3))/6;
c3 = c2-c1;
for i = 1:N
    t(i+1) = t0+h*i;
    k0 = feval(f,t(i),x(i));
    k1 = feval(f,t(i)+c1*h,x(i)+h*c1*k0);
    k2 = feval(f,t(i)+c2*h,x(i)+h*c2*k0);
    while 1 % innere Iteration
        k1n = feval(f,t(i)+c1*h,x(i)+h*c1*k1);
        k2n = feval(f,t(i)+c2*h,x(i)+h*(c3*k1+c1*k2));
        if max(abs(k2n-k2),abs(k1n-k1))<eps, break; end
        k1 = k1n;
        k2 = k2n;
    end
    x(i+1) = x(i)+h/2*(k1n+k2n);
end
end

```

(g) Zweistufiges implizites RKV, (2,4)-IRKV, Gauß-Verfahren, Genauigkeit $\mathcal{O}(h^4)$

Iteration der Kurzform

$$\begin{aligned}
 k_1^{(m+1)} &= f\left(t_i + \frac{3-\sqrt{3}}{6}h, \eta_i + \frac{1}{4}hk_1^{(m)} + \frac{3-2\sqrt{3}}{12}hk_2^{(m)}\right), \quad m = 0, 1, 2, \dots \\
 k_2^{(m+1)} &= f\left(t_i + \frac{3+\sqrt{3}}{6}h, \eta_i + \frac{3+2\sqrt{3}}{12}hk_1^{(m)} + \frac{1}{4}hk_2^{(m)}\right) \\
 k_1^{(0)} &= f(t_i + c_1h, \eta_i + c_1hf(t_i, \eta_i)), \quad k_2^{(0)} = f(t_i + c_2h, \eta_i + c_2hf(t_i, \eta_i)) \\
 \eta_{i+1} &= \eta_i + \frac{h}{2}(k_1^{(M)} + k_2^{(M)})
 \end{aligned}$$

```

% rkv24_impl.m
% (2,4)-IRKV, Gauss-Formel, Genauigkeitsordnung 4
function [t,x] = rkv24_impl(f,t0,t1,x0,N,eps)
h = (t1-t0)/N;
t(1) = t0;
x(1) = x0;
c1 = (3-sqrt(3))/6;
c2 = (3+sqrt(3))/6;
c1s = (3-2*sqrt(3))/12;
c2s = (3+2*sqrt(3))/12;
for i = 1:N
    t(i+1) = t0+h*i;
    k0 = feval(f,t(i),x(i));
    k1 = feval(f,t(i)+c1*h,x(i)+h*c1*k0);
    k2 = feval(f,t(i)+c2*h,x(i)+h*c2*k0);
    while 1 % innere Iteration
        k1n = feval(f,t(i)+c1*h,x(i)+h*(0.25*k1+c1s*k2));
        k2n = feval(f,t(i)+c2*h,x(i)+h*(c2s*k1+0.25*k2));
    end
end

```

```

    if max(abs(k2n-k2),abs(k1n-k1))<eps, break; end
    k1 = k1n;
    k2 = k2n;
end
x(i+1) = x(i)+h/2*(k1n+k2n);
end
end

```

Konsistente und stabile implizite ESV liefern konvergente Näherungslösungen von AWP. Was passiert aber, wenn das AWP eine mehrdeutige Lösung besitzt?

Beispiel 8.2

Das mehrdeutige AWP aus Abschnitt 4.1

$$x'(t) = 2\sqrt{x(t)}, \quad x(t_0) = x_0, \quad t_0 = 0, \quad x_0 = 0, \quad t \in [t_0, T],$$

hat unter anderem die beiden Lösungen $x(t) = 0$ und $x(t) = t^2$.

Jedes explizite ESV wird eine Nullfolge erzeugen und stellt damit an den Integrationsstellen $t_i = t_0 + ih$, $h = \frac{T-t_0}{N}$, eine exakte Approximation der Nulllösung dar.

Mit einer AB $x(t_0) = x_0 > 0$ berechnen wir mit dem PZV eine nicht verschwindende Lösung.

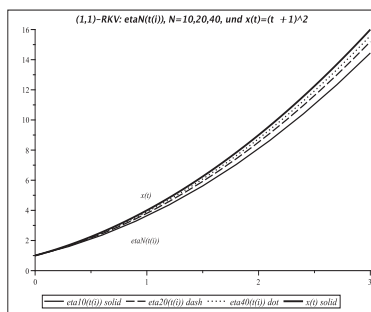


Abb. 8.3 Datei *iesv1_02.ps*,

$$x' = 2\sqrt{x}, \quad x(0) = 1, \quad t \in [0, 3],$$

$$\text{PZV, (1,1)-RKV } \eta_{i+1} = \eta_i + 2h\sqrt{\eta_i},$$

$$h = \frac{3}{N}, \quad N = 10, 20, 40,$$

Illustration der wachsenden Genauigkeit,

$$\text{Lösung } x(t) = (t+1)^2$$

Betrachten wir aber nun zwei einfache implizite ESV.

Für das IEV erhält man

$$\eta_{i+1} = \eta_i + h f(t_{i+1}, \eta_{i+1}) = \eta_i + 2h\sqrt{\eta_{i+1}},$$

$$\eta_{i+1} - 2h\sqrt{\eta_{i+1}} - \eta_i = 0, \quad z^2 = \eta_{i+1} \geq 0,$$

$$z^2 - 2hz - \eta_i = 0,$$

$$z_{1,2} = h \pm \sqrt{h^2 + \eta_i}.$$

Die Wahl der Teillösung $\eta_{i+1,2} = (h - \sqrt{h^2 + \eta_{i,2}})^2$, $\eta_{0,2} = 0$, liefert eine Nullfolge.

Wählt man aber $\eta_{i+1,1} = (h + \sqrt{h^2 + \eta_{i,1}})^2$ dann gilt $\eta_{i+1,1} > x(t_{i+1}) = ((i+1)h)^2$, $i = 0, 1, \dots$, mit der Konvergenz $\lim_{h \rightarrow 0} \eta_{i,1} = x(t_i)$ der Ordnung $\mathcal{O}(h)$, d.h. mit Halbierung der Schrittweite verkleinert sich der Fehler auf ungefähr die Hälfte.

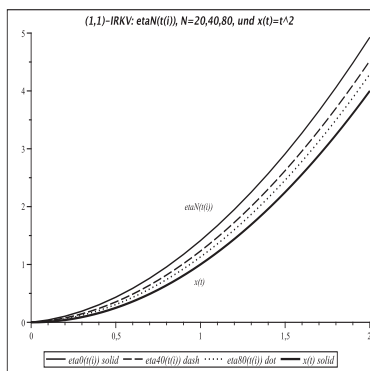


Abb. 8.4 Datei *iesv1.03.ps*,

$$x' = 2\sqrt{x}, \quad x(0) = 0, \quad t \in [0, 2],$$

IEV, (1,1)-IRKV $\eta_{i+1} = \eta_i + h f(t_{i+1}, \eta_{i+1})$

mit exakter Umstellung nach

$$\eta_{i+1} = \left(h + \sqrt{h^2 + \eta_i}\right)^2, \quad i = 0, 1, \dots, N-1,$$

$$\eta_0 = 0, \quad h = \frac{2}{N}, \quad N = 20, 40, 80,$$

Lösungskomponente $x(t) = t^2$

Für das IHV folgt

$$\eta_{i+1} = \eta_i + \frac{h}{2} (f(t_i, \eta_i) + f(t_{i+1}, \eta_{i+1}))$$

$$= \eta_i + h (\sqrt{\eta_i} + \sqrt{\eta_{i+1}}),$$

$$\eta_{i+1} - h \sqrt{\eta_{i+1}} - \eta_i - h \sqrt{\eta_i} = 0, \quad z^2 = \eta_{i+1} \geq 0,$$

$$z^2 - h z - \eta_i - h \sqrt{\eta_i} = 0,$$

$$z_{1,2} = \frac{h}{2} \pm \sqrt{\left(\frac{h}{2}\right)^2 + \eta_i + h \sqrt{\eta_i}}.$$

Die Wahl der Teillösung $\eta_{i+1,2} = \left(h/2 - \sqrt{h^2/4 + \eta_{i,2} + h \sqrt{\eta_{i,2}}}\right)^2$, $i = 0, 1, \dots, N-1$,

$\eta_{0,2} = 0$, ist wiederum eine Nullfolge, während man mit der anderen Beziehung

$\eta_{i+1,1} = \left(h/2 + \sqrt{h^2/4 + \eta_{i,1} + h \sqrt{\eta_{i,1}}}\right)^2$ genau die exakten Werte $x(t_{i+1}) = ((i+1)h)^2$

an den Integrationsstellen trifft. Es ist leicht zu zeigen, dass aus $\eta_{i,1} = x(t_i)$ die Beziehung $\eta_{i+1,1} = x(t_{i+1})$ folgt (vollständige Induktion).

Die beiden impliziten ESV mit der inneren Iteration $\eta_{i+1}^{(m+1)} = \eta_i + 2h \sqrt{\eta_{i+1}^{(m)}}$ bzw.

$\eta_{i+1}^{(m+1)} = \eta_i + h (\sqrt{\eta_i} + \sqrt{\eta_{i+1}^{(m)}})$ zum Startwert $\eta_{m+1,1}^{(0)} = \eta_i$ liefern die Nulllösung.

8.5 Rosenbrock-Wanner-Verfahren

Wir betrachten hier ausschließlich autonome SysgDGL. Sei

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad t \in [t_0, T]. \quad (8.44)$$

Nicht autonome (heteronome) SysgDGL $\mathbf{x}' = \mathbf{f}(t, \mathbf{x})$ mit n gDGL lassen sich einfach in autonome SysgDGL überführen. Man setzt $x_{n+1}(t) = t$, sodass sich daraus die zusätzliche gDGL $x'_{n+1}(t) = 1$ mit der AB $x_{n+1}(t_0) = t_0$ ergibt. Somit kann man immer von einem autonomen SysgDGL ausgehen.

Die Idee der Rosenbrock-Wanner-Verfahren (ROW-Verfahren) ist die Verwendung eines semi-impliziten RKV (vergl Def. 8.1), wo der Reihe nach für jeden Zuwachs

$$k_j = \mathbf{f}\left(\eta_i + h \sum_{l=1}^{j-1} \beta_{jl} k_l + h \beta_{jj} k_j\right), \quad j = 1, 2, \dots, r,$$

dieser durch eine Linearisierung der Funktion \mathbf{f} explizit notiert werden kann. Das bedeutet, dass in jeder Stufe genau ein Iterationsschritt mit einem vereinfachten Newton-Verfahren ausgeführt wird.

Ansonsten erhält man allgemein unter Verwendung des Newton-Verfahrens für die j -te Gleichung

$$\mathbf{g}(k_j) = k_j - \mathbf{f}\left(\eta_i + h \sum_{l=1}^{j-1} \beta_{jl} k_l + h \beta_{jj} k_j\right) = 0 \quad (8.45)$$

die Iterationsfolge

$$k_j^{(m+1)} = k_j^{(m)} - [\mathbf{g}'(k_j^{(m)})]^{-1} \mathbf{g}(k_j^{(m)}), \quad m = 0, 1, 2, \dots,$$

mit der Jacobi-Matrix $\mathcal{J}(\mathbf{g}(\mathbf{x})) = \mathbf{g}'(\mathbf{x})$. Für die Funktion \mathbf{f} heißt das

$$\begin{aligned} k_j^{(m+1)} &= k_j^{(m)} - [I - h\beta_{jj} \mathbf{f}'(\eta_j^{(m)})]^{-1} (k_j^{(m)} - \mathbf{f}(\eta_j^{(m)})), \quad m = 0, 1, \dots, \\ [I - h\beta_{jj} \mathbf{f}'(\eta_j^{(m)})] (k_j^{(m+1)} - k_j^{(m)}) &= -k_j^{(m)} + \mathbf{f}(\eta_j^{(m)}) \end{aligned} \quad (8.46)$$

mit

$$\eta_j^{(m)} = \eta_i + h \sum_{l=1}^{j-1} \beta_{jl} k_l + h \beta_{jj} k_j^{(m)}. \quad (8.47)$$

Man wählt die Startwerte $k_j^{(0)}$ jeweils als Linearkombination der bereits berechneten k -Werte

$$k_j^{(0)} = -\frac{1}{\beta_{jj}} \sum_{l=1}^{j-1} \gamma_{jl} k_l, \quad \gamma_{jl} \in \mathbb{R}.$$

Die Nachteile sind die ständige Neuberechnung der Jacobi-Matrix für jedes m und j sowie die Iteration als solche.

Mit einer Approximation \mathbf{J}_j der Jacobi-Matrix $\mathbf{f}'(\eta_j^{(0)})$ ergibt sich in der j -ten Stufe mit dem ersten Iterationsschritt

$$(I - h\beta_{jj} \mathbf{J}_j) (k_j^{(1)} - k_j^{(0)}) = -k_j^{(0)} + \mathbf{f}(\eta_j^{(0)}).$$

Mit dem Startwert $k_j^{(0)}$ und den Bezeichnungen $k_j = k_j^{(1)}$, $a_{jl} = \beta_{jl} - \gamma_{jl}$, $\gamma_j = \beta_{jj}$ folgt

$$\eta_j^{(0)} = \eta_i + h \sum_{l=1}^{j-1} a_{jl} k_l$$

sowie die Rosenbrock-Typ-Methode

$$\begin{aligned} (I - h\gamma_j \mathbf{J}_j) k_j &= \mathbf{f}\left(\eta_i + h \sum_{l=1}^{j-1} a_{jl} k_l\right) + h \mathbf{J}_j \sum_{l=1}^{j-1} \gamma_{jl} k_l, \quad j = 1, 2, \dots, r, \\ \eta_{i+1} &= \eta_i + h \sum_{l=1}^r b_l k_l. \end{aligned} \quad (8.48)$$

In der Formel (8.48) hat man zahlreiche Freiheiten bzgl. der Wahl der Parameter bzw. Größen. So kann man z.B. auf der rechten Seite für die Matrix \mathbf{J}_j die Einheitsmatrix nehmen. Möglich ist auch $\gamma_{jl} = 0$ zu setzen.

Auf der linken Seite ist \mathbf{J}_j eine Approximation von $\mathbf{f}'(\eta_j^{(0)})$, die aber auch gemäß $\mathbf{J}_j = \mathbf{f}'(\eta_i + h \sum_{l=1}^{j-1} d_{jl} k_l)$ mit weiteren Koeffizienten d_{jl} gewählt werden kann.

So findet man z.B. in [71] die linearisierte Form

$$\begin{aligned} \left[I - h d_{jj} \mathbf{f}'\left(\eta_i + h \sum_{l=1}^{j-1} d_{jl} k_l\right) \right] k_j &= \mathbf{f}\left(\eta_i + h \sum_{l=1}^{j-1} a_{jl} k_l\right), \quad j = 1, 2, \dots, r, \\ \eta_{i+1} &= \eta_i + h \sum_{l=1}^r b_l k_l. \end{aligned} \quad (8.49)$$

Seine Notation als zweistufiges Verfahren

$$\begin{aligned} (I - h d_{11} \mathbf{f}'(\eta_i)) k_1 &= \mathbf{f}(\eta_i), \\ (I - h d_{22} \mathbf{f}'(\eta_i + h d_{21} k_1)) k_2 &= \mathbf{f}(\eta_i + h a_{21} k_1), \\ \eta_{i+1} &= \eta_i + h (b_1 k_1 + b_2 k_2), \end{aligned}$$

führt im Sonderfall auf das (2, 2)-Calahan-Verfahren mit den Koeffizienten

$$d_{11} = d_{22} = \frac{3 + \sqrt{3}}{6}, \quad a_{21} = \frac{-2\sqrt{3}}{3}, \quad b_1 = \frac{3}{4}, \quad b_2 = \frac{1}{4}.$$

Die Lösung der Gleichung (8.45) als lineares Gleichungssystem für k_j durch explizite Umstellung der Formel nach k_j geht nur in besonderen Fällen.

Realisierbar ist jedoch die Lösung eines linearen Gleichungssystems für k_j , wobei die Koeffizientenmatrix $E_j = I - h\gamma_j \mathbf{J}_j$ für alle k_j sogar noch dieselbe sein kann, z.B. $E = I - h\gamma \mathbf{J} = I - h\gamma \mathbf{f}'(\eta_i)$. Damit braucht man pro Integrationsschritt nur eine einmalige Zerlegung von E und r Rücksubstitutionen.

Für autonome SysgDGL und bei der speziellen Wahl $\mathbf{J}_j = \mathbf{f}'(\eta_i)$ im Lösungsschritt ergeben sich die ROW-Methoden. Diese haben wegen der exakten Jacobi-Matrix (ihre Berechnung geschieht aber i. Allg. durch finite Differenzen) den Vorteil einer höheren Konsistenzordnung bei weniger Stufen gegenüber den folgenden W-Methoden.

Für eine beliebige Matrix $\tilde{\mathbf{J}}$ erhält man auf diese Weise (auch für nicht autonome SysgDGl) in Anlehnung an (8.48) die sogenannten W-Methoden

$$\begin{aligned}(I - h\gamma\tilde{\mathbf{J}})k_j &= \mathbf{f}\left(t_i + c_j h, \eta_i + h \sum_{l=1}^{j-1} a_{jl} k_l\right) + h\tilde{\mathbf{J}} \sum_{l=1}^{j-1} \gamma_{jl} k_l, \quad j = 1, 2, \dots, r, \\ \eta_{i+1} &= \eta_i + h \sum_{l=1}^r b_l k_l.\end{aligned}\tag{8.50}$$

Andere allgemeine Darstellungen, die insbesondere bei SysgDGl verwendet werden, sind

$$\begin{aligned}(I - h\gamma\mathbf{J}_i)k_j &= h\mathbf{f}\left(\eta_i + \sum_{l=1}^{j-1} \alpha_{jl} k_l\right) + h\mathbf{J}_i \sum_{l=1}^{j-1} \gamma_{jl} k_l, \quad j = 1, \dots, r, \quad \mathbf{J}_i = \mathbf{f}'(\eta_i), \\ &\text{bzw.} \\ (I - h\gamma\mathbf{J}_i)k_j &= h\mathbf{f}\left(\eta_i + \sum_{l=1}^{j-1} \alpha_{jl} k_l\right) + \sum_{l=1}^{j-1} c_{jl} k_l, \quad j = 1, \dots, r, \\ \eta_{i+1} &= \eta_i + \sum_{l=1}^r b_l k_l.\end{aligned}\tag{8.51}$$

In jedem Fall ist darauf zu achten, welche Formel beim ROW-Verfahren gewählt wurde.

Definition 8.2 *Ein semi-implizites RKV heißt ROW-Verfahren, falls zur Bestimmung aller Zuwächse k_1, k_2, \dots, k_r ein- und dieselbe Matrix*

$$E = I - h\gamma\mathbf{f}'(\eta_i)\tag{8.52}$$

mit der Konstanten γ verwendet wird.

Dies bedeutet eine Berechnung von $\mathbf{f}'(\eta_i)$ pro Schritt sowie die Bedingung (vergl. (8.48))

$$\gamma = \beta_{11} = \beta_{22} = \dots = \beta_{rr}.\tag{8.53}$$

Im Falle einer skalaren linearen gDGl heißt die Lösung des Gleichungssystems soviel wie die Umstellung nach k_j .

Die Verfahrensmodifikationen (8.52) und (8.50) gehen auf G. Wanner (1977) sowie T. Steihaug und A. Wolfbrandt (1979) zurück.

Definition 8.3 *Die Beschreibung eines Schritts des r -stufigen ROW-Verfahrens der Ordnung p ((r, p) -ROW) ist wie folgt.*

(a) *Gegeben:* t_i, η_i, h .

Gesucht: t_{i+1}, η_{i+1} .

(b) *Koeffizientenmatrix:* $E = I - h\gamma\mathbf{f}'(\eta_i) = I - h\gamma\mathbf{J}_i$.

(c) *Gleichungssysteme für die Zuwächse (modifizierte Ansatzformeln):*

$$\begin{aligned}E\tilde{k}_1 &= h\mathbf{f}(\eta_i), \\ E\tilde{k}_2 &= h\mathbf{f}(\eta_i + a_{21}\tilde{k}_1) + c_{21}\tilde{k}_1, \\ E\tilde{k}_3 &= h\mathbf{f}(\eta_i + a_{31}\tilde{k}_1 + a_{32}\tilde{k}_2) + c_{31}\tilde{k}_1 + c_{32}\tilde{k}_2, \\ &\dots \\ E\tilde{k}_r &= h\mathbf{f}(\eta_i + a_{r1}\tilde{k}_1 + a_{r2}\tilde{k}_2 + \dots + a_{r,r-1}\tilde{k}_{r-1}) + c_{r1}\tilde{k}_1 + c_{r2}\tilde{k}_2 + \dots + c_{r,r-1}\tilde{k}_{r-1}.\end{aligned}\tag{8.54}$$

Mittels des lokalen Diskretisierungsfehlers gelangt man zu den Parametern.

$$\begin{aligned} E \tilde{k}_1 &= h f(\eta_i), \\ E \tilde{k}_2 &= h f(\eta_i + a_{21} \tilde{k}_1) + c_{21} \tilde{k}_1, \\ E &= 1 - h \gamma f'(\eta_i), \\ \eta_{i+1} &= \eta_i + b_1 \tilde{k}_1 + b_2 \tilde{k}_2, \\ \tau_{i+1} &= \frac{x(t_{i+1}) - x(t_i)}{h} - \frac{1}{h}(b_1 \tilde{k}_1 + b_2 \tilde{k}_2) = \frac{x_{i+1} - x_i}{h} - \frac{1}{h}(b_1 \tilde{k}_1 + b_2 \tilde{k}_2), \end{aligned}$$

$$\begin{aligned} E^{-1} &= \frac{1}{1 - h \gamma f'(x_i)} = 1 + h \gamma f'(x_i) + \mathcal{O}(h^2), \\ E^{-2} &= \frac{1}{(1 - h \gamma f'(x_i))^2} = 1 + 2h \gamma f'(x_i) + \mathcal{O}(h^2), \\ \tau_{i+1} &= x'(t_i) + \frac{h}{2} x''(t_i) + \mathcal{O}(h^2) - \frac{1}{E} \left(\frac{b_1}{h} E \tilde{k}_1 + \frac{b_2}{h} E \tilde{k}_2 \right) \\ &= f(x_i) + \frac{h}{2} f'(x_i) f(x_i) + \mathcal{O}(h^2) - \frac{1}{E} \left(b_1 f(x_i) + b_2 (f(x_i + a_{21} \tilde{k}_1) + \frac{c_{21} \tilde{k}_1}{h}) \right) \\ &= f(x_i) + \frac{h}{2} f'(x_i) f(x_i) + \mathcal{O}(h^2) - (1 + h \gamma f'(x_i) + \mathcal{O}(h^2)) \cdot \\ &\quad \cdot (b_1 f(x_i) + b_2 (f(x_i) + a_{21} \tilde{k}_1 f'(x_i) + \mathcal{O}(h^2) + \frac{c_{21} \tilde{k}_1}{h})) \\ &= f(x_i) + \frac{h}{2} f'(x_i) f(x_i) + \mathcal{O}(h^2) - (1 + h \gamma f'(x_i)) \cdot \\ &\quad \cdot (b_1 f(x_i) + b_2 f(x_i) + b_2 a_{21} h E^{-1} f(x_i) f'(x_i) + \frac{b_2 c_{21}}{h} h E^{-1} f(x_i)) \\ &= f(x_i) + \frac{h}{2} f'(x_i) f(x_i) + \mathcal{O}(h^2) - (1 + h \gamma f'(x_i)) \cdot \\ &\quad \cdot (b_1 f(x_i) + b_2 f(x_i) + b_2 a_{21} h (1 + h \gamma f'(x_i)) f(x_i) f'(x_i) + \\ &\quad + b_2 c_{21} (1 + h \gamma f'(x_i)) f(x_i)) \\ &= f(x_i) + \frac{h}{2} f'(x_i) f(x_i) + \mathcal{O}(h^2) + \\ &\quad - b_1 f(x_i) - b_2 f(x_i) - b_2 a_{21} h f'(x_i) f(x_i) - b_2 c_{21} f(x_i) + \\ &\quad - 2b_2 c_{21} \gamma h f'(x_i) f(x_i) - (b_1 + b_2) \gamma h f'(x_i) f(x_i) \\ &= f(x_i) (1 - b_1 - b_2 - b_2 c_{21}) + \\ &\quad + h f'(x_i) f(x_i) \left(\frac{1}{2} - b_2 a_{21} - 2b_2 c_{21} \gamma - b_1 \gamma - b_2 \gamma \right) + \mathcal{O}(h^2). \end{aligned}$$

Aus den beiden Bedingungen folgen mit $b_1 + b_2 = 1$

$$\begin{aligned} 0 &= 1 - (b_1 + b_2) - b_2 c_{21} = -b_2 c_{21}, \\ 0 &= \frac{1}{2} - b_2 a_{21} - 2b_2 c_{21} \gamma - (b_1 + b_2) \gamma = \frac{1}{2} - b_2 (a_{21} - 2c_{21} \gamma) - \gamma \end{aligned}$$

sowie die beiden speziellen ROW-Verfahren mit $c_{21} = 0$, $b_1 = 0$, $b_2 = 1$ bzw. $b_1 = \frac{3}{4}$, $b_2 = \frac{1}{4}$, sowie $a_{21} = \frac{1-2\gamma}{2b_2}$ bei entsprechender Wahl von γ , das sind hier $\gamma = 1 - \frac{\sqrt{2}}{2}$ oder $\gamma = \frac{3+\sqrt{3}}{6}$.

Für die große Anzahl der Unbekannten lassen sich noch mehr Bedingungen formulieren, um die Genauigkeit in Richtung $\mathcal{O}(h^3)$ zu verbessern. Dazu braucht man dann nicht nur $x' = f(x)$, $x'' = ff'$, sondern noch $x''' = f^2 f'' + f(f')^2$ und Taylor-Reihenentwicklungen der Funktionen bis h^2 .

Spezielle 2-stufige ROW-Verfahren mit Parametertableau (erweitertes Butcher-Schema) mit den Formeln (8.54)

(2, 2)-Rosenbrock

$$\begin{array}{l} \gamma = \frac{2-\sqrt{2}}{2} \\ a_{21} = \frac{\sqrt{2}-1}{2} \\ \hline c_{21} = 0 \\ \hline b_1 = 0 \quad b_2 = 1 \end{array}$$

(2, 2)-Calahan

$$\begin{array}{l} \gamma = \frac{3+\sqrt{3}}{6} \\ a_{21} = \frac{-2\sqrt{3}}{3} \\ \hline c_{21} = 0 \\ \hline b_1 = \frac{3}{4} \quad b_2 = \frac{1}{4} \end{array}$$

Weitere A-stabile ROW-Verfahren der Ordnung 2, 3 in der Darstellung (8.48), $\gamma_j = \gamma$.

(2, 2)-Rosenbrock

$$\begin{array}{l} \gamma = \frac{2+\sqrt{2}}{2} \\ a_{21} = 1 \\ \hline \gamma_{21} = -2\gamma \\ \hline b_1 = \frac{1}{2} \quad b_2 = \frac{1}{2} \end{array}$$

(2, 2)-Rosenbrock

$$\begin{array}{l} \gamma = \frac{2+\sqrt{2}}{2} \\ a_{21} = 1 \\ \hline \gamma_{21} = -2\gamma \\ \hline b_1 = 1 \quad b_2 = 0 \end{array}$$

(2, 3)-ROW

$$\begin{aligned} (I - h\gamma \mathbf{J}_i) k_1 &= \mathbf{f}(\eta_i), \quad \mathbf{J}_i = \mathbf{f}'(\eta_i), \\ (I - h\gamma \mathbf{J}_i) k_2 &= \mathbf{f}(\eta_i + ha_{21}k_1) + h\mathbf{J}_i \gamma_{21}k_1, \\ \eta_{i+1} &= \eta_i + h(b_1k_1 + b_2k_2). \end{aligned}$$

$$\begin{array}{l} \gamma = \frac{3+\sqrt{3}}{6} \\ a_{21} = \frac{2}{3} \\ \hline \gamma_{21} = -\frac{4}{3}\gamma \\ \hline b_1 = \frac{1}{4} \quad b_2 = \frac{3}{4} \end{array}$$

(3, 3)-ROW

$$\begin{array}{l} \gamma = \frac{3+\sqrt{3}}{6} \\ a_{21} = 1 \\ a_{31} = 1 \quad a_{32} = 0 \\ \hline \gamma_{21} = -1 \\ \gamma_{31} = -\gamma \quad c_{32} = -1.077\ 350\ 269\ 189\ 625\ 7 \\ \hline b_1 = \frac{2}{3} \quad b_2 = 0 \quad b_3 = \frac{1}{3} \quad \text{bzw.} \quad b_1 = \frac{1}{3} \quad b_2 = \frac{1}{3} \quad b_3 = \frac{1}{3} \end{array}$$

(4, 4)-ROW von Kaps/Rentrop

$$\begin{aligned} (I - h\gamma \mathbf{J}_i) k_1 &= \mathbf{f}(\eta_i), \quad \mathbf{J}_i = \mathbf{f}'(\eta_i), \\ (I - h\gamma \mathbf{J}_i) k_2 &= \mathbf{f}(\eta_i + ha_{21}k_1) + h\mathbf{J}_i \gamma_{21}k_1, \\ (I - h\gamma \mathbf{J}_i) k_3 &= \mathbf{f}(\eta_i + h(a_{31}k_1 + a_{32}k_2)) + h\mathbf{J}_i (\gamma_{31}k_1 + \gamma_{32}k_2), \\ (I - h\gamma \mathbf{J}_i) k_4 &= \mathbf{f}(\eta_i + h(a_{41}k_1 + a_{42}k_2 + a_{43}k_3)) + h\mathbf{J}_i (\gamma_{41}k_1 + \gamma_{42}k_2 + \gamma_{43}k_3), \\ \eta_{i+1} &= \eta_i + h(b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4). \end{aligned}$$

$$\begin{aligned}
 \gamma &= 0.231 \\
 a_{21} &= 2\gamma \\
 a_{31} &= -0.081\,566\,816\,837\,71 & a_{32} &= 0.961\,775\,150\,166 \\
 a_{41} &= a_{31} & a_{42} &= a_{32} & a_{43} &= 0 \\
 \hline
 \gamma_{21} &= -0.270\,629\,667\,752 \\
 \gamma_{31} &= 0.311\,254\,483\,294 & \gamma_{32} &= 0.008\,524\,456\,284\,82 \\
 \gamma_{41} &= 0.282\,816\,832\,044 & \gamma_{42} &= -0.457\,959\,483\,281 & \gamma_{43} &= -0.111\,208\,333\,333 \\
 \hline
 b_1 &= 0.217\,487\,371\,653 & b_2 &= 0.486\,229\,037\,990 & b_3 &= 0 & b_4 &= 0.296\,283\,590\,357
 \end{aligned}$$

Durch die spezielle Wahl der a_{4i} werden nur drei Funktionsaufrufe benötigt.

Parameterschema zum (4,4)-ROW

$$\begin{array}{ccccccc}
 \gamma & & & & & & \\
 a_{21} & & & c_{21} & & & \\
 a_{31} & a_{32} & & c_{31} & c_{32} & & \\
 a_{41} & a_{42} & a_{43} & c_{41} & c_{42} & c_{43} & \\
 \hline
 & & & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4
 \end{array}$$

Erweitertes Butcher-Tableau mit Parametern/Feldern a,c,b,bd
zum (4,4)-ROW von Gottwald/Wanner

gamma = 0.395;

a = [0 0 0 0
0.438 0 0 0
0.938948678483 0.0730795420615 0 0
0.938948678483 0.0730795420615 0 0];

c = [0 0 0 0
-1.94347441895 0 0 0
0.416957530989 1.32396782073 0 0
1.51951325778 1.35370815030 -0.854151495258 0];

b = [0.748130767156 -0.201501814389 0.3679809905298 0]';
bd = [0.729044879960 0.0541069773272 0.281599362440 0.25]';
ev = bd-b;

Es werden die beiden Vektoren der Koeffizienten b_j und \hat{b}_j der Zuwächse notiert, um sie anschließend auch bei den eingebetteten Verfahren zu verwenden.

Für ein Verfahren der Ordnung p sind mindestens $r \geq p - 1$ Stufen erforderlich.

Bei geeigneter Wahl des Werts γ ergeben sich A-stabile Verfahren.

Wir notieren zur Implementation die MATLAB-Files.

Das sind die Funktion, die einen Schritt des ROW-Verfahrens realisiert, sowie ausgewählte Kommandos aus dem Rahmenprogramm (Skript-File).

```

% row4_fixstep.m
% Schritt des (4,4)-ROW fuer autonome SysgDgl

function [tn,etan] = row4_fixstep(f,J,ti,etai,h)
% f rechte Seite des SysgDGL
% J Jacobi-Matrix von f

% Parametertableau
global gamma a c b bd

etai = etai(:); % Spaltenvektor erzeugen
r = size(bd,1); % Stufenanzahl
n = size(etai,1); % Dimension des SysgDGL
p = r-1; % Ordnung
I = eye(n);
y = f(etai);
rs = h*y;

E = I-gamma*h*J(etai); % E*k(:,1)=rs
[L,U,P] = lu(E,'vector'); % LU*k(:,1)=r(P)
% k(:,1) = U(L\rs(P)); % Kurzform
rr = rs(P); % Vorwaerts- und Rueckwaertsrechnung
v(1) = rr(1);
for l=2:n, v(l)=rr(l)-L(l,1:l-1)*(v(1:l-1))'; end
w(n) = v(n)/U(n,n);
for l=n-1:-1:1, w(l)=(v(l)-U(l,l+1:n)*(w(l+1:n)))/U(l,l); end
k(:,1) = w';

for j=2:r
z = etai; for m=1:j-1, z=z+a(j,m)*k(:,m); end
y = f(z);
rs = h*y; for m=1:j-1, rs=rs+c(j,m)*k(:,m); end
% k(:,j) = U(L\rs(P)); % Kurzform
rr = rs(P);
v(1) = rr(1);
for l=2:n, v(l)=rr(l)-L(l,1:l-1)*(v(1:l-1))'; end
w(n) = v(n)/U(n,n);
for l=n-1:-1:1, w(l)=(v(l)-U(l,l+1:n)*(w(l+1:n)))/U(l,l); end
k(:,j) = w';
end

tn = ti+h;
etan = etai;
for j=1:r, etan=etan+bd(j)*k(:,j); end
end

```

Die rechte Seite des SysgDGL sowie die Jacobi-Matrix sind Funktionen.

Die LU -Zerlegung des Matrix E wird dann für die wiederholte Lösung des gestaffelten linearen Gleichungssystems (Vorwärts- und Rückwärtsrechnung) verwendet.

In den folgenden Kommandos des Skript-Files sind Ausgaben von Ergebnissen und Zwischengrößen u.ä. nicht berücksichtigt.

```

r_row4_fixstep.m
...
% Eingabe
while 1
    t0 = input('t0 = ');
    t1 = input('t1 = ');
    if t1>t0, break, end
end
x0 = input('x0 (Vektor von AW) = ');
x0 = x0(:); % Spaltenvektor

%while 1
% N = input('Anzahl der Teilintervalle N = ');
% if 0<N, break, end
%end
for N = [5,10,20,40,80,160,320]
    h = (t1-t0)/N;
    i = 1;
    t(i) = t0;
    eta(:,i) = x0;
    ti = t0;
    etai = x0;

    for i = 2:N+1
        [tn,etan] = row4_fixstep(@fas03,@Jfas03,ti,etai,h);
        ti = tn; t(i) = tn;
        etai = etan; eta(:,i) = etan;
    end
end
end

```

8.5.2 Eingebettete ROW-Verfahren

Ausgangspunkt sind zwei ROW-Verfahren, miteinander durchaus “verwandt“, aber von unterschiedlicher Genauigkeit. Durch ihren Vergleich ist eine Fehlerschätzung möglich und damit eine SWS des Verfahrens wie bei eingebetteten RKV (vergl. Abschnitt 5.4.2). Man benötigt also ein $(r - 1, p)$ -ROW mit

$$\eta_{i+1} = \eta_i + b_1 \tilde{k}_1 + b_2 \tilde{k}_2 + \dots + b_{r-1} \tilde{k}_{r-1}$$

und ein $(r, p + 1)$ -ROW mit

$$\hat{\eta}_{i+1} = \hat{\eta}_i + \hat{b}_1 \tilde{k}_1 + \hat{b}_2 \tilde{k}_2 + \dots + \hat{b}_{r-1} \tilde{k}_{r-1} + \hat{b}_r \tilde{k}_r,$$

wobei sich beide nur durch eine zusätzliche Zeile der Koeffizienten \hat{b}_j unterscheiden. Die Fehlerschätzung nimmt man gemäß

$$\begin{aligned} \hat{\eta}_{i+1} - \eta_{i+1} &= (\hat{b}_1 - b_1) \tilde{k}_1 + (\hat{b}_2 - b_2) \tilde{k}_2 + \dots + (\hat{b}_{r-1} - b_{r-1}) \tilde{k}_{r-1} + \hat{b}_r \tilde{k}_r, \\ &= e_1 \tilde{k}_1 + e_2 \tilde{k}_2 + \dots + e_{r-1} \tilde{k}_{r-1} + e_r \tilde{k}_r \end{aligned}$$

vor.

Der lokale Diskretisierungsfehler für das erste Verfahren beträgt

$$\tau_{i+1} = \underbrace{\frac{1}{h}(\hat{\eta}_{i+1} - \eta_{i+1})}_{\mathcal{O}(h^p)} + \mathcal{O}(h^{p+1}).$$

Den lokalen Fehler $\bar{\tau}_{i+1} \approx \hat{\eta}_{i+1} - \eta_{i+1} = \mathcal{O}(h^{p+1})$ verwendet man als Fehlerschätzer für das $(p, p+1)_r$ -ROW im Sinne eines absoluten Fehlers oder absolut-relativen Fehlers

$$err = \frac{|\hat{\eta}_{i+1} - \eta_{i+1}|}{1 + |\hat{\eta}_{i+1}|} = \frac{|e_1 \tilde{k}_1 + e_2 \tilde{k}_2 + \dots + e_{r-1} \tilde{k}_{r-1} + e_r \tilde{k}_r|}{1 + |\hat{\eta}_{i+1}|}. \quad (8.55)$$

Im Vektorfall kann man mit der Maximumnorm arbeiten.

Die SWS bei der Einbettung verwendet die Größe err .

Wenn bei einer Rechnung bei gegebener Toleranz $\varepsilon > 0$ und mit der Schrittweite h der Fehler err die Bedingung $\frac{\varepsilon}{err} \leq 1$ erfüllt, kann man grob gesagt, mit der Schrittweite weiterrechnen.

Das Fehlerkriterium lässt sich aber etwas verfeinern.

Als Schrittweitevorschlag für den nächsten Schritt nimmt man im Erfolgsfall

$$h_{neu} = \delta \rho h, \quad \rho = \sqrt[p]{\frac{\varepsilon}{err}}. \quad (8.56)$$

Dabei bezeichnet $\delta \in (0, 1]$ einen geeigneten Sicherheitsfaktor.

Bei Misserfolg war ja err zu groß, der Schritt muss dann wiederholt werden und zwar mit der kleineren Schrittweite

$$h_{neu} = \max\left(\frac{h}{2}, \delta \rho h\right). \quad (8.57)$$

Dabei wird das bisherige h nicht mehr als halbiert.

Zusätzliche Sicherheitsschranken sind gegebene minimale und maximale Schrittweiten sowie der Abbruch bei zu kleiner Schrittweite.

Günstig ist, wenn für die Toleranz ε die Anfangsschrittweite akzeptiert wird und man nicht gleich die Schrittweite verkleinern muss. Auch eine zu kleine Anfangsschrittweite mit nachfolgenden mehreren Vergrößerungen ist nicht effizient. Dazu spielt die geeignete Wahl des Sicherheitsfaktors δ eine Rolle (in der Regel problemabhängig).

Effiziente eingebettete A-stabile ROW-Verfahren sind die von Gottwald/Wanner oder Kaps/Rentrop mit $r = 4$ Stufen und Ordnung $p = 4$ mit den Formeln (8.54).

Parameterschema zum $(3, 4)_4$ -ROW

$$\begin{array}{c} \gamma \\ \begin{array}{ccc|cc} a_{21} & & & c_{21} & \\ a_{31} & a_{32} & & c_{31} & c_{32} \\ a_{41} & a_{42} & a_{43} & c_{41} & c_{42} & c_{43} \\ \hline & & & b_1 & b_2 & b_3 \\ & & & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 \\ & & & e_1 & e_2 & e_3 & e_4 \end{array} \end{array}$$

Eingebettetes (3,4)-ROW-Verfahren von Gottwald/Wanner

$\gamma = 0.395$		
$a_{21} = 0.438$	$c_{21} = -1.943\,474\,418\,95$	
$a_{31} = 0.938\,948\,678\,483$	$c_{31} = 0.416\,957\,530\,989$	
$a_{32} = 0.073\,079\,542\,061\,5$	$c_{32} = 1.323\,967\,820\,73$	
$a_{41} = a_{31}$	$c_{41} = 1.519\,513\,257\,78$	
$a_{42} = a_{32}$	$c_{42} = 1.353\,708\,150\,30$	
$a_{43} = 0$	$c_{43} = -0.854\,151\,495\,258$	
	$b_1 = 0.748\,130\,767\,156$	$e_1 = -0.019\,085\,887\,199\,99$
	$\hat{b}_1 = 0.729\,044\,879\,960$	
	$b_2 = -0.201\,501\,814\,389$	$e_2 = 0.255\,608\,791\,716$
	$\hat{b}_2 = 0.054\,106\,977\,327\,2$	
	$b_3 = 0.367\,980\,990\,529\,8$	$e_3 = -0.086\,381\,628\,089\,8$
	$\hat{b}_3 = 0.281\,599\,362\,440$	
	$b_4 = 0$	
	$\hat{b}_4 = 0.25$	$e_4 = 0.25$

Eingebettetes (3,4)-ROW-Verfahren von Kaps/Rentrop

$\gamma = 0.395$		
$a_{21} = 0.438$	$c_{21} = -0.767\,672\,395\,484$	
$a_{31} = 0.796\,692\,457\,938$	$c_{31} = -0.851\,675\,323\,742$	
$a_{32} = 0.073\,079\,542\,061\,5$	$c_{32} = 0.522\,967\,289\,188$	
$a_{41} = 0$	$c_{41} = 0.288\,463\,109\,545$	
$a_{42} = 0$	$c_{42} = 0.088\,021\,427\,34$	
$a_{43} = 0$	$c_{43} = -0.337\,389\,840\,627$	
	$b_1 = 0.346\,325\,833\,758$	$e_1 = -0.147\,032\,558\,057$
	$\hat{b}_1 = 0.199\,293\,275\,701$	
	$b_2 = 0.285\,693\,175\,712$	$e_2 = 0.196\,952\,059\,962$
	$\hat{b}_2 = 0.482\,645\,235\,674$	
	$b_3 = 0.367\,980\,990\,529\,8$	$e_3 = -0.299\,919\,501\,904$
	$\hat{b}_3 = 0.068\,061\,488\,625\,6$	
	$b_4 = 0$	
	$\hat{b}_4 = 0.25$	$e_4 = 0.25$

MATLAB-Implementation

```
% row4_varstep.m
% Schritt des eingebetteten (3,4)-ROW mit Schrittweitensteuerung
% fuer autonome SysgDgl

function [tn,etan,err,rho] = row4_varstep(f,J,ti,etai,h,epsi)
% f rechte Seite des SysgDgl
% J Jacobi-Matrix von f
```

```

% Parametertableau
global gamma a c b bd
ev = bd-b;
etai = etai(:); % Spaltenvektor erzeugen
r = size(bd,1); % Stufenanzahl
n = size(etai,1); % Dimension des SysgDGL
p = r-1; % Ordnung
I = eye(n);
y = f(etai);
rs = h*y;

E = I-gamma*h*J(etai); % E*k(:,1)=rs
[L,U,P] = lu(E,'vector'); % LU*k(:,1)=rs(P)
% k(:,1) = U(L\rs(P)); % Kurzform
rr = rs(P); % Vorwaerts- und Rueckwaertsrechnung
v(1) = rr(1);
for l=2:n, v(l)=rr(l)-L(l,1:l-1)*(v(1:l-1))'; end
w(n) = v(n)/U(n,n);
for l=n-1:-1:1, w(l)=(v(l)-U(l,1+1:n)*(w(1+1:n)))/U(l,l); end
k(:,1) = w';

for j=2:r
z = etai; for m=1:j-1, z=z+a(j,m)*k(:,m); end
y = f(z);
rs = h*y; for m=1:j-1, rs=rs+c(j,m)*k(:,m); end
% k(:,j) = U(L\rs(P)); % Kurzform
rr = rs(P);
v(1) = rr(1);
for l=2:n, v(l)=rr(l)-L(l,1:l-1)*(v(1:l-1))'; end
w(n) = v(n)/U(n,n);
for l=n-1:-1:1, w(l)=(v(l)-U(l,1+1:n)*(w(1+1:n)))/U(l,l); end
k(:,j) = w';
end
tn = ti+h;
etan = etai;
for j=1:r, etan=etan+bd(j)*k(:,j); end
er = zeros(n,1); for j=1:r, er=er+ev(j)*k(:,j); end
er = abs(er);
err = max(er./(1+abs(etan)));
rho = (epsi/err)^(1/(p+1));
end

%-----
% r_row4_varstep.m
...
disp('Eingebettetes (3,4)-ROW von Gottwald/Wanner mit SWS');
% Eingabe
while 1
t0 = input('t0 = ');
t1 = input('t1 = ');
if t1>t0, break, end
end
x0 = input('x0 (Vektor von AW) = ');
x0 = x0(:); % Spaltenvektor
epsi = input('eps (Toleranz) = ');
while 1
h0 = input('Startschrittweite h0 = ');
hmin = input('Minimale Schrittweite hmin = ');
hmax = input('Maximale Schrittweite hmax = ');
if 0<hmin && hmin<h0 && h0<hmax, break, end

```



```

end
delta = 0.9; % 0<delta<=1, Sicherheitsfaktor
h = h0;
zaehl = 0; % Zaehler der Wiederholungen (Misserfolg)
i = 1; % Zaehler der Erfolgsschritte
t(i) = t0; eta(:,i) = x0;
err = 0;
ti = t0; etai = x0;

while ti<t1
if h<hmin, disp('Schrittweite zu klein'); exit, end
if h>hmax, h = hmax; end
if ti+h>t1, h=t1-ti; end
while 1
[tn,etan,err,rho] = row4_varstep(@fas03,@Jfas03,ti,etai,h,epsi);
if err<epsi % Erfolg
i = i+1;
hv(i-1) = h;
ti = tn; t(i) = tn;
etai = etan; eta(:,i) = etan;
h = fak*rho*h;
break
else % Misserfolg
zaehl = zaehl+1;
h = max(h/2,fak*rho*h);
if h<hmin, disp('Schrittweite zu klein'); exit, end
end
end
end
end

```

Bemerkung 8.1

In der ODE-Suite von MATLAB ist als ODE23s das folgende eingebettete Paar von linear impliziten Verfahren der Ordnungen 2 und 3 bei Verwendung der Zuwächse implementiert. Das SysgDGI kann auch nicht autonom sein.

$$\begin{aligned}
 \gamma &= \frac{1}{2+\sqrt{2}}, \quad \mathbf{J}_i \approx \mathbf{f}_x(t_i, \eta_i), \quad \mathbf{t}_i \approx \mathbf{f}_t(t_i, \eta_i), \\
 E &= I - h\gamma \mathbf{J}_i, \\
 \mathbf{f}_i &= \mathbf{f}(t_i, \eta_i), \\
 E k_1 &= \mathbf{f}_i + h\gamma \mathbf{t}_i, \\
 \mathbf{f}_{i+1/2} &= \mathbf{f}\left(t_i + \frac{h}{2}, \eta_i + \frac{h}{2} k_1\right), \\
 E k_2 &= \mathbf{f}_i - k_1 + E k_1, \\
 \eta_{i+1} &= \eta_i + h k_2, \\
 \mathbf{f}_{i+1} &= \mathbf{f}(t_{i+1}, \eta_{i+1}), \\
 E k_3 &= \mathbf{f}_{i+1} - (6 + \sqrt{6})(k_2 - \mathbf{f}_{i+1/2}) - 2(k_1 - \mathbf{f}_i) + h\gamma \mathbf{t}_i, \\
 err &\approx \frac{h}{6}(k_1 - 2k_2 + k_3).
 \end{aligned}$$

Ist ein Schritt erfolgreich ($|err| < \varepsilon$), so kann \mathbf{f}_{i+1} dieses Schrittes als \mathbf{f}_i des folgenden Schrittes verwendet werden. Wählt man \mathbf{J}_i als exakte Jacobi-Matrix und nicht nur als Approximation, so erhält man die sogenannte L-Stabilität des Verfahrens.

Zu den Verfahren werden einige Beispiele betrachtet.

Die Funktion `row4_fixstep` wird im Lösungsintervall mit einer Folge von Anzahlen N von Teilintervallen gerechnet. Damit kann man i. Allg. für ROW-Verfahren der Ordnung p bestätigen, dass die Verdopplung von N den Fehler um den Faktor 2^p verkleinert, wenn N moderat (nicht zu klein oder zu groß) ist.

Zur Funktion `row4_varstep` testen wir zahlreiche Parametersituationen, weitere werden dem Leser überlassen.

Da ist zunächst die Wahl des Sicherheitsfaktors $\delta \in (0, 1]$, der eher etwas größer als zu klein gewählt werden sollte, um den Gesamtaufwand gering zu halten.

Dann haben wir die Anfangsschrittweite h_0 . Je kleiner die Toleranz ε ist, desto kleiner sollte auch h_0 sein. Ist h_0 für die Toleranz ε zu groß, dann werden anfängliche Verkleinerungen eintreten (Misserfolgssituation und Schrittreduktion) bis zur ersten akzeptierten Schrittweite. Im einem Schritt genügt zumeist auch eine Reduktion.

Ist sie zu klein, dann treten zwei Fälle auf. Entweder werden dann eine Reihe von Vorschläge für Vergrößerungen gemacht, oder der Vergrößerungsvorschlag wird zu groß, so dass wieder eine Schrittreduktion erfolgt. Damit braucht die Anfangsrechnung insgesamt zu viele Schritte.

Die Misserfolgsrate sollte jedoch niedrig bleiben. In den Rechnungen wird versucht, mit einer möglichst günstigen Anfangsschrittweite h_0 zu beginnen. Weiterhin wird auf die Schrittweitevorschläge (8.56) und (8.57) verwiesen.

Beispiel 8.3

(1) Riccati-DGL $x' = t - x^2$ (vergl. Abschnitt 2.2.3)

Die allgemeine parameterabhängige Lösung ist

$$x(t, c) = \frac{c \operatorname{AiryAi}(1, t) + \operatorname{AiryBi}(1, t)}{c \operatorname{AiryAi}(t) + \operatorname{AiryBi}(t)}.$$

Lösung zur AB $x(0) = 1$

$$x(t) = \frac{c \operatorname{AiryAi}(1, t) + \operatorname{AiryBi}(1, t)}{c \operatorname{AiryAi}(t) + \operatorname{AiryBi}(t)}, \quad c = \frac{-2\pi 3^{5/6} + 3^{5/3} \Gamma^2(2/3)}{2\pi 3^{1/3} + 3^{7/6} \Gamma^2(2/3)}, \quad -1 < t < \infty.$$

Lösung zur AB $x(0) = 0$, $t \in [t_0, t_1] = [0, t_1]$

`% Maple`

```
x(t) = (sqrt(3)*AiryAi(1,t)+AiryBi(1,t))/(sqrt(3)*AiryAi(t)+AiryBi(t));
```

`% MATLAB`

```
x(t) = (sqrt(3)*airy(1,t)+airy(3,t))/(sqrt(3)*airy(0,t)+airy(2,t));
```

```
x(t) = (3^(1/2)*AiryAi(1,t)+AiryBi(1,t))/(3^(1/2)*AiryAi(t)+AiryBi(t));
```

Das autonome SysgDGL ist

$$\begin{aligned} x'_1(t) &= x_2(t) - x_1(t)^2, & x_1(0) &= 0, & \eta_{1,i} &\approx x_1(t_i), \\ x'_2(t) &= 1, & x_2(0) &= 0, & \eta_{2,i} &\approx x_2(t_i). \end{aligned}$$

Von Interesse ist natürlich die Lösung $x(t) = x_1(t)$ und ihre numerische Berechnung.

r_row4_varstep.m

```

eps      - Toleranz
h0       - Anfangsschrittweite
i        - Anzahl der akzeptierten Schritte in [0,t1]=[0,1]
zaehl    - Anzahl der Misserfolge
[Schritt] - Stellen, wo Schritt wiederholt wurde
h-min    - minimale Schrittweite der Rechnung,
          meist angenommen nahe dem Intervallende
max      - =max(|eta(1,t(i))-x(t(i))|)

```

delta=0.95

eps	h0	i	zaehl [Schritt]	h-min	max
1e-04	0.500	3	1 [2]	0.341690	7.376e-005
1e-05	0.400	5	3 [2,3,4]	0.147096	2.770e-005
1e-06	0.250	9	7 [2,3,4,5,6,7,8]	0.075613	2.913e-006
1e-07	0.150	15	8 [2,3,4,5,6,7,8,9]	0.043021	2.765e-007
1e-08	0.100	27	9 [2,3,4,5,6,7,8,9,10]	0.023123	3.940e-008
1e-09	0.050	48	9 [2,3,4,5,6,7,8,9,10]	0.012800	2.626e-009
1e-10	0.035	86	6 [2,3,4,5,6]	0.007107	3.763e-010

delta=0.9

eps	h0	i	zaehl [Schritt]	h-min	max
1e-04	0.500	3	1 [2]	0.323706	7.376e-005
1e-05	0.400	5	3 [2,3,4]	0.141160	2.770e-005
1e-06	0.250	9	4 [2,3,4,5]	0.076516	2.913e-006
1e-07	0.150	16	5 [2,3,4,5,6]	0.040026	2.677e-007
1e-08	0.100	28	3 [2,3,4]	0.021839	3.766e-008
1e-09	0.050	50	2 [2,3]	0.012125	2.488e-009
1e-10	0.035	90	2 [2,3]	0.006744	3.509e-010

delta=0.8

eps	h0	i	zaehl [Schritt]	h-min	max
1e-04	0.500	3	1 [2]	0.287739	7.376e-005
1e-05	0.400	5	2 [2,3]	0.149535	2.770e-005
1e-06	0.250	10	2 [2,3]	0.065306	2.913e-006
1e-07	0.150	17	1 [2]	0.035466	2.578e-007
1e-08	0.100	31	1 [2]	0.019376	3.626e-008
1e-09	0.050	57	1 [2]	0.010694	2.059e-009
1e-10	0.035	101	1 [2]	0.005990	2.914e-010

delta=0.7

eps	h0	i	zaehl [Schritt]	h-min	max
1e-04	0.500	3	1 [2]	0.251772	7.376e-005
1e-05	0.400	6	1 [2]	0.113081	2.770e-005
1e-06	0.250	11	1 [2]	0.056273	2.913e-006
1e-07	0.150	19	0	0.030776	2.874e-007
1e-08	0.100	35	0	0.016895	3.839e-008
1e-09	0.050	64	0	0.009377	2.242e-009
1e-10	0.035	115	0	0.005237	2.926e-010

Einige Vergleichsrechnungen bezüglich des Sicherheitsfaktors δ ergeben die Empfehlung $\delta \in [0.8, 0.9]$. Bei $\delta = 1$ kommt es bei einigen Integrationsschritten zu mehreren Reduktionen. Die Anfangsschrittweite h_0 ist günstig und wird in jedem Fall akzeptiert. Der Fehler bei $t_1 = 1$ ist zugleich auch der betragsgrößte im Lösungsintervall $[0, 1]$.

```
r_row4_fixstep.m
t1=1
```

N	h	eta(1,N) (t(N)=1)	e(N,h)= eta(1,N)-x(t(N))	e(N,h)/ e(2N,h/2)
5	0.2000000	0.455526551738	-1.797e-005	15.6
10	0.1000000	0.455543374355	-1.152e-006	15.8
20	0.0500000	0.455544453339	-7.274e-008	15.9
40	0.0250000	0.455544521511	-4.570e-009	15.9
80	0.0125000	0.455544525794	-2.883e-010	14.4
160	0.0062500	0.455544526062	-2.002e-011	6.2
320	0.0031250	0.455544526079	-3.240e-012	1.5
640	0.0015625	0.455544526080	-2.196e-012	
x(t(N))= 0.455544526082				16

```
=====
t1=5
```

N	h	eta(1,N) (t(N)=5)	e(N,h)= eta(1,N)-x(t(N))	e(N,h)/ e(2N,h/2)
5	1.0000000	2.182068136527	-7.163e-004	1219.9
10	0.5000000	2.182783857830	-5.872e-007	0.2
20	0.2500000	2.182786827644	2.383e-006	7.7
40	0.1250000	2.182784755204	3.102e-007	11.2
80	0.0625000	2.182784472623	2.761e-008	13.4
160	0.0312500	2.182784447073	2.064e-009	14.9
320	0.0156250	2.182784445147	1.385e-010	22.6
640	0.0078125	2.182784445015	6.131e-012	
x(t(N))= 2.182784445009				16

Die ROW-Verfahren im Intervall $[0, 1]$ mit fester Schrittweite sowie mit Einbettung/SWS bei $\delta = 0.8$ zeigen bzgl. der Toleranz, Schrittweite, Genauigkeit und Aufwand ähnliche Ergebnisse. Der Faktor 16 verweist auf die Genauigkeitsordnung 4 des Verfahrens.

(2) Aus den Abschnitten 7.3.1 und 8.1 entnehmen wir das steife autonome SysgDGL und AWP

$$\mathbf{x}'(t) = A \mathbf{x}(t), \quad \mathbf{x}(0) = (1, 1)^T, \quad t \in [0, 1], \quad \text{wobei } A = \begin{pmatrix} -29\,998 & -39\,996 \\ 29\,997/2 & 19\,997 \end{pmatrix}$$

mit der Lösung $\mathbf{x} = (x, y)^T$

$$x(t) = -6e^{-t} + 7e^{-10000t}, \quad y(t) = \frac{9}{2}e^{-t} - \frac{7}{2}e^{-10000t}, \quad \eta_i = (\eta_{1,i}, \eta_{2,i}) \approx (x(t_i), y(t_i)).$$

Die Eigenwerte der Systemmatrix A sind $\lambda_{1,2} = -1, -10\,000$. Damit müssen explizite Verfahren mit einer Schrittweitenbeschränkung der Größenordnung $h = 10^{-4} \dots 10^{-3}$ rechnen. Das SysgDGL besitzt eine transiente und asymptotische Phase der Lösung mit der Phasengrenze bei ungefähr $t = 0.0005$.

Welche Ergebnisse liefern die ROW-Verfahren?

Das ROW-Verfahren mit konstanter Schrittweite in $[0, 1]$ kann nur vernünftigt rechnen, wenn die Schrittweite hinreichend klein ist, z.B. $h \approx \mathcal{O}(10^{-4})$, um in der transienten Phase die sehr steilen Lösungsverläufe der Lösungskomponenten gut zu approximieren.

In dieser Phase entstehende Fehler wirken sich natürlich auf die weitere Rechnung schon aus. Man bemerkt es z.B. bei der Kontrolle der Verbesserung der Genauigkeit bei Schrittweithalbung, wo der Faktor 16 nur in einem Fall näherungsweise erhalten wird. Der Fehler bei $t_1 = 1$ in der asymptotischen Phase wird hingegen sehr klein werden.

r_row4_fixstep1.m

```
max=max|e(i,h)|=max|eta(i)-xy(t(i))|
t1=1
```

N	h	eta(N) (t(N)=1)	e(N,h)= eta(N)-xy(t(N))	max	e(h)/ e(h/2)
125	0.0080000	-2.207276639217 1.655457481366	7.812e-009 -3.905e-009	5.936e+000 2.968e+000	1.2
250	0.0040000	-2.207276647034 1.655457485276	-5.454e-012 4.092e-012	5.060e+000 2.530e+000	1.4
500	0.0020000	-2.207276647034 1.655457485276	-5.647e-012 4.236e-012	3.684e+000 1.842e+000	1.9
1000	0.0010000	-2.207276647034 1.655457485276	-5.658e-012 4.244e-012	1.964e+000 9.818e-001	3.3
2000	0.0005000	-2.207276647035 1.655457485276	-5.892e-012 4.420e-012	5.928e-001 2.964e-001	7.3
4000	0.0002500	-2.207276647034 1.655457485276	-5.418e-012 4.064e-012	8.111e-002 4.055e-002	16.5
8000	0.0001250	-2.207276647035 1.655457485276	-6.263e-012 4.698e-012	4.924e-003 2.462e-003	32.6
16000	0.0000625	-2.207276647035 1.655457485276	-5.545e-012 4.159e-012	1.512e-004 7.558e-005	

```
x(t(N))=-2.207276647029
y(t(N))= 1.655457485272
```

16

Vergleichen wir das eingebettete (3,4)-ROW mit automatischer SWS in $[0, 1]$ bei der Toleranz $\varepsilon = 10^{-8}$ und Startschrittweite $h_0 = 2 \cdot 10^{-6}$ mit dem eingebetteten (5,6)-RKV mit Tab. 8.1. Durch die A-Stabilität hat man hier einen deutlich geringeren Aufwand an Funktionswertberechnungen. Auch die Rechenzeit ist kürzer als beim (5,6)-RKV.

	Anzahl der Schritte	aktueller Wert t	aktuelle Schrittweite h
Phase 1	32	6.24e-5	2.40e-6
	64	1.56e-4	3.47e-6
	128	5.06e-4	8.77e-6
	150	7.73e-4	1.70e-5
Phase 2	160	1.00e-3	2.94e-5
	170	1.56e-3	1.03e-4
	175	4.58e-3	1.80e-3
	177	4.09e-2	2.91e-2
	180	1.20e-1	2.63e-2
	190	3.84e-1	2.66e-2
	200	6.51e-1	2.70e-2
	212	9.79e-1	2.76e-2
	213	1	2.11e-2

Tab. 8.3

Eingebettetes (3,4)-ROW von Gottwald/Wanner mit SWS, zu ausgewählten Integrationsstellen, jeweilige Schrittzahl bis dahin sowie Schrittweite

Die Fehlerauswertung an den Integrationsstellen ergibt

$$\max_{i=1,\dots,N} |\eta_{1,i} - x(t_i)| = 1.51e - 9, \quad \max_{i=1,\dots,N} |\eta_{2,i} - y(t_i)| = 7.55e - 10, \quad N = 213.$$

In einer Auswertung von Testrechnungen des Verfahrens mit SWS kann man die minimale und maximale Schrittweite ermitteln, die an der Phasengrenze und wo die Maximalabweichungen auftreten, dazu die jeweiligen Integrationsstellen als t -Werte und ihre Nummerierung. Die allerletzte Schrittweite wird dabei nicht berücksichtigt.

r_row4_varstep1.m

```

eps      - Toleranz
h0       - Anfangsschrittweite
i        - Anzahl der akzeptierten Schritte in [0,t1]=[0,1]
zaehl    - Anzahl der Misserfolge
[Schritte] - Stellen, wo Schritt wiederholt wurde
max      - =max(|eta(1/2,t(i))-x/y(t(i))|)
[index-maxabs bei t=...] Index der Stelle der max. Abweichungen und Stelle
          (max meist angenommen kurz nach der Phasengrenze)

h-min    - minimale Schrittweite der Rechnung mit Stelle und Nummer
          (meist angenommen nahe dem Intervallanfang, h-min~h0)
h-pha    - Schrittweite an der Phasengrenze t=0.0005 mit Nummer
          (Nummer in [i/2,2i/3])
h-max    - maximal Schrittweite der Rechnung mit Stelle und Nummer
          (meist angenommen nahe dem Intervallende)

```

delta=0.9

eps	h0	i	zaehl	max(eta(1/2,t(i))-x/y(t(i)))	[index-maxabs bei t=...]
1e-10	0.05e-5	668	0	1.541e-11	7.701e-12 [546 bei t=0.0022567]
1e-09	0.10e-5	377	0	1.528e-10	7.641e-11 [306 bei t=0.0019557]
1e-08	0.20e-5	213	0	1.510e-09	7.548e-10 [171 bei t=0.0016929]
1e-07	0.30e-5	122	0	1.543e-08	7.717e-09 [96 bei t=0.0015557]
1e-06	0.60e-5	70	0	1.533e-07	7.663e-08 [53 bei t=0.0012710]
1e-05	1.00e-5	42	1 [36]	1.522e-06	7.610e-07 [29 bei t=0.0010203]
1e-04	1.50e-5	26	1 [23]	1.537e-05	7.686e-06 [16 bei t=0.0008350]
1e-03	2.00e-5	17	1 [16]	1.515e-04	7.573e-05 [9 bei t=0.0006832]

eps	h0	i	h-min[t,iminh]	h-pha[t,ipha]	h-max[t,imaxh]
1e-10	0.05e-5	668	0.05e-5 [0.05e-5, 1]	0.28e-5 [404]	0.008711 [0.997155,667]
1e-09	0.10e-5	377	0.09e-5 [1.65e-5,17]	0.49e-5 [227]	0.015495 [0.993760,376]
1e-08	0.20e-5	213	0.16e-5 [1.74e-5,10]	0.88e-5 [128]	0.029077 [0.040886,177]
1e-07	0.30e-5	122	0.29e-5 [1.82e-5, 6]	1.54e-5 [72]	0.049090 [0.980305,121]
1e-06	0.60e-5	70	0.52e-5 [2.22e-5, 4]	2.78e-5 [41]	0.087113 [0.913872, 69]
1e-05	1.00e-5	42	0.97e-5 [1.97e-5, 2]	4.66e-5 [23]	0.156120 [0.968282, 41]
1e-04	1.50e-5	26	1.50e-5 [1.50e-5, 1]	8.95e-5 [14]	0.278085 [0.903748, 25]
1e-03	2.00e-5	17	2.00e-5 [2.00e-5 1]	1.26e-4 [8]	0.512226 [0.643619, 16]

Bei $\delta = 0.8$ werden ca. 10% mehr Schritte benötigt als im Fall $\delta = 0.9$.

Tests ergaben bzgl. einer kleinen Schrittzahl den günstigen Wert $\delta = 0.97$ (bis zu 10% weniger Schritte), wobei mit fallender Genauigkeit ε die Unterschiede immer kleiner werden.

delta=0.97

eps	h0	i	zaehl [Schritt]	max(eta(1/2,t(i))-x/y(t(i))) [index-maxabs bei t=...]		
1e-10	0.05e-5	620	0	2.030e-11	1.012e-11	[506 bei t=0.0021268]
1e-09	0.10e-5	350	0	2.075e-10	1.037e-10	[284 bei t=0.0019494]
1e-08	0.20e-5	198	0	2.079e-09	1.040e-09	[159 bei t=0.0017310]
1e-07	0.30e-5	113	2 [4,95]	2.082e-08	1.041e-08	[89 bei t=0.0015121]
1e-06	0.60e-5	66	1 [2]	2.046e-07	1.023e-07	[49 bei t=0.0012112]
1e-05	1.00e-5	39	1 [34]	2.070e-06	1.035e-06	[27 bei t=0.0010151]
1e-04	1.50e-5	25	1 [22]	2.078e-05	1.039e-05	[15 bei t=0.0008394]
1e-03	2.00e-5	17	0	2.022e-04	1.011e-04	[8 bei t=0.0005866]

eps	h0	i	h-min[t,iminh]	h-pha[t,ipha]	h-max[t,imaxh]
1e-10	0.05e-5	620	0.05e-5[0.05e-5, 1]	0.30e-5[375]	0.009391[0.997937,619]
1e-09	0.10e-5	350	0.10e-5[1.66e-5,16]	0.53e-5[211]	0.016700[0.992509,349]
1e-08	0.20e-5	198	0.17e-5[1.68e-5, 9]	0.95e-5[119]	0.029686[0.972801,197]
1e-07	0.30e-5	113	0.30e-5[0.30e-5, 1]	1.66e-5[67]	0.052795[0.947168,112]
1e-06	0.60e-5	66	0.57e-5[2.32e-5, 4]	2.95e-5[38]	0.094298[0.968904, 65]
1e-05	1.00e-5	39	1.00e-5[1.00e-5, 1]	5.37e-5[22]	0.167448[0.886544, 38]
1e-04	1.50e-5	25	1.50e-5[1.50e-5, 1]	9.31e-5[13]	0.301648[0.992901, 24]
1e-03	2.00e-5	17	2.00e-5[2.00e-5 1]	1.54e-4[8]	0.530724[0.890666, 16]

Die Vorteile des Verfahrens `row4_varstep` gegenüber `row4_fixstep` sind sehr deutlich. Die maximale Abweichung $\mathcal{O}(10^{-4})$ erreicht das erste bei $\varepsilon = 10^{-3}$ mit $h_0 = 2 \cdot 10^{-5}$ in 17 Schritten. Die mögliche Schrittweitenvergrößerung kommt dabei bis auf die Schrittweite $h \approx 0.5$. Das Verfahren ohne SWS braucht hier mindestens $N = 16000$ Schritte bei konstanter Schrittweite $h = 6.25 \cdot 10^{-5}$

(3) Betrachten wir noch das steife autonome SysgDGl und AWP

$$\mathbf{x}'(t) = A \mathbf{x}(t), \quad \mathbf{x}(0) = (2, 1, 0)^T, \quad t \in [0, 1], \quad \text{wobei } A = \begin{pmatrix} -1 & -24 & 0 \\ 0 & -25 & 0 \\ 0 & 125 & -150 \end{pmatrix}$$

mit der Lösung $\mathbf{x} = (x, y, z)^T$

$$\begin{aligned} x(t) &= e^{-25t} + e^{-t}, & y(t) &= e^{-25t}, & z(t) &= e^{-25t} - e^{-150t}, \\ \eta_i &= (\eta_{1,i}, \eta_{2,i}, \eta_{3,i}) \approx (x(t_i), y(t_i), z(t_i)). \end{aligned}$$

Die Eigenwerte der Systemmatrix A sind $\lambda_{1,2,3} = -1, -25, -150$. Damit müssen explizite Verfahren mit einer Schrittweitenbeschränkung der Größenordnung $h \approx 10^{-2}$ rechnen. Das SysgDGl besitzt eine transiente und asymptotische Phase der Lösung mit der ersten Phasengrenze bei ungefähr $t = 0.04$, der zweiten Grenze bei ungefähr $t = 0.25$. Dort haben wir $e^{-150 \cdot 0.04} \approx 0.002$, $e^{-25 \cdot 0.25} \approx 0.002$.

Welche Ergebnisse liefern die ROW-Verfahren?

Das ROW-Verfahren mit konstanter Schrittweite in $[0, 1]$ kann nur vernünftigerweise rechnen, wenn die Schrittweite hinreichend klein ist, z.B. $h \approx \mathcal{O}(10^{-2})$, um in der transienten Phase die steilen Lösungsverläufe der Lösungskomponenten gut zu approximieren. In dieser Phase entstehende Fehler wirken sich etwas auf die weitere Rechnung aus. Man bemerkt es z.B. bei der Kontrolle der Verbesserung der Genauigkeit bei der Schrittweitenhalbierung, wo der Faktor 16 nur in wenigen Fällen näherungsweise erhalten wird. Der Fehler bei $t_1 = 1$ in der asymptotischen Phase wird hingegen sehr klein werden.

r_row4_fixstep1.m

```
max=max|e(i,h)|=max|eta(i)-xyz(t(i))|
t1=1
```

N	h	eta(N) (t(N)=1)	e(N,h)= eta(N)-xyz(t(N))	max	e(h)/ e(h/2)
25	0.0400000	0.367879440888	-2.972e-010	2.432e-004	5.9
		0.000000000014	2.314e-013	2.432e-004	
		0.000000000014	2.314e-013	1.243e-001	
50	0.0200000	0.367879441166	-1.980e-011	6.525e-006	13.4
		0.000000000014	6.159e-015	6.525e-006	
		0.000000000014	6.159e-015	2.120e-002	
100	0.0100000	0.367879441185	-3.850e-013	6.980e-009	28.6
		0.000000000014	6.588e-018	6.980e-009	
		0.000000000014	6.588e-018	1.583e-003	
200	0.0050000	0.367879441186	8.952e-013	1.565e-008	49.3
		0.000000000014	-1.477e-017	1.565e-008	
		0.000000000014	-1.477e-017	5.530e-005	
400	0.0025000	0.367879441186	9.726e-013	1.555e-009	30.7
		0.000000000014	-1.468e-018	1.555e-009	
		0.000000000014	-1.468e-018	1.123e-006	
800	0.0012500	0.367879441186	9.754e-013	1.156e-010	5.7
		0.000000000014	-1.092e-019	1.157e-010	
		0.000000000014	-1.092e-019	3.655e-008	
1600	0.0006250	0.367879441186	9.738e-013	6.839e-012	11.8
		0.000000000014	-6.557e-021	6.941e-012	
		0.000000000014	-6.557e-021	6.374e-009	
3200	0.0003125	0.367879441186	1.012e-012	1.012e-012	14.2
		0.000000000014	4.651e-022	4.678e-013	
		0.000000000014	4.651e-022	5.402e-010	
6400	0.0001563	0.367879441186	9.648e-013	1.059e-012	15.4
		0.000000000014	8.734e-022	9.481e-013	
		0.000000000014	8.734e-022	3.790e-011	
12800	0.0000781	0.367879441186	1.066e-012	1.079e-012	
		0.000000000014	1.002e-021	9.755e-013	
		0.000000000014	1.002e-021	1.984e-012	

In einer Auswertung von Testrechnungen des Verfahrens mit SWS kann man die minimale und maximale Schrittweite ermitteln, die an der Phasengrenze und wo die Maximalabweichungen auftreten, dazu die jeweiligen Integrationsstellen als t -Werte und ihre Nummerierung. Die allerletzte Schrittweite wird dabei nicht berücksichtigt.

r_row4_varstep2.m

```

eps      - Toleranz
h0       - Anfangsschrittweite
i        - Anzahl der akzeptierten Schritte in [0,t1]=[0,1]
zaehl    - Anzahl der Misserfolge
[Schritte] - Stellen, wo Schritt wiederholt wurde
max      - =max(|eta(1/2,t(i))-x/y(t(i))|)
[indizes-maxabs bei t=...] Indizes der Stelle der max. Abweichungen und Stelle
          (fuer x/y/z an unterschiedlichen Stellen angenommen)

h-min    - minimale Schrittweite der Rechnung mit Stelle und Nummer
          (meist angenommen nahe dem Intervallanfang, h-min~h0)
h-pha    - Schrittweite an der Phasengrenze t=0.0005 mit Nummer
          (Nummer ~ i/2)
h-max    - maximal Schrittweite der Rechnung mit Stelle und Nummer
          (meist angenommen nahe dem Intervallende)

```

```

delta=0.9
eps      h0      i      zaehl      max(|eta(1/2,3,t(i))-x/y/z(t(i))|)
          [Schritt] [indizes-maxabs bei t=...]
-----
1e-10   0.5e-4   779    0          1.045e-12  1.309e-12  2.929e-12 [(346,626,330) bei t=0.0374274]
1e-09   1.0e-4   438    0          1.252e-11  1.288e-11  2.049e-11 [(315,318,152) bei t=0.1553273]
1e-08   1.5e-4   247    0          1.146e-10  1.166e-10  1.762e-10 [(171,232, 77) bei t=0.1394642]
1e-07   3.0e-4   139    0          1.886e-09  1.945e-09  1.945e-09 [(131,131,131) bei t=0.5567518]
1e-06   6.0e-4   79     0          2.149e-08  2.160e-08  2.935e-08 [( 73, 73, 40) bei t=0.4932409]
1e-05   1.0e-3   45     0          2.199e-07  2.203e-07  4.643e-07 [( 41, 41, 23) bei t=0.4502516]
1e-04   2.0e-3   26     0          2.184e-06  2.184e-06  3.960e-06 [( 22, 22, 13) bei t=0.3157265]
1e-03   4.0e-3   16     0          2.237e-05  2.237e-05  3.393e-05 [( 12, 12, 6) bei t=0.2517801]

```

```

eps      h0      i      h-min[t,iminh]      h-pha[t,ipha] h-max[t,imaxh]
-----
1e-10   0.5e-4   779    0.50e-4[0.50e-4, 1] 2.65e-4[357] 0.011005[0.997401,778]
1e-09   1.0e-4   438    0.95e-4[1.95e-4, 2] 4.73e-4[201] 0.019537[0.990349,437]
1e-08   1.5e-4   247    1.50e-4[1.50e-4, 1] 8.30e-4[113] 0.034739[0.993149,246]
1e-07   3.0e-4   139    3.00e-4[3.00e-4, 1] 1.50e-3[ 64] 0.061404[0.966142,138]
1e-06   6.0e-4   79     5.57e-4[1.16e-3, 2] 2.62e-3[ 36] 0.109311[0.985788, 78]
1e-05   1.0e-3   45     1.00e-3[1.00e-3, 1] 4.94e-3[ 21] 0.189480[0.906824, 44]
1e-04   2.0e-3   26     1.96e-3[3.96e-3, 2] 8.27e-3[ 12] 0.261355[0.784343, 25]
1e-03   4.0e-3   16     3.97e-3[7.97e-3, 2] 1.35e-2[ 7] 0.397893[0.909444, 15]

```

Bei $\delta = 0.8$ werden ca. 10% mehr Schritte benötigt als im Fall $\delta = 0.9$.

Tests ergaben bzgl. einer kleinen Schrittzahl den günstigen Wert $\delta = 1$ (bis zu 10% weniger Schritte), wobei mit fallender Genauigkeit ε die Unterschiede immer kleiner werden.

```

delta=1
eps      h0      i      zaehl      max(|eta(1/2,3,t(i))-x/y/z(t(i))|)
          [Schritt] [indizes-maxabs bei t=...]
-----
1e-10   0.5e-4   701    0          1.582e-12  2.000e-12  3.969e-12 [(526,551,284) bei t=0.1768457]
1e-09   1.0e-4   394    0          1.902e-11  1.937e-11  3.037e-11 [(282,283,134) bei t=0.1526992]
1e-08   1.5e-4   222    0          1.974e-10  2.042e-10  2.598e-10 [(209,209, 69) bei t=0.5374368]
1e-07   3.0e-4   125    0          3.120e-09  3.201e-09  3.201e-09 [(118,118,118) bei t=0.5537086]
1e-06   6.0e-4   71     0          3.340e-08  3.360e-08  5.269e-08 [( 66, 66, 36) bei t=0.5032386]
1e-05   1.0e-3   41     0          3.342e-07  3.345e-07  6.951e-07 [( 37, 37, 21) bei t=0.4406409]
1e-04   2.0e-3   24     0          3.367e-06  3.367e-06  5.913e-06 [( 20, 20, 12) bei t=0.3148906]
1e-03   4.0e-3   14     0          3.409e-05  3.409e-05  5.103e-05 [( 11, 11, 5) bei t=0.2475283]

```

eps	h0	i	h-min[t, iminh]	h-pha[t, ipha]	h-max[t, imaxh]
1e-10	0.5e-4	701	0.50e-4[0.50e-4, 1]	2.93e-4[321]	0.012219[0.993940, 700]
1e-09	1.0e-4	394	1.00e-4[1.00e-4, 1]	5.25e-4[181]	0.021672[0.981667, 393]
1e-08	1.5e-4	222	1.50e-4[1.50e-4, 1]	9.28e-4[102]	0.038454[0.973297, 221]
1e-07	3.0e-4	125	3.00e-4[3.00e-4, 1]	1.69e-3[58]	0.067907[0.941500, 124]
1e-06	6.0e-4	71	6.00e-4[6.00e-4, 1]	3.05e-3[33]	0.120393[0.940609, 70]
1e-05	1.0e-3	41	1.00e-3[1.00e-3, 1]	5.37e-3[19]	0.211352[0.939369, 40]
1e-04	2.0e-3	24	2.00e-3[2.00e-3, 1]	9.11e-3[11]	0.310657[0.861080, 23]
1e-03	4.0e-3	14	4.00e-3[4.00e-3, 1]	1.20e-2[6]	0.185964[0.535267, 13]

Die Vorteile des Verfahrens `row4_varstep` gegenüber `row4_fixstep` sind wiederum deutlich. Die maximale Abweichung $\mathcal{O}(10^{-5})$ erreicht das erste bei $\varepsilon = 10^{-3}$ mit $h_0 = 4 \cdot 10^{-3}$ in 14 Schritten. Die mögliche Schrittweitenvergrößerung kommt dabei bis auf die Schrittweite $h \approx 0.5$. Das Verfahren ohne SWS braucht hier mindestens $N = 200$ Schritte bei konstanter Schrittweite $h = 0.005$

Kapitel 9

Übungsserien

In mehreren Abschnitten sind schon einzelne Übungsaufgaben ergänzt worden. Diese sowie die angefügten Übungsserien dienen der Vertiefung des Stoffes, geben Hinweise zur rechen-technischen Umsetzung der Algorithmen und sind Anleitung zum Experimentieren. Sie werden sich an den behandelten Themen dieses Gebietes der Mathematik orientieren und sich im Wesentlichen auf AWP von gewöhnlichen Differentialgleichungen und ihre numerische Lösung beziehen.

Die Serien 1 und 2 sind ein Einstieg und eine Wiederholung von stofflichen Aspekten. Sie beinhalten die Darstellung und den Umgang mit Modellen der Mathematik, Physik und Technik, wo Differentialgleichungen auftreten. Dabei geht es mehr um Theorie und die analytische Lösung.

In den weiteren Übungsserien stehen dann die numerischen Verfahren zur Lösung von gDGL und Systemen dieser als AWP im Mittelpunkt. Dazu gehören die Analyse der Methoden, zahlreiche Verfahren, die Untersuchungen der Algorithmen und ihrer Eigenschaften sowie die numerischen Rechnungen und Bewertungen der Ergebnisse. Um den unmittelbaren Bezug zum Algorithmus zu haben, werden in den Aufgabenstellungen zum Teil noch einmal der Formelapparat oder die Schemata notiert.

Mit jeder Lösung einer Aufgabe vertieft der Anwender sein Wissen und seine Erkenntnisse im Umgang mit Differentialgleichungen. Die Numerik der gDGL sowie die Anschaulichkeit werden dabei durch den Einsatz von CAS sehr gut unterstützt. So können mit Maple und MATLAB, aber auch mit anderer Software, umfangreiche numerische und grafische Auswertungen durchgeführt werden.

Zu den Aufgaben der Übungsserien wurden umfangreiche Rechnungen gemacht. Anhand einer ausgewählten interessanten Aufgabe zu den MSV zeigen wir eine mögliche Vorgehensweise bei der Lösung.

Aufgabe

Gegeben sind zwei lineare MSV.

– 4-stufiges explizites Milne-Verfahren 4. Ordnung

$$\eta_{n+1} = \eta_{n-3} + \frac{h}{3}(8f_n - 4f_{n-1} + 8f_{n-2}), \quad n = 3, 4, \dots, \quad f_n = f(t_n, \eta_n),$$

– 2-stufiges implizites Adams-Störmer-Verfahren 4. Ordnung

(Milne-Simpson(4), Simpson-Regel)

$$\eta_{n+1} = \eta_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}), \quad n = 1, 2, \dots .$$

- Man bestimme die erzeugenden Polynome $\rho(\xi)$ bzw. $\sigma(\xi)$.
- Man bestimme die Konsistenzordnungen.
- Man bestimme das charakteristische Polynom $\phi(\xi; z)$ und die Konsistenzordnung des aus den beiden Formeln bestehenden PC-Verfahrens von Milne.
- Man überprüfe das folgende Wurzelkriterium:

Das erzeugende Polynom $\rho(\xi)$ hat keine Nullstellen ξ mit $|\xi| > 1$ und alle Nullstellen mit $|\xi| = 1$ sind einfach.

Sind die Verfahren konvergent?

Milne(4) = 4-stufiges explizites Milne-Verfahren 4. Ordnung

Man notiert das MSV auch in der Gestalt

$$\eta_{i+4} - \eta_i = \frac{h}{3}(8f_{i+3} - 4f_{i+2} + 8f_{i+1}), \quad r = 4, \quad a_r = 1.$$

Es ist instabil.

Die zugehörigen charakteristischen Polynome sind

$$\begin{aligned} \rho(\xi) &= \xi^4 - 1, \\ \sigma(\xi) &= \frac{1}{3}(8\xi^3 - 4\xi^2 + 8\xi), \\ \phi(\xi; z) &= \rho(\xi) - z\sigma(\xi) = \xi^4 - \frac{z}{3}(8\xi^3 - 4\xi^2 + 8\xi) - 1, \end{aligned}$$

wobei $z = h\lambda$. Das Stabilitätskriterium des Satzes 6.5 zu $\rho(\xi) = \xi^4 - 1$ ist erfüllt, denn seine vier Nullstellen sind $\pm 1, \pm i$. Dieser Umstand bzw. auch die Wachstumsbeschränkung (6.55) garantieren i. Allg. jedoch nicht die absolute Stabilität.

Wir untersuchen die Gleichung $\phi(\xi; z) = 0$ und berechnen das Stabilitätsgebiet \mathcal{E} durch die ‘‘Randbedingung‘‘

$$z = \frac{\rho(\xi)}{\sigma(\xi)}.$$

Natürlich kann man auch die vier Nullstellen $\xi_{1,2,3,4}(z)$ des Polynoms 4. Grades $\xi^4 - \frac{z}{3}(8\xi^3 - 4\xi^2 + 8\xi) - 1$ als Funktionen von z betrachten. Es gelten

$$\xi_{1,2}(0) = \pm 1, \quad \xi_{3,4}(0) = \mp i$$

und für andere z -Werte entweder $\xi_1(z) > 1, |\xi_{3,4}| > 1$ oder $\xi_4(z) < -1$.

Es gibt also keinen Bereich, wo alle Beträge kleiner als Eins sind. Für reelles z sieht man es in der Abbildung 9.3.

Für die Konsistenz des expliziten MSV der Form

$$a_r \eta_{i+r} + \sum_{k=0}^{r-1} a_k \eta_{i+k} = h \sum_{k=0}^{r-1} b_k f_{i+k}, \quad i \geq 0, \quad a_r = 1, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben,}$$

und ihre Ordnung p bezieht man sich auf den Satz 6.1.

Zu überprüfen sind die Beziehungen $c_0 = c_1 = \dots = c_p = 0$, wobei

$$\begin{aligned} c_0 &= a_0 + a_1 + \dots + a_r, \quad a_r = 1, \\ c_1 &= (a_1 + 2a_2 + \dots + ra_r) - (b_0 + b_1 + \dots + b_r), \\ c_k &= \frac{1}{k!}(a_1 + 2^k a_2 + \dots + r^k a_r) - \frac{1}{(k-1)!}(b_1 + 2^{k-1} b_2 + \dots + r^{k-1} b_r), \quad k \geq 2. \end{aligned}$$

Mit den Koeffizienten $a = (1, 0, 0, 0, -1)$ und $b = (\frac{8}{3}, -\frac{4}{3}, \frac{8}{3}, 0)$ erhält man $c_0 = c_1 = c_2 = c_3 = c_4 = 0$, $c_5 = \frac{28}{90} \neq 0$ und damit die Konsistenzordnung 4.

Die Bestimmung des Stabilitätsgebiets \mathcal{E} erfolgt durch die Bedingung

$$z = \frac{\rho(\xi)}{\sigma(\xi)} = \frac{3(\xi^4 - 1)}{8\xi^3 - 4\xi^2 + 8\xi}, \quad \xi = e^{i\varphi}, \quad \varphi \in [0, 2\pi).$$

Neben der 1-Kontur ($|\xi| = |e^{i\varphi}| = 1$) können weitere Konturen zu den Werten $c \in (0, 2)$ mit $c = |ce^{i\varphi}|$ hinzugefügt werden. Dadurch erkennt man neben dem Stabilitätsgebiet, in welchen Teilgebieten die Funktion $\phi(\xi; z)$ ihre weiteren Nullstellen $\xi_k(z)$ mit Betrag größer bzw. kleiner als Eins hat. Insbesondere, wenn die 1-Kontur irgendwelche Schleifen macht, ist es interessant, was innerhalb dieser passiert.

Hier liegt die 1-Kontur genau auf der imaginären Achse, denn man hat

$$z(e^{i\varphi}) = i \frac{3 \sin(2\varphi)}{2(4 \cos(\varphi) - 1)}.$$

In jedem anderen Gebiet verlaufen Konturen für $c < 1$ und zugleich für $c > 1$, sodass kein stabiles Verhalten vorliegt.

Die Auswertung von z kann entweder mittels `solve`, `subs`, `double`, `real`, `imag`, `plot` oder `polyval`, `real`, `imag`, `plot` vorgenommen werden.

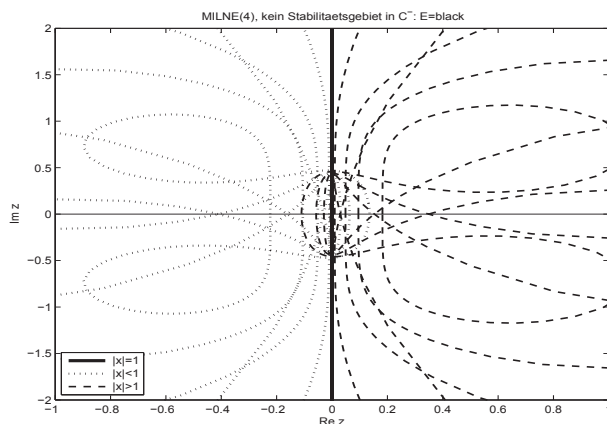


Abb. 9.1

Datei `milne4_stab27sw.ps`,

Konturen $z = \frac{\rho(\xi)}{\sigma(\xi)}$,
 $z \in [-1, 1] \times [-2, 2]$,

$\xi = ce^{i\varphi}$, $\varphi = \frac{2\pi k}{m}$,

$k = 0(1)m$, $m = 200$,

$c = 0.8, 0.9, 0.95, 0.99, 1,$

$1.01, 1.05, 1.1, 1.2$

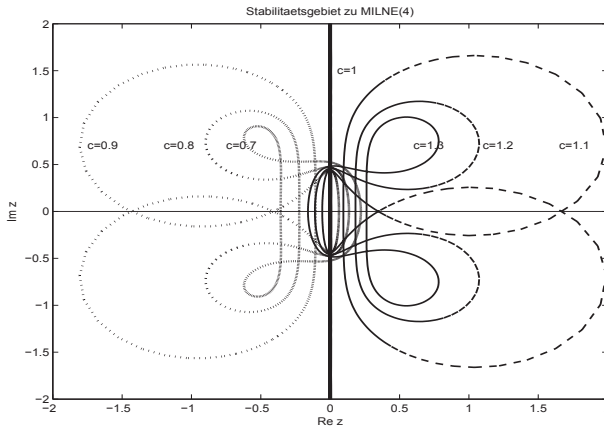


Abb. 9.2

Datei *milne4_stab28sw.ps*,
 Konturen $z = \frac{\rho(\xi)}{\sigma(\xi)}$,
 $z \in [-2, 2] \times [-2, 2]$,
 $z = \text{polyval}(\alpha, w) / \text{polyval}(\beta, w)$,
 $w = c e^{i\varphi}$, $\varphi = \frac{2\pi k}{m}$,
 $k = 0(1)m$, $m = 400$,
 $c = 0.7(0.1)1.3$

Die vier Nullstellenfunktionen $\xi_{1,2,3,4}(z)$ werden gezeichnet, wenn sie reellwertig sind, oder im komplexen Fall plottet man ihren Betrag.

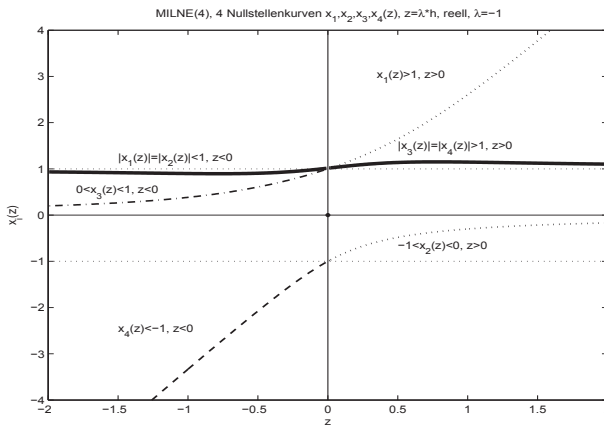


Abb. 9.3

Datei *milne4_stab29sw.ps*,
 Nullstellenkurven
 $\xi_{1,2,3,4}(z)$ bzw. $|\xi_{1,2,3,4}(z)|$
 $\in [-2, 2] \times [-4, 4]$,
 reelles $z = h\lambda \in [-2, 2]$,
 $\lambda = -1$

Dazu werden die MATLAB-Anweisungen angegeben. Für die weiteren Fälle ist eine ähnliche Vorgehensweise möglich. Grafiken in Farbe unterstützen das Verständnis.

```

disp('6) MSV(p), PCV und ihre charakteristischen Polynome rho(x), sigma(x)')
disp(' und phi(x,z)=rho(x)-z*sigma(x), |x|<1 -> z=la*h in C')
disp(' Test-Dgl y'=la*y, la<0')
disp(' ')
disp('Milne(4)')
disp('Instabilitaet von expliziten Milne-MSV 4.Ordnung, 4-stufig')
disp('eta(n+1) = eta(n-3)+h/3(8f(n)-4f(n-1)+8f(n-2)), n=3,4,...,')
disp(' eta(0),eta(1),eta(2),eta(3) gegeben')
% Parameterschema zum expliziten Milne(4), 4-stufig
m_alpha = [ 1 0 0 0 -1 ];
m_beta = [ 0 8/3 -4/3 8/3 0 ];

r = 4 % Stufenzahl
p = 4 % Ordnung
al = m_alpha(1:r+1)
be = m_beta(2:r+1)
    
```

```

% Wurzelkriterium
% rho(x) hat Wurzeln mit |x|<=1 und bei |x|=1 ist x einfach
% rho(x)=al(1)x^r+al(2)x^(r-1)+...+al(r+1)x^0
syms x rho4 sigma4 z phi
rho4 = al(1)*x^r+al(2)*x^(r-1)+al(3)*x^(r-2)+al(4)*x^(r-3)+al(5)*x^(r-4)
r4 = solve(rho4,x)
vpa(r4)
% sigma(x)=be(1)x^(r-1)+be(2)x^(r-2)+...+be(r)x^0
sigma4 = be(1)*x^(r-1)+be(2)*x^(r-2)+be(3)*x^(r-3)+be(4)*x^(r-4)
s4 = solve(sigma4,x)
vpa(s4)
phi = rho4-z*sigma4
spx = solve(phi,z) % z=rho4(x)/sigma4(x)
subs(spx,x,1)
double(subs(spx,x,1))
%
spz = solve(phi,x)
subs(spz,z,0)
% 4 Nullstellenkurven xi(z), Verlauf bei z reell, |xi(z)|<=1?
spz(1)
spz(2)
spz(3)
spz(4)
subs(spz(1),z,0)
subs(spz(1),z,0:0.2:1)
[subs(spz(1),z,0:0.2:1)]';
%[subs(spz,z,0:0.2:1)]'

[subs(spz(1),z,0:0.2:1)
subs(spz(2),z,0:0.2:1)
subs(spz(3),z,0:0.2:1)
subs(spz(4),z,0:0.2:1)]'

% Konsistenzordnung, r-stufiges MSV
% c(0) = a(0)+a(1)+...+a(r)
% c(k) = 1/k!*(a(1)+2^k*a(2)+...+r^k*a(r))
% -1/(k-1)!*(b(1)+2^(k-1)*b(2)+...+r^(k-1)*b(r)), k>=1
% K0 = 0(h^p) <-> c(0)=c(1)=...=c(p)=0
% hier: a(j) = m_alpha(r+1-j), j=0,1,...,r
% b(j) = m_beta(r+1-j)
v = [1:r];
vh2 = v;
ma = m_alpha(r+1:-1:1)
mb = m_beta(r+1:-1:1)
ch(1) = sum(m_alpha)
ch(2) = sum(ma(2:r+1).*vh2)-sum(m_beta)
for j=3:6
    vh1 = vh2;
    vh2 = vh1.*v;
    ch(j) = 1/factorial(j-1)*sum(ma(2:r+1).*vh2)...
        -1/factorial(j-2)*sum(mb(2:r+1).*vh1)
end
% ch(5)=c(4)=0, ch(6)=c(5)>0 -> Ordnung p=4

% Untersuchung der Stabilitaet von Milne(4)
% z=rho4(x)/sigma4(x), x=c*exp(1i*phi), 0<=phi<=2*pi, |x|=c, 0<c<2
m = 200;
% Achtung
% z=a+bi mit a=0, dann double(z)=>b, Fehler!!!

%for k=0:m, pp(k+1)=double(subs(spx,x,exp(2*pi*k/m*1i))); end
for k=0:m

```

```

    pph=subs(spx,x,exp(2*pi*k/m*1i));
    pp(k+1)=double(real(pph))+double(imag(pph))*1i
end

for k=0:m, q1(k+1)=double(subs(spx,x,0.8*exp(2*pi*k/m*1i))); end
for k=0:m, q2(k+1)=double(subs(spx,x,0.9*exp(2*pi*k/m*1i))); end
for k=0:m, q3(k+1)=double(subs(spx,x,0.95*exp(2*pi*k/m*1i))); end
for k=0:m, q4(k+1)=double(subs(spx,x,0.99*exp(2*pi*k/m*1i))); end
for k=0:m, p1(k+1)=double(subs(spx,x,1.2*exp(2*pi*k/m*1i))); end
for k=0:m, p2(k+1)=double(subs(spx,x,1.1*exp(2*pi*k/m*1i))); end
for k=0:m, p3(k+1)=double(subs(spx,x,1.05*exp(2*pi*k/m*1i))); end
for k=0:m, p4(k+1)=double(subs(spx,x,1.01*exp(2*pi*k/m*1i))); end

% Test Ausgabe
%for k=1:201, fprintf('%+9.6e %+9.6ei\n',real(pp(k)),imag(pp(k))); end
%subs(spx,x,exp(2*pi*25/m*1i))
%double(subs(spx,x,exp(2*pi*25/m*1i)))

figure(27)
plot(real(pp),imag(pp),'b-', 'Linewidth',2)
title('MILNE(4), kein Stabilitaetsgebiet in  $C^{\{-\}}$ : E=blue');
xlabel('Re z');
ylabel('Im z');
hold on
plot(real(q1),imag(q1),'c:');
plot(real(p1),imag(p1),'r:');
legend(' |x|=1', ' |x|<1', ' |x|>1', 'Location', 'SouthWest');
%legend(' |x|=1', ' |x|<1', ' |x|>1',3);
axis([-1 1 -2 2])
plot(real(q2),imag(q2),'c:');
plot(real(q3),imag(q3),'c-.');
plot(real(q4),imag(q4),'c--');
plot(real(p2),imag(p2),'r:');
plot(real(p3),imag(p3),'r-.');
plot(real(p4),imag(p4),'r--');
plot([-1 1],[0 0],'k-', [0 0],[-2 2],'k-')
print milne4_stab27.ps -dpsc
hold off
pause

% allgemein fuer Milne(4)
a = 2;
b = 2;
ta = [-a,a];
tb = [-b,b];

figure(28)
plot(ta,[0,0],'k',[0,0],tb,'k');
axis([-a a -b b]);
hold on

% Stabilitaetsgebiet
ch = 0.1;
cg = 0.7;
alpha = m_alpha(1:r+1);
beta = m_beta(1:r+1);
m = 400;
dphi = 2*pi/m;
for k = 1:7
    c = cg+(k-1)*ch;
    for i = 0:1:m
        phi_i = i*dphi;

```



```

w = c*exp(1i*phi_i);
zz = polyval(alpha,w)/polyval(beta,w);
zr = real(zz);
zi = imag(zz);
if i==0
    z1r = zr;
    z1i = zi;
end;
% fprintf('%12.9e %12.9e\n',zr,zi);
switch k
    case 1, plot([zr,z1r],[zi,z1i],'k');
    case 2, plot([zr,z1r],[zi,z1i],'b');
    case 3, plot([zr,z1r],[zi,z1i],'r');
    case 4, plot([zr,z1r],[zi,z1i],'g','LineWidth',2);
    case 5, plot([zr,z1r],[zi,z1i],'m');
    case 6, plot([zr,z1r],[zi,z1i],'c');
    case 7, plot([zr,z1r],[zi,z1i],'y');
end;
z1r = zr;
z1i = zi;
%pause(0.001)
end;
end;

title('Stabilitaetsgebiet zu MILNE(4)');
xlabel('Re z');
ylabel('Im z');
text(0.05,1.5,'c=1');
text(0.6,0.7,'c=1.3');
text(1.1,0.7,'c=1.2');
text(1.65,0.7,'c=1.1');
text(-1.75,0.7,'c=0.9');
text(-1.2,0.7,'c=0.8');
text(-0.75,0.7,'c=0.7');
print milne4_stab28.ps -dpsc
hold off

disp(' ')
disp('4 Nullstellenkurven x1,2,3,4(z), z=lambda*h, reell, lambda=-1')
a = 2;
b = 4;
h = 0.1;
xx = -a:h:a;
xxl = -a:0.01:-0.01;
xxr = 0.01:0.01:a;
ta = [-a,a];
tb = [-b,b];
% Test zum Verlauf der Nullstellenkurven, wann reell, komplex, ...
%[xx',subs(spz,z,xx)']
[xx
 subs(spz(1),z,xx)
 subs(spz(2),z,xx)
 subs(spz(3),z,xx)
 subs(spz(4),z,xx)]'

figure(29)
plot(ta,[0,0],'k',[0,0],tb,'k');
axis([-a a -b b]);
title('MILNE(4), 4 Nullstellenkurven x_1,x_2,x_3,x_4(z),...
      z=\lambda*h, reell, \lambda=-1');
xlabel('z');
ylabel('x_i(z)');

```

```

hold on
fplot('1', [-a a], 'k:');
fplot('-1', [-a a], 'k:');
plot(0,0, 'k.', 'MarkerSize', 12);
plot(xxl, abs(double(subs(spz(1), z, xxl))), 'r-');
plot(xxr, double(real(subs(spz(1), z, xxr))), 'r-');
plot(xxl, 0.03+abs(double(subs(spz(2), z, xxl))), 'm-');
plot(xxr, double(real(subs(spz(2), z, xxr))), 'm-');
plot(xxl, double(real(subs(spz(3), z, xxl))), 'b-');
plot(xxr, abs(double(subs(spz(3), z, xxr))), 'b-');
plot(xxl, double(real(subs(spz(4), z, xxl))), 'c-');
plot(xxr, 0.03+abs(double(subs(spz(4), z, xxr))), 'c-');
text(-1.5, 1.2, '|x_1(z)|=|x_2(z)|<1, z<0');
text(0.35, 3.0, 'x_1(z)>1, z>0');
text(0.5, -0.7, '-1<x_2(z)<0, z>0');
text(-1.5, -2.5, 'x_4(z)<-1, z<0');
text(-1.8, 0.5, '0<x_3(z)<1, z<0');
text(0.5, 1.45, '|x_3(z)|=|x_4(z)|>1, z>0');
print milne4_stab29.ps -dpsc
hold off

```

AS(4) = 2-stufiges implizites Adams-Störmer-Verfahren 4. Ordnung, Milne-Simpson(4)
 Man notiert es auch in der Gestalt

$$\eta_{i+2} - \eta_i = \frac{h}{3} (f_{i+2} + 4f_{i+1} + f_i), \quad r = 2, \quad a_r = 1.$$

Es ist instabil.

Detaillierte Untersuchungen zur Genauigkeit findet man im Abschnitt 6.2 Bsp. 6.5 (b) sowie zur Stabilität im Abschnitt 7.2.1.

Die zugehörigen charakteristischen Polynome sind

$$\begin{aligned} \rho(\xi) &= \xi^2 - 1, \quad \sigma(\xi) = \frac{1}{3} (\xi^2 + 4\xi + 1), \\ \phi(\xi; z) &= \rho(\xi) - z\sigma(\xi) = (1 - \frac{z}{3})\xi^2 - \frac{4z}{3}\xi - \frac{z}{3}, \quad z = h\lambda. \end{aligned}$$

Das Stabilitätskriterium des Satzes 6.5 zu $\rho(\xi) = \xi^2 - 1$ ist erfüllt, denn seine zwei Nullstellen sind ± 1 . Dieser Umstand bzw. auch die Wachstumsbeschränkung (6.55) garantieren i. Allg. jedoch nicht die absolute Stabilität.

Wir untersuchen $\phi(\xi; z) = 0$ und berechnen das Stabilitätsgebiet \mathcal{E} durch die "Randbedingung"

$$z = \frac{\rho(\xi)}{\sigma(\xi)} = \frac{3(\xi^2 - 1)}{\xi^2 + 4\xi + 1}, \quad \xi = e^{i\varphi},$$

und erhalten nach wenigen Umformungen

$$z = i \frac{3 \sin(\varphi)}{2 + \cos(\varphi)}, \quad \varphi \in [0, 2\pi),$$

und damit nur imaginäre Werte z im Intervall $[-i\sqrt{3}, i\sqrt{3}]$.

Damit ist das Gebiet \mathcal{E} , wenn es sich innerhalb der Kontur befinden soll, eine leere Menge. Da aber auch außerhalb der Kontur kein z liegt, wo $|\xi_{1,2}(z)| < 1$ gilt, ist das Stabilitätsgebiet \mathcal{E} damit leer.

Natürlich kann man auch die beiden Nullstellen

$$\xi_{1,2}(z) = \frac{2z \pm \sqrt{3z^2 + 9}}{3 - z}$$

der quadratischen Gleichung $(1 - \frac{z}{3})\xi^2 - \frac{4z}{3}\xi - (1 + \frac{z}{3}) = 0$ als Funktionen von z betrachten. Es gelten $\xi_1(0) = 1$, $\xi_2(0) = -1$ und für andere z -Werte stets entweder $|\xi_1(z)| > 1$ oder $\xi_2(z) < -1$. Es gibt also keinen Bereich, wo beide Beträge kleiner als Eins sind. Für reelles z sieht man es in der Abbildung 9.6.

Für die Konsistenz des impliziten MSV der Form

$$a_r \eta_{i+r} + \sum_{k=0}^{r-1} \alpha_k \eta_{i+k} = h \sum_{k=0}^r \beta_k f_{i+k}, \quad a_r = 1, \quad i \geq 0, \quad a_r = 1, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben,}$$

gehen wir so vor wie beim Milne(4).

Mit den Koeffizienten $a = (1, 0, -1)$ und $b = (\frac{1}{3}, \frac{4}{3}, \frac{1}{3})$ erhält man $c_0 = c_1 = c_2 = c_3 = c_4 = 0$, $c_5 = -\frac{1}{90} \neq 0$ und damit die Konsistenzordnung 4.

Auch bei der Bestimmung des Stabilitätsgebiets \mathcal{E} kann man die Vorgehensweise aus dem Verfahren Milne(4) verwenden.

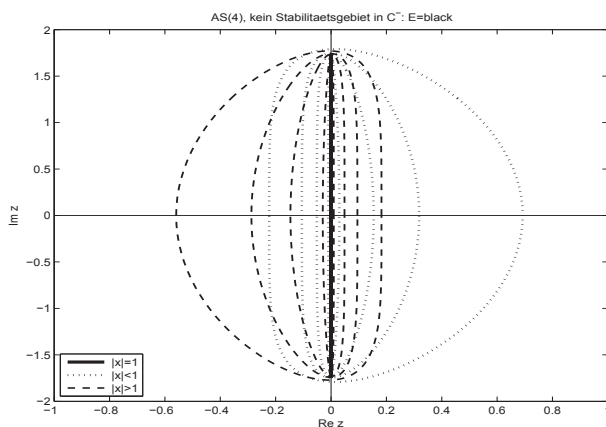


Abb. 9.4

Datei `as4_stab30sw.ps`,

Konturen $z = \frac{\rho(\xi)}{\sigma(\xi)}$,

$z \in [-1, 1] \times [-2, 2]$,

$\xi = c e^{i\varphi}$, $\varphi = \frac{2\pi k}{m}$,

$k = 0(1)m$, $m = 200$,

$c = 0.8, 0.9, 0.95, 0.99, 1,$

$1.01, 1.05, 1.1, 1.2$

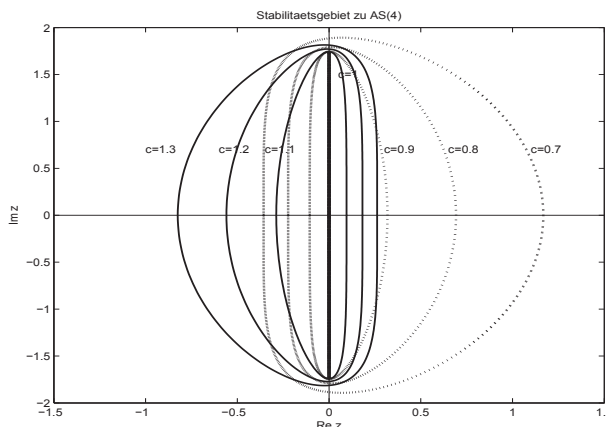


Abb. 9.5

Datei `as4_stab31sw.ps`,

Konturen $z = \frac{\rho(\xi)}{\sigma(\xi)}$,

$z \in [-1.5, 1.5] \times [-2, 2]$,

$z = \text{polyval}(\alpha, w) /$

$\text{polyval}(\beta, w)$,

$w = c e^{i\varphi}$, $\varphi = \frac{2\pi k}{m}$,

$k = 0(1)m$, $m = 400$,

$c = 0.7(0.1)1.3$

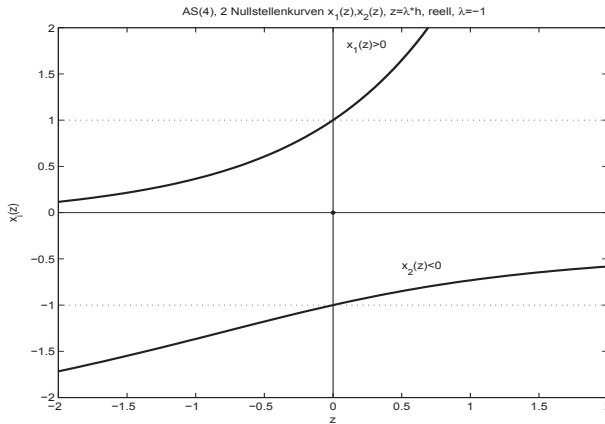


Abb. 9.6

Datei *as4_stab32sw.ps*,

Nullstellenkurven

$$\xi_1(z) = \frac{2z + \sqrt{3z^2 + 9}}{3 - z} \text{ und}$$

$$\xi_2(z) = \frac{2z - \sqrt{3z^2 + 9}}{3 - z}$$

$\in [-2, 2] \times [-2, 2]$,

reelles $z = h\lambda \in [-2, 2]$,

$\lambda = -1$

$$PCV(4) = Milne(4) + AS(4)$$

Man notiert das PCV auch in der allgemeinen Gestalt

$$\eta_{i+r}^{(P)} = - \sum_{k=0}^{r-1} a_k \eta_{i+k} + h \sum_{k=0}^{r-1} b_k f_{i+k},$$

$$\eta_{i+r} = - \sum_{k=0}^{r-1} \alpha_k \eta_{i+k} + h \beta_r f(t_{i+r}, \eta_{i+r}^{(P)}) + h \sum_{k=0}^{r-1} \beta_k f_{i+k}.$$

Es ist instabil.

Betrachtungen zur Konsistenzordnung von PCV findet man in Abschnitt 6.1.3.

Man rechnet nach, dass der lokale Diskretisierungsfehler $\tau(t_i, \eta_i, h)$ bei den gegebenen Koeffizienten die Ordnung $\mathcal{O}(h^4)$ hat.

$$\begin{aligned} \tau(t_i, \eta_i, h) &= \\ &= \frac{1}{h} (z_{i+4} - z_{i+2}) - \left(\frac{1}{3} (z_{i+4}^{(P)})' + \frac{4}{3} f_{i+3} + \frac{1}{3} f_{i+2} \right) \\ &= \frac{1}{h} (z_{i+4} - z_{i+2}) - \frac{1}{3} \left(z_i' + \frac{h}{3} (8z_{i+3}'' - 4z_{i+2}'' + 8z_{i+1}'') \right) - \frac{4}{3} z_{i+3}' - \frac{1}{3} z_{i+2}'. \end{aligned}$$

Mit Hilfe von Taylor-Entwicklungen um den Punkt (t_i, η_i)

$$z(t_{i+k}) = z(t_i) + kh z'(t_i) + \frac{(kh)^2}{2!} z''(t_i) + \frac{(kh)^3}{3!} z'''(t_i) + \dots,$$

$$z'(t_{i+k}) = z'(t_i) + kh z''(t_i) + \frac{(kh)^2}{2!} z'''(t_i) + \frac{(kh)^3}{3!} z''''(t_i) + \dots,$$

$$z''(t_{i+k}) = z''(t_i) + kh z'''(t_i) + \frac{(kh)^2}{2!} z''''(t_i) + \frac{(kh)^3}{3!} z'''''(t_i) + \dots$$

und Vereinfachungen folgen der Fehler und seine Ordnung.

Zur Stabilität betrachten wir wieder die zugehörigen charakteristischen Polynome. Aus

$$\eta_{n+1}^{(P)} = \eta_{n-3} + \frac{h}{3} (8f_n - 4f_{n-1} + 8f_{n-2}),$$

$$\eta_{n+1} = \eta_{n-1} + \frac{h}{3} (f(t_{n+1}, \eta_{n+1}^{(P)}) + 4f_n + f_{n-1}),$$

ergeben sich mit $f(t, x) = \lambda x$ und $\eta_n = \xi^n$ die Beziehungen

$$\begin{aligned} \eta_{n+1} - \eta_{n-1} &= \frac{h}{3} (f(t_{n+1}, \eta_{n-3}) + \frac{h}{3} (8f_n - 4f_{n-1} + 8f_{n-2})) + 4f_n + f_{n-1}) \\ &= \frac{h}{3} (\lambda (\eta_{n-3} + \frac{h}{3} (8\lambda\eta_n - 4\lambda\eta_{n-1} + 8\lambda\eta_{n-2})) + 4\lambda\eta_n + \lambda\eta_{n-1}) \\ &= \frac{h\lambda}{3} (\eta_{n-3} + 4\eta_n + \eta_{n-1}) + \frac{(h\lambda)^2}{9} (8\eta_n - 4\eta_{n-1} + 8\eta_{n-2}) \\ &= \frac{z}{3} (4\eta_n + \eta_{n-1} + \eta_{n-3}) + \frac{z^2}{9} (8\eta_n - 4\eta_{n-1} + 8\eta_{n-2}), \quad z = h\lambda, \\ \xi^{n+1} - \xi^{n-1} &= \frac{z}{3} (4\xi^n + \xi^{n-1} + \xi^{n-3}) + \frac{z^2}{9} (8\xi^n - 4\xi^{n-1} + 8\xi^{n-2}), \\ \xi^4 - \xi^2 &= \frac{z}{3} (4\xi^3 + \xi^2 + 1) + \frac{z^2}{9} (8\xi^3 - 4\xi^2 + 8\xi) \end{aligned}$$

und damit die charakteristischen Polynome

$$\begin{aligned} \rho(\xi) &= \xi^4 - \xi^2, \\ \sigma_1(\xi) &= \frac{1}{3} (4\xi^3 + \xi^2 + 1), \\ \sigma_2(\xi) &= \frac{1}{9} (8\xi^3 - 4\xi^2 + 8\xi), \\ \phi(\xi; z) &= \rho(\xi) - z\sigma_1(\xi) - z^2\sigma_2(\xi), \quad z = h\lambda. \end{aligned}$$

Die Umstellung von $\phi(\xi; z) = 0$ nach z liefert die beiden Wurzeln

$$\begin{aligned} z_1 &= \frac{-\sigma_1(\xi) + \sqrt{\sigma_1(\xi)^2 + 4\sigma_2(\xi)\rho(\xi)}}{2\sigma_2(\xi)}, \\ z_2 &= \frac{-\sigma_1(\xi) - \sqrt{\sigma_1(\xi)^2 + 4\sigma_2(\xi)\rho(\xi)}}{2\sigma_2(\xi)}. \end{aligned}$$

Für die Charakterisierung des Stabilitätsgebiets plotten wir mehrere Konturen zu den Argumenten $\xi = ce^{i\varphi}$, $\varphi = \frac{2\pi k}{m}$, $k = 0(1)m$, $m = 1000$, wobei c den Wert 1 sowie andere Werte annimmt.

Die Plots enthalten der Reihe nach die einzelnen Kurven $z_1(\xi)$, $z_2(\xi)$ sowie dann beide zusammen.

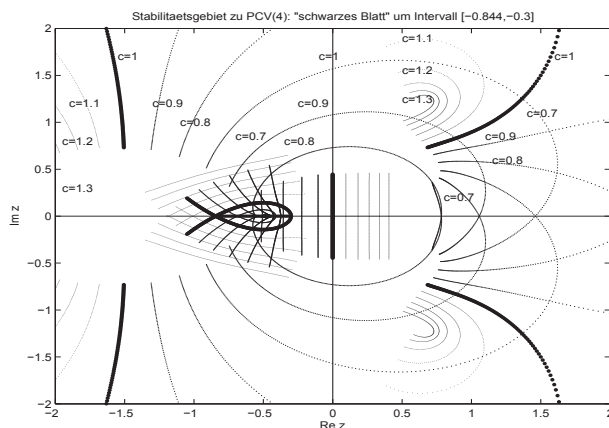


Abb. 9.7

Datei *pc4_stab33sw.ps*,

Kurve $z_1(\xi)$ mit

$$\xi = ce^{i\varphi}, \quad \varphi = \frac{2\pi k}{m},$$

$k = 0(1)m$, $m = 1000$,

$\in [-2, 2] \times [-2, 2]$,

$c = 0.5(0.1)1.5$

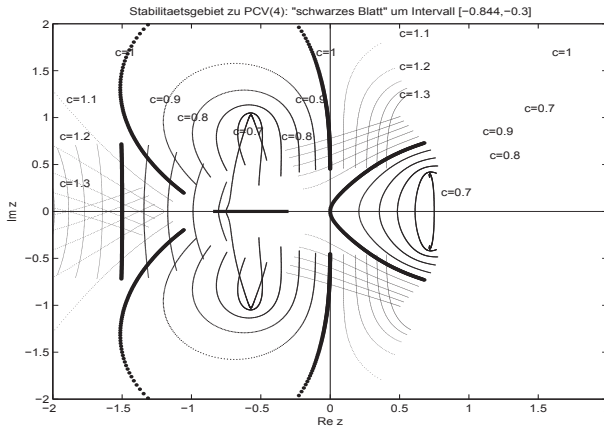


Abb. 9.8

Datei *pc4_stab34sw.ps*,
 Kurve $z_2(\xi)$ mit
 $\xi = c e^{i\varphi}$, $\varphi = \frac{2\pi k}{m}$,
 $k = 0(1)m$, $m = 1000$,
 $\in [-2, 2] \times [-2, 2]$,
 $c = 0.5(0.1)1.5$

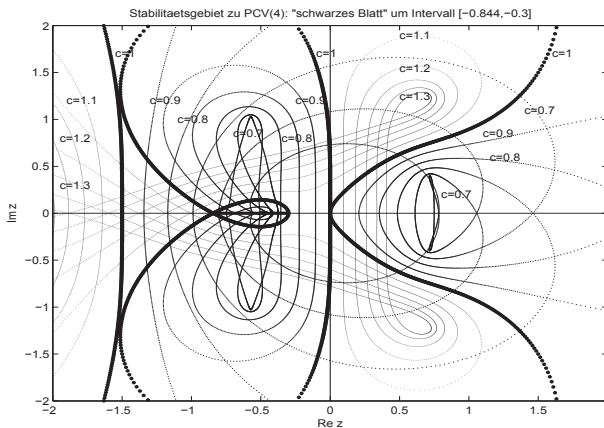


Abb. 9.9

Datei *pc4_stab35sw.ps*,
 Kurven $z_{1,2}(\xi)$ mit
 $\xi = c e^{i\varphi}$, $\varphi = \frac{2\pi k}{m}$,
 $k = 0(1)m$, $m = 1000$,
 $\in [-2, 2] \times [-2, 2]$,
 $c = 0.5(0.1)1.5$

Erst wenn beide Nullstellenkurven gemeinsam in einer Abbildung gezeichnet werden, erhält man den kompletten Konturenverlauf. Das entspricht dem Plot einer Kontur mittels der impliziten Formel $\rho(\xi) - z\sigma_1(\xi) - z^2\sigma_2(\xi) = 0$.

Es zeigt sich, dass die anfangs genannte Instabilität nicht ganz korrekt ist.

Die Grafiken geben Aufschluss darüber.

Es gibt im Reellen ein kleines Intervall und symmetrisch drum herum einen kleinen Bereich, durch den nur Konturen mit $c < 1$ verlaufen. Das Gebiet hat die Gestalt eines kleinen Blattes bzw. einer Schlinge.

Dieses Intervall ist $(-0.844, -0.3)$. Im reellen Fall könnte man also die Schrittweite h so wählen, dass bei gegebenem $\lambda < 0$ der Wert z zu diesem Intervall gehört. Der Wertebereich von zulässigen h ist natürlich nicht besonders groß. Und ob man bei beliebigen gDGI mit der Schrittweite des PCV in diesen Bereich kommt, ist fraglich.

Bei der Implementation in MATLAB wurde nicht unbedingt Wert auf Eleganz gelegt, sondern mehr auf die Verständlichkeit. In der Testphase ist es angebracht, geeignete Zwischenausgaben und Kontrollen einzufügen. Insbesondere muss man beim Umgang mit den Kommandos `subs`, `double`, `vpa` sowie mit komplexen Zahlen und Ergebnissen aufmerksam sein.

Serie 1 - Einführung, Theorie und Anwendung

1. Einstiegsbeispiele

a) Veränderungen bei großen Populationen

Unter der Annahme, dass es eine maximale Population m gibt, die z.B. aus Gründen des Lebensraumes nicht überschritten werden kann, sei die Bestandsänderung der Population proportional sowohl zum Bestand als auch zur jeweiligen Abweichung vom möglichen Höchststand.

Dies ergibt die gDGL

$$y' = py(m - y), \quad y = y(t), \quad \text{AB } y(0) = y_0.$$

Sie wird auch als Wachstumsgleichung oder logistische Gleichung bezeichnet.

Man löse das AWP.

Hinweis: Trennung der Variablen, Partialbruchzerlegung

b) Das Fadenpendel

Bekanntlich gilt nach Newton $F = ma$.

Wie läuft die Bewegung eines Pendelkörpers unter den Vereinfachungen: keine Berücksichtigung des Luftwiderstandes, Masse des Fadens wird vernachlässigt, Pendelkörper ist Massepunkt (mathematisches Pendel).

Für kleine Auslenkungen $\alpha(t)$ ergibt sich aus dem Newtonschen Grundgesetz

$$mg \sin(\alpha(t)) = -ml \alpha''(t), \quad l \text{ Fadenlänge,}$$

die gDGL 2. Ordnung

$$\alpha''(t) + \frac{g}{l} \alpha(t) = 0.$$

Man finde die Lösungsmenge der gDGL.

Man diskutiere die folgenden AB.

– Das Pendel wird aus der Lage $\alpha = 0$ mit der Geschwindigkeit v_0 angestoßen.

– Das Pendel wird nach Auslenkung um einen Winkel α_0 losgelassen.

c) Abstände bei unbekanntenen Kurven

Gesucht sind alle Kurven mit folgenden Eigenschaften: Die Tangente an einen Kurvenpunkt P schneide die x -Achse in T . Wie lautet die Gleichung der Kurve, wenn der Abstand zwischen P und T gleich dem Abstand zwischen T und dem Koordinatenursprung O ist für alle Punkte P der Kurve? Aus der Abstandsbedingung $|OT|^2 = |TP|^2$ und der Tangentengleichung $y - y(x_0) = y'(x_0)(x - x_0)$ am Punkt $P(x_0, y_0)$ einer Kurve $y = y(x)$ folgt die gDGL

$$y' = \frac{2xy}{x^2 - y^2}.$$

Man finde die allgemeine Lösung dieser nichtlinearen gDGL unter Verwendung der Substitution $u = \frac{y}{x}$ sowie mittels Partialbruchzerlegung.

d) Zweispezimodell – ökologisches Gleichgewicht bei Populationen

Dies wird beschrieben durch das Volterrasche System von zwei gDGL 1. Ordnung

$$\dot{x} = ax(1 - y), \quad x = x(t),$$

$$\dot{y} = -cy(1 - x), \quad y = y(t),$$

bzw. entkoppelt in der Form

$$\begin{aligned}x\ddot{x} &= \dot{x}^2 + acx^2 - cx\dot{x} + cx^2\dot{x} - acx^3, \\y\ddot{y} &= \dot{y}^2 + acy^2 + ay\dot{y} - ay^2\dot{y} - acy^3.\end{aligned}$$

Man löse die gDGL für $a = 2$, $c = 1$ mit den AB $x(0) = 1$, $y(0) = 3$.

2. Richtungsfelder und Lösungsgebiete

Gegeben sei die explizite gDGL 1. Ordnung $y' = f(x, y)$.

In jedem Punkt $P(x, y)$ eines Gebietes $\Omega \in \mathbb{R}^2$ kann man die Steigung y' durch Eintragen eines kleinen Strichs für die Tangentenrichtung andeuten. Ein solcher Strich heißt Linienelement oder Richtungselement (x, y, y') der Lösungskurve.

Die Gesamtheit aller Linienelemente heißt Richtungsfeld.

Kurven mit konstanter Steigung y' , wo also $f(x, y) = \text{const}$ ist, heißen Isoklinen.

Wie verlaufen die Isoklinen im Fall $f(x, y) = f(x)$ bzw. $f(x, y) = f(y)$?

Eine Lösungskurve ist durch die Vorgabe eines Punktes $P(x_0, y_0)$ (AB) aus dem Definitionsbereich der gDGL eindeutig bestimmt.

Man diskutiere die Richtungsfelder, Isoklinen und Lösungen von gDGL.

a) $y' = 1 + y$. Was passiert auf der Geraden $y = -1$?

b) $y' = \frac{1}{x+y}$. Was passiert auf der Geraden $x + y = 0$?

c) $y' = \frac{2xy}{x^2 - y^2}$. In welche Lösungsgebiete zerfällt die Ebene?

d) $y' = x^2 + y^2$.

e) $y' = y(1 - y)$, logistische Gleichung.

Man bestimme auch alle stationären Lösungen $y = \text{const}$.

3. Orthogonale Trajektorien

Definition: Alle Kurven, die in jedem Punkt eine vorgegebene Kurvenschar orthogonal schneiden, heißen orthogonale Trajektorien.

Für die gDGL $y' = f(x, y)$ kann man die orthogonale Kurvenschar bestimmen, indem man lediglich y' durch $-1/y'$ ersetzt. Die Vektoren $(1, y')$ und $(1, -1/y')$ sind nämlich orthogonal.

Man diskutiere die folgenden Beispiele.

a) $y' = 1 + y$, $y(x) = -1 + ce^x$.
 $-1/y' = 1 + y$, $y(x) = \sqrt{c - 2x}$.

b) $y' = y/x$, $y(x) = cx$.
 $-1/y' = y/x$, $y' = -x/y$, $x^2 + y^2 = c$.

4. Phasenkurve

Für die Lösung $y = y(x)$ der gDGL $y' = f(x, y)$ ist manchmal auch eine andere Darstellung zu empfehlen.

Das trifft zum Beispiel auf Bewegungsdifferentialgleichungen bei Abhängigkeit von der Zeit t zu. Formal kann man $x = t$ setzen und erhält das SysgDGL

$$\dot{x} = 1, \quad \dot{y} = f(t, y).$$

Die Lösung $(x(t), y(t))$ wird Phasenkurve genannt und ihre Darstellung kann im Phasenraum (x, y) in Abhängigkeit von der Zeit t erfolgen.

Mit einer gegebenen Anfangsbedingung $(y(t_0), x(t_0)) = (y_0, x_0)$ löst man das AWP.

Allgemein lautet das SysgDGL

$$\dot{x} = f_1(t, x, y), \quad \dot{y} = f_2(t, x, y).$$

Der eigentliche Typ von Phasenkurven entsteht bei der gDGL 2. Ordnung

$$y'' = f(t, y, y'),$$

wenn man diese durch die Definition $x = y' = \dot{y}$ in das System von zwei gDGL

$$\begin{aligned} \dot{y} &= x = f_1(t, x, y), \\ \dot{x} &= \ddot{y} = f(t, y, x) = f_2(t, x, y), \end{aligned}$$

überführt und somit die Phasenkurve $(y, x) = (y, \dot{y})$ erhält.

Man konstruiere für folgende gDGL die Phasenkurven.

a) $y' = 1 + y, \quad y(0) = \frac{1}{2}.$

b) $y'' + y = 0, \quad y(0) = 0, \quad y'(0) = 1,$ Pendel ohne Reibung.

c) $y'' + ky' + y = 0, \quad y(0) = y_0, \quad y'(0) = y'_0,$ Pendel mit Reibung $k > 0$, z.B. $k = \frac{10}{3}.$

Serie 2 - Theorie, Anwendung und Lösung

1. Aufstellen von gDGL

a) Man bestimme die gDGL für folgende Kurvenschar $y + y^2 = cx^2, \quad c(x-2)^2 + y^2 = c.$

Lösung: $(x-2)y = -y'(x^2 - 4x + 5).$ Hat diese gDGL noch weitere Lösungen?

b) Man bestimme die gDGL für die Kurvenschar $y = cx, \quad x = 0.$

2. Gesucht sind alle Kurven mit folgender Eigenschaft.

Der Schnittpunkt der Kurvennormalen n in einem Kurvenpunkt P mit der x -Achse soll von der Abszisse von P stets den Abstand 2 haben.

a) Man bestimme anhand einer Zeichnung zunächst die zugehörige gDGL.

Mögliche Lösung: $yy' = 2.$

b) Man zeichne das Richtungsfeld und gib die Gebiete an, in denen die gDGL definiert ist.

c) Man bestimme die zugehörige Kurvenschar in der Form $F(y) = G(x) + C$ und gib für die jeweiligen Gebiete $y = F^{-1}(G(x) + C)$ an.

d) Wie lautet die Gleichung derjenigen eindeutig bestimmten Lösung, die durch den Punkt $P(2, -3)$ geht? Man zeichne diese in das Richtungsfeld ein.

e) Man bestimme die orthogonalen Trajektorien der Kurvenschar und zeichne diejenige orthogonale Trajektorie in das Richtungsfeld ein, die durch den Punkt $P(-1, 1)$ verläuft. Wie lautet deren Funktionsgleichung?

3. Orthogonale Trajektorien

Man bestimme zu folgenden Kurvenscharen die gDGL der orthogonalen Trajektorien.

a) Schar der Kreise $(x-c)^2 + y^2 = c^2.$

b) Geradenschar $y = 3x + b.$ Man bestimme die orthogonalen Trajektorien.

c) Parabelschar $y = ax^2$. Welche Kurven sind das?

Man bestimme die orthogonalen Trajektorien zur Kurvenschar

d) $y = \sqrt{x+c}$. Wie lautet die Lösungskurve durch den Punkt $P(-2, 5)$?

4. Lösung von linearen gDGl 1. Ordnung bzw. AWP

a) $y' = \lambda y$, $\lambda \in \mathbb{R}$, $y(0) = 1$

Lösung: $y(x) = e^{\lambda x}$

b) $y' = 2x$, $y(1) = -1$

Lösung: $y(x) = x^2 - 2$

c) $y' = -2x$, $y(1) = -1$

Lösung: $y(x) = -x^2$

d) $y' = 2|x|$, $y(-1) = -1$

Lösung: $y(x) = x|x|$

e) $y' = -200y + 2000 - (1991 + 199x)e^{-x}$, $y(0) = 10$

Lösung: $y(x) = 10(1 + e^{-200x}) - (10 + x)e^{-x}$

f) $y' = 10 \operatorname{sign}(\sin(10x)) \cos(10x)$, $y(0) = 0$

Lösung: $y(x) = |\sin(10x)|$

g) $y' = -\frac{3}{x}y + \frac{2}{x^3}$, $y(1) = 1$

Lösung: $y(x) = -\frac{1}{x^3} + \frac{2}{x^2}$

h) $y' = 1 + \frac{y}{2x}$, $y(1) = 3$

Lösung: $y(x) = 2x + \sqrt{x}$

h) $y' = 1 + \frac{y}{x}$, $y(1) = 1$

Lösung: $y(x) = x(1 + \ln(x))$

i) $y' = x - y$, $y(0) = 1$

Lösung: $y(x) = -1 + x + 2e^{-x}$

j) $y' = -y \frac{c_1 + x}{c_0 + c_1x + c_2x^2}$, $c_i \in \mathbb{R}$, $y(0) = \frac{1}{2}$

Lösung: $y(x) = \frac{1}{2} c_0^{1/(2c_2)} (c_0 + c_1x + c_2x^2)^{-1/(2c_2)} \cdot e^{(2c_2-1)c_1/(c_2\sqrt{4c_0c_2-c_1^2}) (\arctan(c_1/\sqrt{4c_0c_2-c_1^2}) - \arctan((c_1+2c_2x)/\sqrt{4c_0c_2-c_1^2}))}$

5. Lösung von linearen gDGl 2. Ordnung

a) $y'' = \lambda^2 y$, $\lambda \in \mathbb{R}^+$, $y(0) = 2$, $y'(0) = 0$

Lösung: $y(x) = e^{\lambda x} + e^{-\lambda x}$

b) $y'' = xy$, $y(0) = c_0$, $y'(0) = c_1$, Airys Gleichung

Lösung: $y(x) = \frac{1}{6\Gamma(2/3)} (-2 \cdot 3^{5/6} c_1 \pi - 3 \cdot 3^{2/3} c_0 \Gamma(2/3)^2) \operatorname{AiryAi}(x) + 3^{1/3} (2c_1 \pi + 3^{5/6} c_0 \Gamma(2/3)^2) \operatorname{AiryBi}(x)$.

c) $y'' = 2x + y$, $y(0) = 0$, $y'(0) = 0$

Lösung: $y(x) = e^x - e^{-x} - 2x$

- d) $y'' = 3y' - 5y + 1$, $y(0) = 0$, $y'(0) = 1$
 Lösung: $y(x) = \frac{13\sqrt{11}}{55}e^{3x/2} \sin(\sqrt{11}x/2) - \frac{1}{5}e^{3x/2} \cos(\sqrt{11}x/2) + \frac{1}{5}$
- e) $y'' = -12y' - 20y$, $y(0) = 4$, $y'(0) = 0$
 Lösung: $y(x) = 5e^{-2x} - e^{-10x}$
- f) $y'' = -12y' - 20y$, $y(0) = \frac{1}{2}$, $y'(0) = -1$
 Lösung: $y(x) = \frac{1}{2}e^{-2x}$
- g) $y'' = -\omega^2 y + \varepsilon f(x, y, y')$, $\omega \in \mathbb{R}^+$, $\varepsilon \in \mathbb{R}$, $y(0) = c_0$, $y'(0) = c_1$
 Lösung für $\varepsilon = 0$: $y(x) = a \sin(\omega x + \phi)$

6. Lösung von linearen gDGL höherer Ordnung mit physikalischen Anwendungen (Schwingungs-DGL) und mit Verbindungen zur Differential- und Integralrechnung

- a) Beschreibung der Resonanz zwischen Eigenschwingung $y(x)$ und Störung $3e^{-x}$.
 $y'' + 6y' + 5y = 3e^{-x}$
- b) Welche Lösung der gDGL genügt der AB $y(0) = y'(0) = 0$?
 Wie verhält sich die Funktion $y(x)$ für sehr große x ?
 $y'' + 6y' + 25y = 3 \sin(4x)$
- c) Welche Lösungskurve hat im Punkt $P(0, 0.5)$ einen Extremwert?
 Von welcher Art ist dieses Extremum?
 $y'' - 4y' + 4y = 0$
- d) Welche Lösungskurve hat im Punkt $P(0, 1)$ den Neigungswinkel 150° ?
 $y'' + y' + \frac{1}{4}y = 0$
- e) Man bestimme den Grenzwert $\lim_{x \rightarrow \infty} y(x)$.
 $y''' + 12y'' + 6y' + y = 2e^{-x}$

7. Es sei gegeben die lineare gDGL $y'' + p_1(x)y' + p_2(x)y = q(x)$.

- a) Man zeige.
- Sind y_1 und y_2 partikuläre Lösungen dieser gDGL, so ist $y_3 = y_1 - y_2$ eine partikuläre Lösung der zugehörigen homogenen gDGL.
 - Ist y_1 eine partikuläre Lösung der gDGL, so ist auch $y_1 + y_h$ eine Lösung für jedes y_h , welches Lösung der zugehörigen homogenen gDGL ist.
- b) Die obige gDGL habe nunmehr speziell die Form $y'' + 5y' + 6y = 6x^2 + 10x$.
 Man zeige.
- Als eine partikuläre Lösung der inhomogenen linearen gDGL lässt sich ein Polynom 2. Grades angeben. Welches?
 - Die zugehörige homogene gDGL hat Lösungen der Form $y = e^{kx}$ mit geeigneten Werten für k . Man bestimme diese. Wie viele linear unabhängige Lösungen lassen sich auf diese Weise finden?
 - Man gebe die insgesamt gefundene Lösungsmenge an.

8. Man diskutiere die Lösungen $y(t)$ der parametrischen gDGL (Parameteraufgaben).

- a) Schwingungsgleichung ohne Einwirkung äußerer Kräfte
 $y'' + 2hy' + k^2y = 0$
- b) gDGL der erzwungenen Schwingung $y(t)$ mit Einwirkung äußerer Kräfte
 Man untersuche $y(t)$ für $t \rightarrow \infty$.

$$y'' + 2\rho y' + \omega^2 y = a \sin(\omega_1 t), \quad y(0) = 0, \quad y'(0) = \omega$$

- c) Man bestimme die Lösung, die im Punkt $P(0, 1)$ eine Wendetangente mit der Steigung -2 und für $t \rightarrow \infty$ die Asymptote $y = -1$ hat.

Man zeige, dass diese Lösung Nullstellen und Extremstellen besitzt (gegebenenfalls numerische Berechnung).

$$y'''' + 3y''' + 3y'' + y' = 10e^{-2t}$$

9. Betrachtung von nichtlinearen gDGL

- a) Man diskutiere die Lösung der gDGL, die ein relatives Extremum an einer Stelle hat. Von welcher Art ist dieses Extremum?

$$x^2 y' - y^2 - xy = 0$$

- b) Man diskutiere die Lösungsschar der gDGL.

$$x^2 y' + (1 - x)y = 0$$

- c) Riccatische DGL $y' = x - y^2$, $y(0) = 0$, $0 \leq x \leq 1$.

10. Gegeben ist die nichtlineare gDGL $y' = y^2 x^3 + xy$.

- a) Man bestimme den Typ der gDGL und transformiere sie in eine lösbare Form.

- b) Man ermittle die allgemeine Lösung.

$$\text{Hinweis: } z' + xz = -x^3.$$

- c) Man zeige, dass alle Lösungen symmetrisch zur y -Achse sind.

- d) Man bestimme diejenige Trajektorie, die durch den Punkt $P(0, 1)$ geht.

- e) Man zeige, dass die partikuläre Lösung $y = 1/(2 - x^2 - e^{-x^2/2})$ mindestens zwei Polstellen hat und genau ein relatives Extremum im Punkt Null besitzt. Von welcher Art ist das Extremum?

11. Lösung von nichtlinearen gDGL 1. Ordnung

- a) $y' = -y^2 + \frac{y}{x}$, $y(1) = 2$

$$\text{Lösung: } y(x) = \frac{2}{x}$$

- b) $y' = y - \frac{2x}{y}$, $y(0) = 1$

$$\text{Lösung: } y(x) = \sqrt{2x + 1}$$

- c) $y' = (2y \ln(x) - 1)\frac{y}{x}$, $y(1) = \frac{1}{2}$

$$\text{Lösung: } y(x) = \frac{1}{2(1 + \ln(x))}$$

- d) $y' = \frac{1}{1 + \tan^2(y)}$, $y(0) = 0$

$$\text{Lösung: } y(x) = \arctan(x)$$

- e) $y' = -200xy^2$, $y(-1) = \frac{1}{101}$

$$\text{Lösung: } y(x) = \frac{1}{1 + 100x^2}$$

- f) $y' = \frac{1}{y + 10^{-p}}$, $p \in \mathbb{R}$, $y(0) = 0$

$$\text{Lösung: } y(x) = -10^{-p} + \sqrt{10^{-2p} + 2x}$$

g) Spezielle Riccatische DGI $y' = 1 - y^2$, $y(0) = 0$, $0 \leq x \leq 1$.

Man transformiere die gDGI auf eine Gleichung 2. Ordnung und löse diese.

Hinweis: Man setze $y = \frac{u'}{u}$. Die exakte Lösung ist $y(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

12. Lösung von nichtlinearen gDGI 2. Ordnung

Die Angabe einer exakten Lösung ist i. Allg. nur in wenigen Fällen möglich.

a) $y'' = -y + \frac{y^3}{6} + 2 \sin(2.78535x)$, $y(0) = 0$, $y'(0) = 0$

b) $y'' = -y + \varepsilon y^3$, $\varepsilon \in \mathbb{R}$, $y(0) = 1$, $y'(0) = 0$

c) $y'' = -\frac{2}{x}y' - y^p$, $p \in \mathbb{R}^+$, $y(0) = 1$, $y'(0) = 0$

Lösung für $p = 5$: $y(x) = \frac{1}{\sqrt{1 + x^2/3}}$

d) $y'' + y^3 = 0$, $y(0) = 1$, $y'(0) = 0$

e) $\ddot{y} = 6y^2$, $y(0) = 1$, $\dot{y}(0) = 0$, $t \in [0, 1]$

$\ddot{y} = 2y^3 + ty$, $y(0) = 1$, $\dot{y}(0) = 0$, $t \in [0, 1]$

f) Van der Polsche DGI

$\ddot{y} - \varepsilon(1 - y^2)\dot{y} + y = 0$, $y(0) = y_0$, $\dot{y}(0) = \dot{y}_0$, $t \geq 0$, $\varepsilon \geq 0$

Der Parameter ε legt die Einschwingdauer fest.

Lösung für $\varepsilon = 0$: $y(t) = \dot{y}_0 \sin(t) + y_0 \cos(t)$

AB: $y(0) = 0$, $\dot{y}(0) = -0.05$, $y(0) = 2$, $\dot{y}(0) = 0$

g) Rotation eines Pendels mit der Winkelgeschwindigkeit ω

$\ddot{y} + \sin(y) - \frac{1}{2}\omega^2 \sin(2y) = 0$, $y(0) = \frac{\pi}{4}$, $\dot{y}(0) = 0$, $\omega \in [0, 10]$

h) Bewegungsgleichung

$$\ddot{y} + f(y) = 0, \quad y(0) = 2, \quad \dot{y}(0) = 0, \quad f(z) = \begin{cases} -1, & \text{falls } z \leq -1, \\ z, & \text{falls } -1 < z < 1, \\ 1, & \text{falls } 1 \leq z \end{cases}$$

i) Duffingsche Bewegungsgleichungen

$\ddot{y} + 0.7\dot{y} + y^3 = 0.75 \cos(t)$, $y(0) = \dot{y}(0) = 0$

$\ddot{y} + 0.2\dot{y} + y|y| = 1.5 \cos(2t) + 0.5$, $y(0) = \dot{y}(0) = 0$

$\ddot{y} + \dot{y} - \frac{1}{6}y^3 = 0.8 \sin(0.27\omega t)$, $y(0) = \dot{y}(0) = 0$, $\omega = 0.92845$

j) Bewegungsgleichung mit quadratischer Dämpfung

$\ddot{y} + \frac{1}{2}\dot{y}^2 + y = 0$, $y(0) = 1$, $\dot{y}(0) = 0$

k) Emdens Bewegungsgleichungen

$\frac{d}{dt}(t^2 \dot{y}) + t^2 y^n = 0$, $y(0) = 1$, $\dot{y}(0) = 0$, $n = 2, 3, 5, \dots$

$\frac{1}{t^2} \frac{d}{dt}(t^2 \dot{y}) + f(y) = \ddot{y} + \frac{2}{t} \dot{y} + f(y) = 0$, $y(0) = 1$, $\dot{y}(0) = 0$, $f(y) = \begin{cases} \sin(y), \\ \cos(y) \end{cases}$

k) Liouvillesche DGI

$$\frac{d}{dt}(t^2 \dot{y}) + t^2 e^y = 0, \quad y(0) = \dot{y}(0) = 0$$

l) Mathieus DGI für ein Pendel mit zeitabhängiger Länge

$$\ddot{y} + (a - 2q \cos(2t)) y = 0, \quad y(0) = 1, \quad \dot{y}(0) = 0$$

$a = 1.0, q = 0.1,$ aufschaukelnde Schwingung,
 $a = 1.2, q = 0.1,$ gedämpfte Schwingung

Serie 3 - Einschrittverfahren

1. Gegeben sei die gDGI 1. Ordnung bzw. das AWP

$$y' = f(x, y), \quad \text{AB } y(0), \quad x \geq 0, \quad f \in C^2(\mathbb{R}^2).$$

Für seine Lösung verwenden wir das explizite Einschrittverfahren (ESV) von Euler-Cauchy (Polygonzugverfahren, PZV)

$$\eta_{n+1} = \eta_n + hf(x_n, \eta_n), \quad n = 0, 1, \dots, \quad x_n = nh, \quad \eta_0 = y(0).$$

a) Man löse die gDGI $y' = y$ mittels PZV für die beiden AB $y(0) = 0, y(0) = 1$.

– Welches sind die Näherungswerte für $y(1)$, wenn die Schrittweite h die Werte 1, 0.1, 0.01, 0.001 annimmt? Man diskutierte den Fehler.

– Das Lösungsintervall sei $[0, 1]$ und $h = \frac{1}{N}$.
Welcher Term beschreibt η_N ?

Welche Grenzwerte ergeben sich für beide AB bei $N \rightarrow \infty$?

b) Welche Lösung liefert das PZV für die gDGI $y' = f(x), y(a) = 0, x \in [a, b]$?

– Man untersuche den Sonderfall $y' = cx, y(0) = 0, c = \text{const}$.

c) Gegeben sei das AWP $y' = c\sqrt{y}, y(0) = 0, c = \text{const} > 0$.

– Welche Funktion liefert das PZV als Grenzwert, wenn die Schrittgröße $h \rightarrow 0$ strebt? Erfüllt diese Funktion die gDGI?

– Angenommen, die gDGI beschreibt den freien Fall eines Körpers.

Wie verhält sich die gefundene Lösung im Vergleich zur Wirklichkeit?

– Man untersuche die Glattheit der rechten Seite $f(x, y) = c\sqrt{y}$.

– Gegeben sei $\tau \geq 0$ und die Funktion

$$z(x) = \begin{cases} 0, & \text{falls } 0 \leq x \leq \tau, \\ \frac{1}{4}c^2(x - \tau)^2, & \text{falls } x > \tau. \end{cases}$$

Erfüllt $z(x)$ die gDGI? Welche Auswirkung hat die Antwort auf das Problem der Eindeutigkeit der Lösung? Worin liegt die Ursache für diese Situation?

d) Die Bewegungsgleichung eines Körpers mit der Masse m in einer zähen Flüssigkeit (laminare Strömung) mit dem Stokesschen Widerstandsgesetz $R(v) = K_1 v$ (K_1 Materialeigenschaften des Mediums/Geometrie des Körpers) ergibt sich aus dem Kraftgesetz

$$m\dot{v} = A + G + R(v), \quad G \text{ Erdanziehung, } A \text{ Auftrieb.}$$

Die abgeleitete Modellgleichung für die Geschwindigkeit $v(t)$ sei $\dot{v} = c + k_1 v, c > 0$.

– Man experimentiere mit dem PZV bei verschiedenen Startwerten und Schrittweiten.

– Es sei $v_0 = 0$, $k_1 < 0$, $c > 0$.

Gibt es eine Gleichgewichtsgeschwindigkeit v_∞ , wo sich zwischen Erdanziehung und Widerstand ein Gleichgewicht einstellt?

- e) Die Bewegungsgleichung eines Körpers mit der Masse m in einer turbulenten Strömung (Neigung zur Wirbelbildung) mit dem Newtonschen Widerstandsgesetz $R(v) = K_2 v^2$ (K_2 Materialeigenschaften/Geometrie) ergibt sich aus dem Kraftgesetz

$$m\dot{v} = A + G + R(v).$$

Die abgeleitete Modellgleichung sei $\dot{v} = c + k_2 v^2$, $c > 0$.

– Es sei $v(0) = v_0$, $k_2 < 0$, $c > 0$.

Man experimentiere mit dem PZV bei verschiedenen Startwerten und Schrittweiten (auch relativ große).

Man suche experimentell die optimale Schrittweite.

Ist chaotisches Verhalten zu erkennen?

Gibt es eine Gleichgewichtsgeschwindigkeit v_∞ ?

- f) Die Bewegungsgleichung für den freien Fall (ohne Reibung) mit der von der Höhe $z(t)$ abhängigen Erdbeschleunigung $g(z)$ lautet $\ddot{z} = -g(z)$.

Nach dem Newtonschen Gravitationsgesetz gilt $g(z) = \frac{g_0 z_0^2}{z^2}$, wobei $g_0 = g(z_0)$ den Wert der Erdbeschleunigung auf der Höhe z_0 bezeichnet.

Für kleine Fallhöhen kann man drei Approximationen (3 Modellgleichungen) der rechten Seite der gDGL anwenden.

– $g_0 = g = 9.806 \text{ m s}^{-2}$ Normwert, $r_E = 6.375 \cdot 10^6 \text{ m}$ Erdradius, $z_0 = z$,

$$\ddot{z} = -g,$$

– $g(z) \approx g - 2 \frac{g}{r_E} (z - r_E) = 3g - 2 \frac{g}{r_E} z$,

$$\ddot{z} = -3g + 2 \frac{g}{r_E} z,$$

– $z_0 = r_E$, $g_0 = g$,

$$\ddot{z} = -\frac{g r_E^2}{z^2}.$$

Die AB sind jeweils $z(0) = r_E + H$, $\dot{z}(0) = 0$, H Fallhöhe.

– Welche konstante Funktion ist Lösung der zweiten Modellgleichung?

Man finde ihre allgemeine Lösung.

– Man berechne die Fallzeit t_F und Endgeschwindigkeit $\dot{z}(t_F)$ bei einer Fallhöhe H und gegebenen AB für die Modellgleichungen 1 und 2.

– Man bestimme mittels PZV für $H = 10^3, 10^4, 10^5 \text{ m}$ die Fallzeiten für alle drei Modelle und vergleiche diese.

Werden die Unterschiede durch die Linearisierung des Modells oder durch den Diskretisierungsfehler des PZV verursacht?

2. Man löse das AWP mit dem PZV

$$\eta_{n+1} = \eta_n + hf(x_n, \eta_n), \quad n = 0, 1, \dots, N-1, \quad x_n = nh, \quad \eta_0 = y(0),$$

$$h = \frac{x_N - x_0}{N}, \quad [x_0, x_N] = [a, b].$$

Man berechne die Fehler $|y(x_N) - \eta_N|$, $N = 5, 10, 20$.

Woran erkennt man die Konvergenzordnung 1 des PZV?

Man vergleiche den Wert $|y(x_N) - \eta_N|$ mit der Formel für die Abschätzung des globalen Diskretisierungsfehlers $|e_N|$ gemäß

$$|e_n| \leq (e^{K(x_n - x_0)} - 1) \frac{M_2}{2K} h, \quad n = 1, 2, \dots, N, \quad e_0 = 0,$$

$$K = \max_{a \leq x \leq b} \left| \frac{\partial f}{\partial y}(x, y(x)) \right|,$$

$$M_2 = \max_{a \leq x \leq b} |y''(x)|.$$

Man bewerte die Güte der Abschätzung $|e_n|$. Welche geometrische Deutung haben die Größen K und M_2 . Welchen Einfluss hat das auf die Wahl von h ?

a) $y' = x - y^2$, $y(0) = 0$, $0 \leq x \leq 1$, Riccati-DGL, $y(x) = \frac{\sqrt{3} \text{AiryAi}(1,x) + \text{AiryBi}(1,x)}{\sqrt{3} \text{AiryAi}(x) + \text{AiryBi}(x)}$,

$$y(1) = 0.455\,544\,526\,081\,874.$$

b) $y' = -y$, $y(0) = 1$, $0 \leq x \leq 5$, $y(x) = e^{-x}$, $y(5) = e^{-5}$.

c) $y' = x - y$, $y(0) = 1$, $0 \leq x \leq 1$, $y(x) = 2e^{-x} + x - 1$, $y(1) = 2e^{-1}$.

3. Man löse das AWP mit dem PZV gemäß Aufgabe 2.

Die rechte Seite der gDGL genüge einer Lipschitz-Bedingung der Form

f beschränkt auf $\Omega \subset \mathbb{R}^2$,

$$|f(x, y) - f(\bar{x}, \bar{y})| \leq K|y - \bar{y}|, \quad (x, y), (\bar{x}, \bar{y}) \in \Omega, \quad x \approx \bar{x}.$$

Sei $\Delta\eta_n = \eta_{n+1} - \eta_n$. Man zeige, dass $|\Delta\eta_n| \leq (1 + hK)^n |\Delta\eta_0|$.

Man werte die Abschätzung aus, indem man für die gDGL $y' = y$, $y(0) = 1$, und $y' = -y$, $y(0) = 1$, sie jeweils mit der exakten rekursiven Formel $\Delta\eta_n = C(h)\Delta\eta_0$ vergleicht.

Serie 4 - Explizite Einschrittverfahren

1. Die gDGL $y' = y(1 - y)$ ist für $0 \leq x < \infty$ zu lösen.

Die exakte Lösung mit der AB $y(0) = y_0$ ist $y(x) = \frac{y_0 e^x}{1 + y_0(e^x - 1)}$.

a) Man zeichne das Richtungsfeld und bestimme alle stationären Lösungen

$$y(x) = \text{const.}$$

b) Welche stationäre Lösung $y^*(x) = \text{const}$, $0 \leq x < \infty$, ist asymptotisch stabil? Begründung.

c) Man wende das PZV

$$\eta_{n+1} = \eta_n + h\eta_n(1 - \eta_n), \quad n = 0, 1, 2, \dots,$$

mit $\eta_0 \geq 0$ und den Schrittweiten $h = 0.25, 0.5, 1.0, 2.0, 2.5$ an.

Für welche Schrittweiten $h \leq h_0$ konvergiert das PZV gegen die exakte Lösung $y^*(x) = \text{const}$?

d) Welche Lösungen liefert das PZV für $h > h_0$?

e) Man teste auf analoge Weise für diese gDGL das klassische Runge-Kutta-Verfahren 4. Ordnung (KRKV).

2. Man gebe eine geometrische Interpretation der ESV an (Schema mit Zuwächsen hk_i).

a) Euler-Cauchy-Verfahren (PZV)

$$\begin{aligned}k_1 &= f(x_n, \eta_n), \\ \eta_{n+1} &= \eta_n + hk_1.\end{aligned}$$

b) Collatz-Verfahren
(modifiziertes PZV=MPZV)

$$\begin{aligned}k_1 &= f(x_n, \eta_n), \\ k_2 &= f\left(x_n + \frac{h}{2}, \eta_n + \frac{h}{2}k_1\right), \\ \eta_{n+1} &= \eta_n + hk_2.\end{aligned}$$

c) Heun-Verfahren
(HV, verbessertes PZV=VPZV)

$$\begin{aligned}k_1 &= f(x_n, \eta_n), \\ k_2 &= f(x_n + h, \eta_n + hk_1), \\ \eta_{n+1} &= \eta_n + \frac{h}{2}(k_1 + k_2).\end{aligned}$$

d) KRKV

$$\begin{aligned}k_1 &= f(x_n, \eta_n), \\ k_2 &= f\left(x_n + \frac{h}{2}, \eta_n + \frac{h}{2}k_1\right), \\ k_3 &= f\left(x_n + \frac{h}{2}, \eta_n + \frac{h}{2}k_2\right), \\ k_4 &= f(x_n + h, \eta_n + hk_3), \\ \eta_{n+1} &= \eta_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).\end{aligned}$$

3. Welche Integrationsformeln vom Newton-Côtes-Typ werden durch folgende ESV verallgemeinert?

- Collatz-Verfahren, modifiziertes PZV (1960).
- Heun-Verfahren, verbessertes PZV (1900).
- (3,3)-Verfahren von Kutta (1901).
- KRKV (1895).

4. Man löse das AWP $y' = x - y$, $y(0) = 1$, $x \in [0, 1]$.

- Man verwende die ESV von Euler-Cauchy, Collatz und Heun mit den Schrittweiten $h = 0.2, 0.1$.
- Man vergleiche die Lösungen miteinander sowie mit der exakten Lösung $y(x) = 2e^{-x} + x - 1$.
- Man berechne explizit für alle drei Verfahren $\eta_{n+1} = F(x_n, \eta_n; h, f)$ die Verfahrensfunktion. Was stellt man beim Vergleich von Collatz- und Heun-Verfahren fest?

5. Man löse das AWP $y' = x - y$, $y(0) = 1$, $x \in [0, 1]$, $y(x) = 2e^{-x} + x - 1$, mit dem KRKV bei einer Schrittweite $h = 0.2$.

- Man gebe explizit das Verfahren in der Form $\eta_{n+1} = F(x_n, \eta_n; h, f)$ an.
- Um eine vernünftige (variable) Schrittweite für das KRKV anzuwenden, ist man durch heuristische Betrachtungen zur Schrittkenzahl κ gelangt.

Diese ist definiert durch

$$\kappa = 2 \left| \frac{k_3 - k_2}{k_2 - k_1} \right|$$

und sollte sich zwischen 0.1 und 0.2 bewegen.

Man weise nach, dass dies durch entsprechende Wahl der Schrittweite h erreichbar ist. Man zeige dazu, dass gilt

$$\kappa \approx h \left| \frac{\partial f}{\partial y} \right|.$$

c) Man vervollständige die Tabelle der Schrittkenzzahlen und vergleiche.

x	$2 \left \frac{k_3 - k_2}{k_2 - k_1} \right $	$h \left \frac{\partial f}{\partial y} \right $
0	0.2000	0.2
0.2		
0.4		
0.6		
0.8		
1.0		

d) Können die gDGL $y' = y$ und $y' = -y$ mit der gleichen Schrittweite gerechnet werden?

e) Man untersuche analog das AWP $y' = xy$, $y(0) = 1$, $0 \leq x \leq 2$.

6. Gegeben sei die gDGL

$$y' = \frac{1}{y + 10^{-p}}, \quad p = 1, \quad y(0) = 0, \quad y(x) = -10^{-p} + \sqrt{10^{-2p} + 2x}.$$

a) Man finde mittels Trennung der Variablen die exakte Lösung $y(x) \geq 0$.

b) Man bestimme $f_y = \frac{\partial f}{\partial y}$ sowie mittels der Schrittkenzzahl $\kappa = h |f_y| = 0.1$ eine Formel für die Schrittweite h . Wie groß ist die Anfangsschrittweite h_0 ?

Hinweis: Darstellung von h als $h(x) = \alpha x + \beta$.

c) Man ergänze die folgende Tabelle der Integrationsstützstellen und Schrittweiten.

n	$x_{n+1} = x_n + h_n$	$h_n = \alpha x_n + \beta$
0	$x_0 = 0$	$h_0 = \beta = \dots$
1		
2		
3		
4		
5		

d) Man zeige durch vollständige Induktion die Beziehungen

$$x_n = \beta \sum_{i=0}^{n-1} (\alpha + 1)^i = \beta \frac{(\alpha + 1)^n - 1}{\alpha}, \quad h_n = \beta (\alpha + 1)^n.$$

e) Wieviel RK-Schritte sind (theoretisch) notwendig, um das AWP im Intervall $[0,5]$ bei $\kappa = 0.1$ zu lösen?

Man bestimme die Endschrittweite und die letzten Stützstellen.

Hinweis: $x_N = \beta \frac{(\alpha + 1)^N - 1}{\alpha} = 5$.

f) Was passiert mit f_y , h_n und N , wenn der Parameter p sehr groß ist?

7. Für die gDGL $y' = f(x, y)$ betrachten wir auf Basis des PZV $\bar{\eta}_{n+1} = \eta_n + hf(x_n, \eta_n)$ die Verbesserung als HV in verschiedenen Formen.

$$\begin{aligned} \eta_{n+1} &= \eta_n + \frac{h}{2}(f(x_n, \eta_n) + f(x_{n+1}, \bar{\eta}_{n+1})) = \eta_n + \frac{h}{2}[f(x_n, \eta_n) + f(x_{n+1}, \eta_n + hf(x_n, \eta_n))] \\ &= \bar{\eta}_{n+1} + \frac{h}{2}(f(x_{n+1}, \bar{\eta}_{n+1}) - f(x_n, \eta_n)) = \bar{\eta}_{n+1} + \Delta, \quad \Delta \text{ Korrektur} \\ &= \eta_n + hk, \quad k = \frac{1}{2}(k_1 + k_2), \quad k_1 = f(x_n, \eta_n), \quad k_2 = f(x_{n+1}, \bar{\eta}_{n+1}), \\ &\quad k_i \text{ Anstiege, } hk_i \text{ Zuwächse} \\ &= \eta_n + h\Phi(x_n, \eta_n; h, f), \quad k = \Phi \text{ Inkrementfunktion} \\ &= F(x_n, \eta_n; h, f), \quad F \text{ Verfahrensfunktion} \\ &= E(x_n; h, f)\eta_n, \quad E \text{ Verstärkungs-, Transformations-, Stabilitätsfunktion.} \end{aligned}$$

Man zeige. Falls $f(x, y)$ stetig differenzierbar ist, so genügt die Verfahrensfunktion $F(x, u; h, f)$ des HV einer Lipschitz-Bedingung.

8. Man notiere das KRKV komponentenweise für die gDGL 3. Ordnung

$$y''' = f(x, y, y', y''), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0, \quad y''(x_0) = y''_0.$$

Hinweis: Transformation auf ein SysgDGL 1. Ordnung.

Beispiel: $y''' = -\frac{1}{2}y''^3$, $y(0) = 1$, $y'(0) = 3$, $y''(0) = 1$, $y(x) = \frac{4}{3}(x+1)^{3/2} + x - \frac{1}{3}$,
allgemeine Lösung $y(x) = \pm \frac{4}{3}(x+C_1)^{3/2} + C_2x + C_3$ mittels Substitution
 $z = y''$ und Transformation der Variablen.

9. Man notiere die Formeln des KRKV für das SysgDGL 1. Ordnung

$$y' = f(x, y, z), \quad y(x_0) = y_0, \quad z' = g(x, y, z), \quad z(x_0) = z_0.$$

Man nutze und vereinfache diese bei der Lösung von

a) $y'' = f(x, y, y')$, $y(x_0) = y_0$, $y'(x_0) = y'_0$,

b) $y'' = f(x, y)$, $y(x_0) = y_0$, $y'(x_0) = y'_0$.

Man löse die AWP mittels KRKV.

c) $y'' - 3y' + 5y - 1 = 0$, $y(0) = 0$, $y'(0) = 1$, $x \in [0, 1]$, $h = 0.2$,

das Fundamentalsystem der homogenen Gleichung ist

$$\{e^{3x/2} \cos(\sqrt{11}x/2), e^{3x/2} \sin(\sqrt{11}x/2)\},$$

die Lösung ist $y(x) = \frac{1}{5} + e^{\frac{3}{2}x} \left(-\frac{1}{5} \cos\left(\frac{\sqrt{11}}{2}x\right) + \frac{13}{55} \sqrt{11} \sin\left(\frac{\sqrt{11}}{2}x\right) \right)$.

d) $y'' = 2x + y$, $y(0) = y'(0) = 0$, $x \in [0, 1]$, $h = 0.2$,

die exakte Lösung ist $y(x) = e^x - e^{-x} - 2x$.

10. Für die Lösung $y(1)$ des AWP (Riccati-DGI) $y' = x - y^2$, $y(0) = 0$,

$y(x) = \frac{\sqrt{3} \operatorname{AiryAi}(1,x) + \operatorname{AiryBi}(1,x)}{\sqrt{3} \operatorname{AiryAi}(x) + \operatorname{AiryBi}(x)}$, liegen die Näherungen η_N , $N = 5, 10, 20, 40$, des KRKV vor.

N	h	$\eta_N(x_N = 1)$	$e_N(h)$	$\frac{e_N(h)}{e_{2N}(h/2)}$
5	0.2	0.455 533 498 11	-1.10e-5	16.1
10	0.1	0.455 543 842 70	-6.84e-7	16.1
20	0.05	0.455 544 483 62	-4.24e-8	16.0
40	0.025	0.455 544 523 43	-2.65e-9	16.0

Exakter Wert (15 Stellen): $y(1) = 0.455\,544\,526\,081\,874$

- Man ermittle Schätzungen des globalen Diskretisierungsfehlers e_N nach dem Runge-Prinzip und vergleiche diese mit dem wahren Fehler e_N , $N = 10, 20, 40$.
- Man verbessere die Näherungswerte der jeweiligen Feinrechnung durch Richardson-Extrapolation (Ergebnis: η_N^*).
- Man ermittle experimentell die Konvergenzordnung der verbesserten Lösung η_N^* .

11. Man ermittle experimentell die Konvergenzordnung des 3-stufigen RKV mit dem Parameterschema

$\frac{1}{4}$	$\frac{1}{4}$		
$\frac{27}{40}$	$-\frac{189}{800}$	$\frac{729}{800}$	
1	$\frac{214}{891}$	$\frac{1}{33}$	$\frac{650}{891}$

12. Gegeben sei das lineare homogene SysgDGI 1. Ordnung mit konstanten Koeffizienten

$$y' = A y, \quad A(m, m)\text{-Matrix.}$$

- Man gebe die allgemeine Lösung in Matrixschreibweise an.
- Man wende das Euler-Cauchy-Verfahren an und notiere es in der Matrixform $\eta_{n+1} = M \eta_n$, $n = 0, 1, 2, \dots$, mit der Transformations- oder Verfahrensmatrix $M = M(x_n, \eta_n; h, A)$.
- Man notiere das HV und das KRKV ebenfalls in Matrixform und bestimme die jeweilige Transformationsmatrix M .

13. Gegeben ist das SysgDGI 2. Ordnung bzw. AWP.

Man bestimme $y_1(1)$ und $y_2(1)$ näherungsweise mittels eines ESV bis auf eine Genauigkeit von 10^{-4} .

a) Lineares SysgDGI

$$y_1' = x(y_2 - y_1), \quad y_1(0) = 1,$$

$$y_2' = x(y_2 + y_1), \quad y_2(0) = 1,$$

$$\text{Lösung: } y_1(x) = \cosh(x^2/\sqrt{2}),$$

$$y_2(x) = \cosh(x^2/\sqrt{2}) + \sqrt{2} \sinh(x^2/\sqrt{2}).$$

b) Nichtlineares SysgDGI

$$\begin{aligned} y_1' &= x - y_2, & y_1(0) &= 1, \\ y_2' &= -x + y_1 y_2, & y_2(0) &= 1. \end{aligned}$$

c) Autonomes nichtlineares SysgDGI, Riccati-DGI

$$\begin{aligned} y_1' &= y_2 - y_1^2, & y_1(0) &= 0, \\ y_2' &= 1, & y_2(0) &= 0. \end{aligned}$$

d) Steifes lineares SysgDGI

$$\begin{aligned} y_1' &= -29998y_1 - 39996y_2, & y_1(0) &= 1, \\ y_2' &= 14998.5y_1 + 19997y_2, & y_2(0) &= 1, \\ \text{Lösung : } y_1(x) &= 7e^{-10000x} - 6e^{-x}, \\ y_2(x) &= -3.5e^{-10000x} + 4.5e^{-x}, & \text{asymptotisch stabil.} \end{aligned}$$

14. Man schreibe ein Programm für das PZV, das ein AWP mit einer vorgegebenen Schrittweite h löst. Man löse das AWP.

$$y' = y(1 - y), \quad y(0) = y_0, \quad 0 \leq x \leq 10, \quad y(x) = \frac{y_0 e^x}{1 + y_0(e^x - 1)},$$

mit $h = 0.25, 0.5, 1.0, 2.0, 2.5$ und $y_0 = -1, 0, 0.5, 1, 2$.

15. Man schreibe ein Programm mit ausgewählten (4,4)-RKV, die AWP für gDGI 1. Ordnung bei einer Startschrittweite h mit einer gegebenen Genauigkeit $\varepsilon > 0$ lösen.

Man notiere dazu folgende Programmteile als Prozeduren.

- RK-Schritt von x bis $x + h$ bei gegebenem h ,
- SWS durch Grob- und Feinrechnung, Algorithmussskizze:

$$\begin{aligned} k_h &\text{ - Zuwachs bei Grobrechnung in } [x_n, x_n + h], \\ k'_{h/2}, k''_{h/2} &\text{ - Zuwächse bei Feinrechnung in } [x_n, x_n + \frac{h}{2}] \text{ bzw. } [x_n + \frac{h}{2}, x_n + h], \\ err &= \frac{k'_{h/2} + k''_{h/2} - k_h}{15}, \quad \text{Test auf } |err| \in [\frac{\varepsilon}{50}, \varepsilon], \end{aligned}$$

falls $|err| > \varepsilon$, dann Schritt ab x_n mit $\frac{h}{2}$ wiederholen,

falls $|err| \leq \varepsilon$, dann Schritt akzeptieren, aber bei $|err| < \frac{\varepsilon}{50}$ für nächsten Schritt Verdopplung der Schrittweite,

bei Akzeptanz $\eta_{n+1} = \eta_n + k'_{h/2} + k''_{h/2} + err$.

Man motiviere die Idee der Festlegung des Toleranzintervalls $[\frac{\varepsilon}{50}, \varepsilon]$.

Man teste es für selbst gewählte Beispiele.

Folgende RKV mit gegebenem Parameterschema sind zu implementieren.

a) KRKV

$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	

b) 3/8-Regel

$\frac{1}{3}$	$\frac{1}{3}$			
$\frac{2}{3}$	$-\frac{1}{3}$	1		
1	1	-1	1	
$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$	

c) (4,4)-England-Formel

$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$		
1	0	-1	2	
$\frac{1}{6}$	0	$\frac{2}{3}$	$\frac{1}{6}$	

d) Kuntzmann-Formel

$$\begin{array}{c|ccc} \frac{2}{5} & \frac{2}{5} & & \\ \frac{3}{5} & -\frac{3}{20} & \frac{3}{4} & \\ 1 & \frac{19}{44} & -\frac{15}{44} & \frac{40}{44} \\ \hline & \frac{55}{360} & \frac{125}{360} & \frac{125}{360} \quad \frac{55}{360} \end{array}$$

e) Gill-Modifikation der RK-Formel, $c = 1/\sqrt{2}$

$$\begin{array}{c|ccc} \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & c - \frac{1}{2} & 1 - c & \\ 1 & 0 & -c & 1 + c \\ \hline & \frac{1}{6} & \frac{1-c}{3} & \frac{1+c}{3} \quad \frac{1}{6} \end{array}$$

f) RKV mit minimalen Diskretisierungsfehler

$$\alpha_2 = \frac{2}{5}, \quad \alpha_3 = \frac{14 - 3\sqrt{5}}{16}, \quad \beta = \frac{1 - \alpha_2}{(\alpha_3 - \alpha_2)(6\alpha_2\alpha_3 - 4(\alpha_2 + \alpha_3) + 3)}$$

$$\begin{array}{c|ccc} \alpha_2 = \frac{2}{5} & \frac{2}{5} & & \\ \alpha_3 = \frac{14-3\sqrt{5}}{16} & \alpha_3 - \beta_{32} & \beta_{32} & \\ 1 & 1 - \beta_{42} - \beta_{43} & \beta_{42} & \beta_{43} \\ \hline & 1 - \gamma_2 - \gamma_3 - \gamma_4 & \gamma_2 & \gamma_3 \quad \gamma_4 \end{array}$$

wobei

$$\beta_{32} = \frac{\alpha_3(\alpha_3 - \alpha_2)}{2\alpha_2(1 - 2\alpha_2)}, \quad \beta_{42} = \frac{\beta(-1 + \alpha_2 + \alpha_3 - (2\alpha_3 - 1)^2)}{2\alpha_2}, \quad \beta_{43} = \frac{\beta(1 - 2\alpha_2)(1 - \alpha_3)}{\alpha_3},$$

$$\gamma_2 = \frac{2\alpha_3 - 1}{12\alpha_2(\alpha_3 - \alpha_2)(1 - \alpha_2)}, \quad \gamma_3 = \frac{1 - 2\alpha_2}{12\alpha_3(\alpha_3 - \alpha_2)(1 - \alpha_3)}, \quad \gamma_4 = \frac{1}{2} + \frac{2(\alpha_2 + \alpha_3) - 3}{12(1 - \alpha_2)(1 - \alpha_3)}.$$

16. Man schreibe ein Computerprogramm für das KRKV, das AWP für SysgDGI

1. Ordnung mit einer gegebenen Genauigkeit löst.

Man notiere dazu folgende Programmteile als Prozeduren.

- RK-Schritt von x bis $x + h$ bei gegebenem h ,
- SWS durch Grob- und Feinrechnung.

17. Man löse mittels PZV bzw. KRKV das autonome SysgDGI 1. Ordnung für $t \in [0, T]$

$$\begin{aligned} u' &= -u + 25v, & u(0) &= 0, \\ v' &= -2v, & v(0) &= c, \quad c > 0, \end{aligned}$$

$$\begin{aligned} \text{Lösung: } u(t) &= 25c(e^{-t} - e^{-2t}), \\ v(t) &= ce^{-2t}, \quad \text{asymptotisch stabil.} \end{aligned}$$

Man teste $c = 0.1, 0.02, 0.01$ und $N = 200(200)1000$, $h = \frac{T}{N}$, $T = 10, 20$.

Das AWP sei nun erweitert durch eine moderate Nichtlinearität zu

$$\begin{aligned} u' &= -u + 25v - v\sqrt{u^2 + v^2}, & u(0) &= 0, \\ v' &= -2v + u\sqrt{u^2 + v^2}, & v(0) &= c, \quad c > 0. \end{aligned}$$

Man teste die RKV mit den Größen

- $c = 0.1, 0.02;$ $N = 500, 700, 1000;$ $T = 6, 10,$
- $c = 0.01, 0.0105;$ $N = 500, 700, 1000;$ $T = 15, 30,$
- $c = 0.0106;$ $N = 700, 1000, 2000;$ $T = 10, 20, 50, 54, 55, 65,$
- $c = 0.0105556757;$ $N = 700, 1000, 2000;$ $T = 40, 48, 50, 55, 56, 59, 60, 61, 68, 69, 70, 94.$

Man vergleiche die Ergebnisse mit Berechnungen durch ein CAS, z.B. mit Maple, MATLAB oder *Mathematica*.

Serie 5 - Implizite Einschrittverfahren

1. Man zeige unter entsprechenden Voraussetzungen, dass das implizite (1,1)-Euler-Verfahren (Gear(1)-Verfahren)

$$\eta_{n+1} = \eta_n + hf(x_{n+1}, \eta_{n+1}), \quad n = 0, 1, \dots, \eta_0 \text{ gegeben, d.h.}$$

$$\eta_{n+1} = \eta_n + hk_1, \quad k_1 = f(x_n + h, \eta_n + hk_1),$$

- a) konsistent mit Ordnung 1 ist,
b) konvergent mit Ordnung 1 ist.

2. Die sogenannten θ -Verfahren sind parametrische ESV der Gestalt

$$\eta_{n+1} = \eta_n + h[(1 - \theta)f(x_n, \eta_n) + \theta f(x_{n+1}, \eta_{n+1})], \text{ d.h.}$$

$$\eta_{n+1} = \eta_n + h[(1 - \theta)f(x_n, \eta_n) + \theta k_1],$$

$$k_1 = f(x_{n+1}, \eta_{n+1}) = f(x_n + h, \eta_n + h[(1 - \theta)f(x_n, \eta_n) + \theta k_1]),$$

mit dem reellen Parameter $\theta \geq 0$. Sei $f \in C^2(\mathbb{R}^2)$.

Welche Konsistenzordnung besitzen diese Verfahren? Man beachte die Sonderfälle $\theta = 0$, $\theta = \frac{1}{2}$ (Schnentrapezregel, HV) und $\theta = 1$.

Oft wird auch folgende Parameterkonstellation verwendet.

$$\eta_{n+1} = \eta_n + h\left[\frac{1}{2}f(x_n, \eta_n) + \frac{1}{2}f(x_{n+1}, \tilde{\eta}_{n+1})\right], \text{ d.h.}$$

$$\eta_{n+1} = \eta_n + h\left[\frac{1}{2}f(x_n, \eta_n) + \frac{1}{2}k_1\right], \quad k_1 = f(x_{n+1}, \tilde{\eta}_{n+1}) = f(x_n + h, \eta_n + hk_1).$$

Es besitzt die Konsistenzordnung 2.

3. Zu den θ -Verfahren gehören auch die sogenannten One-leg-Verfahren

$$\eta_{n+1} = \eta_n + hf((1 - \theta)x_n + \theta x_{n+1}, (1 - \theta)\eta_n + \theta\eta_{n+1}), \quad 0 \leq \theta \leq 1.$$

Für $\theta = \frac{1}{2}$ erhält man die Tangententrapezregel (MPZV).

Man untersuche die Konsistenzordnung.

4. Man zeige unter entsprechenden Voraussetzungen die Konsistenz und Konvergenz mit Ordnung 2 des impliziten (1,2)-Gauß-Verfahrens (Mittelpunktsregel)

$$\eta_{n+1} = \eta_n + hf\left(x_n + \frac{h}{2}, \frac{1}{2}\eta_n + \frac{1}{2}\eta_{n+1}\right), \text{ d.h.}$$

$$\eta_{n+1} = \eta_n + hk_1, \quad k_1 = f\left(x_n + \frac{h}{2}, \eta_n + \frac{h}{2}k_1\right).$$

5. Das linear-implizite Euler-Verfahren für autonome SysgDGL $y' = f(y)$ lautet

$$\eta_{n+1} = \eta_n + h \left[I - h \frac{\partial f(\eta_n)}{\partial y} \right]^{-1} f(\eta_n), \quad n = 0, 1, 2, \dots,$$

mit der Einheitsmatrix I .

- a) Unter welchen Voraussetzungen ist das Verfahren konsistent?
b) Man bestimme die Stabilitätsfunktion $F(z) = F(\lambda, h)$, $z = \lambda h$.
Ist das Verfahren A-stabil?
c) Man zeige die Konvergenz des Verfahrens.

6. Gegeben sei das AWP $y' = 2\sqrt{y}$, $y(0) = y_0 = 0$.

Die exakte Lösung ist nicht eindeutig. Neben der trivialen Lösung $y(x) = 0$ gibt es eine zweite Lösung

$$z(x) = \begin{cases} 0, & \text{falls } 0 \leq x \leq \tau, \\ (x - \tau)^2, & \text{falls } x > \tau. \end{cases}$$

a) Welche Näherung liefert das PZV als Grenzwert, wenn die Schrittgröße $h \rightarrow 0$ strebt? Erfüllt diese Funktion die gDGI?

b) Man notiere für die gDGI das implizite (1,1)-Euler-Verfahren

$$\eta_{n+1} = \eta_n + hf(x_{n+1}, \eta_{n+1}), \quad n = 0, 1, \dots, \quad \eta_0 = 0.$$

Kann man damit die beiden Lösungen $y(x) = 0$ und $z(x)$ approximieren?

Serie 6 - Mehrschrittverfahren

1. Die allgemeine einheitliche Form von r -stufigen expliziten und impliziten Mehrschrittverfahren (MSV) ist

$$\sum_{j=0}^r \alpha_{r-j} \eta_{i+1-j} - h \sum_{j=0}^r \beta_{r-j} f_{i+1-j} = 0, \quad \alpha_r = 1, \quad i \geq r-1, \quad f_k = f(x_k, \eta_k), \quad (9.1)$$

$\eta_0, \eta_1, \dots, \eta_{r-1}$ gegeben.

Eine zweite etwas anschaulichere, aber spezielle Darstellung ist

$$\eta_{i+p} = \eta_{i-q} + h \sum_{j=0}^r b_j f_{i-j}, \quad p \geq 0, \quad q \geq 0, \quad i \geq s = \max(r, q), \quad \eta_0, \eta_1, \dots, \eta_{s+p-1} \text{ geg.} \quad (9.2)$$

Eine dritte besondere Form ist

$$\alpha_r \eta_{i+1} + \alpha_{r-1} \eta_i + \dots + \alpha_0 \eta_{i+1-r} = h \beta_r f_{i+1}, \quad \alpha_r, \beta_r \neq 0, \quad i \geq r-1, \quad \eta_0, \dots, \eta_{r-1} \text{ geg.} \quad (9.3)$$

Daraus kann man die grundlegenden Klassen von MSV bei entsprechender Wahl von p, q, r ableiten.

Formel (9.2)

$p \geq 1$: explizite MSV,

$p = 0, b_0 \neq 0$: implizite MSV,

$p = 1, q = 0$: expl. $(r+1)$ -stufige Adams-Bashforth-Formeln der Ordn. $r+1$,

$p = 0, q = 1$: implizite s -stufige Adams-Moulton-Formeln der Ordn. $r+1$,
 $s = \max(r, 1)$,

$p = 1, q = 1$: explizite s -stufige Nyström-Formeln, $s = \max(r+1, 2)$,

$p = 0, q = 2$: implizite s -stufige Milne-Simpson-Formeln, $s = \max(r, 2)$,
aus der Klasse vom Adams-Störmer-Typ,

Formel (9.3) : Backward Difference Formulas (BDF), steif-stabile
Methoden von Gear.

Bei der Herleitung der Formeln (9.1) – (9.3) wird gemäß ihrem interpolativen Charakter (Newton-Côtes-Formeln der Integration, Newton-Interpolationspolynom, Schema der dividierten Differenzen, Vorwärts- bzw. Rückwärtsdifferenzen) oftmals die Darstellung mittels Rückwärtsdifferenzen

$$\nabla^j f_i = \nabla^{j-1} f_i - \nabla^{j-1} f_{i-1}, \quad \nabla^0 f_i = f_i,$$

genommen.

Jede Summe von Funktionswerten in (9.1) – (9.3) kann in eindeutiger Weise mittels Rückwärtsdifferenzen dargestellt werden.

Koeffizienten der expliziten Adams-Bashforth(n)-Verfahren n -ter Ordnung

$$\eta_{i+1} = \eta_i + h \sum_{j=0}^r b_j f_{i-j}, \quad i = r, r+1, \dots, \quad \eta_0, \eta_1, \dots, \eta_r \text{ gegeben}$$

r	Ordnung n Stufen	Fehler- konstante $ch^n f^{(n+1)}$	K	j						
				0	1	2	3	4	5	
0	1	$\frac{1}{2}$	1	1						
1	2	$\frac{5}{24}$	2	3	-1					
2	3	$\frac{3}{8}$	12	23	-16	5				
3	4	$\frac{251}{720}$	24	55	-59	37	-9			
4	5	$\frac{95}{288}$	720	1901	-2774	2616	-1274	251		
5	6	$\frac{19087}{60480}$	1440	4277	-7923	9982	-7298	2877	-475	

Koeffizienten der impliziten Adams-Moulton(n)-Verfahren n -ter Ordnung

$$\eta_i = \eta_{i-1} + h \sum_{j=0}^r b_j f_{i-j}, \quad i = r, r+1, \dots, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben}$$

r	Stufen s	Ordnung n	Fehler- konstante $ch^n f^{(n+1)}$	K	j					
					0	1	2	3	4	5
0	1	1	$-\frac{1}{2}$	1	1					
1	1	2	$-\frac{1}{12}$	2	1	1				
2	2	3	$-\frac{1}{24}$	12	5	8	-1			
3	3	4	$-\frac{19}{720}$	24	9	19	-5	1		
4	4	5	$-\frac{3}{160}$	720	251	646	-264	106	-19	
5	5	6	$-\frac{863}{60480}$	1440	475	1427	-798	482	-173	27

Koeffizienten der impliziten Milne-Simpson-Verfahren n -ter Ordnung

$$\eta_i = \eta_{i-2} + h \sum_{j=0}^r b_j f_{i-j}, \quad i = r, r+1, \dots, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben}$$

r	Stufen	Ordnung	K	j						
	s	n		0	1	2	3	4	5	6
				$K \cdot b_j$						
0	2	1	1	2						
1	2	2	1	0	2					
2	2	4	3	1	4	1				
4	4	5	90	29	124	24	4	-1		
5	5	6	90	28	129	14	14	-6	1	
6	6	7	3780	1139	5640	33	1328	-807	264	-37

- Bemerke: $r = 0$ Adams-Moulton(1) mit Schrittweite $2h$,
 $r = 1$ Nyström-Formel, explizite Mittelpunktsregel,
 $r = 2$ klassisches Adams-Störmer-Verfahren oder Milne-Verfahren,
 Simpson-Regel.

Einige explizite Nyström-Verfahren n -ter Ordnung.

$$\eta_{i+1} = \eta_{i-1} + h \sum_{j=0}^r b_j f_{i-j}, \quad i = r, r+1, \dots, \quad \eta_0, \eta_1, \dots, \eta_r \text{ gegeben}$$

r	Ordnung	K	j			
	n		0	1	2	3
	Stufen		$K \cdot b_j$			
0	2	1	2			
2	3	3	7	-2	1	
3	4	3	8	-5	4	-1

Koeffizienten der impliziten BDF(n)-Verfahren n -ter Ordnung

$$\alpha_r \eta_{i+1} + \alpha_{r-1} \eta_i + \dots + \alpha_0 \eta_{i+1-r} = h \beta_r f_{i+1}, \quad i = r-1, r, \dots, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben}$$

r	Ordnung	β_r	j						
	n		r	$r-1$	$r-2$	$r-3$	$r-4$	$r-5$	$r-6$
	Stufen		α_j						
1	1	1	1	-1					
2	2	2	3	-4	1				
3	3	6	11	-18	9	-2			
4	4	12	25	-48	36	-16	3		
5	5	60	137	-300	300	-200	75	-12	
6	6	60	147	-360	450	-400	225	-72	10

a) Darstellung von MSV mit Rückwärtsdifferenzen

– Adams-Bashforth-Formeln

$$\eta_{i+1} = \eta_i + h \sum_{j=0}^r b_j f_{i-j} = \eta_i + h \sum_{j=0}^r c_j \nabla^j f_i$$

Die neuen Koeffizienten berechnen sich aus $c_j = (-1)^j \int_0^1 \binom{-s}{j} ds$ und sind

j	0	1	2	3	4	5
c_j	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$

Die jeweils nächste Formel ergibt sich einfach durch Hinzunahme eines neuen Gliedes $\nabla^j f_i$. Man überprüfe die Richtigkeit der angegebenen Koeffizienten c_j für $r = 0, 1, 2$.

– Adams-Moulton-Formeln

$$\eta_i = \eta_{i-1} + h \sum_{j=0}^r b_j f_{i-j} = \eta_{i-1} + h \sum_{j=0}^r c_j \nabla^j f_i$$

Die neuen Koeffizienten berechnen sich aus $c_j = (-1)^j \int_0^1 \binom{-s+1}{j} ds$ und sind

j	0	1	2	3	4	5
c_j	1	$-\frac{1}{2}$	$-\frac{5}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{27}{1440}$

Man überprüfe die Richtigkeit der angegebenen Koeffizienten c_j für $r = 0, 1, 2$.

– Milne-Simpson-Formeln

$$\eta_i = \eta_{i-2} + h \left(2f_{i-1} + \frac{1}{3} \nabla^2 f_i - \frac{1}{90} \nabla^4 f_i - \frac{1}{90} \nabla^5 f_i - \frac{37}{3780} \nabla^6 f_i \right)$$

Man überprüfe die Richtigkeit bis zur Formel 4. Ordnung.

– BDF, Gear-Formeln

$$\frac{1}{r} \nabla^r \eta_{i+1} + \frac{1}{r-1} \nabla^{r-1} \eta_{i+1} + \dots + \nabla^1 \eta_{i+1} = h f_{i+1}$$

Man überprüfe die Richtigkeit für $r = 1, 2, 3$.

2. Man leite die Konsistenzordnung des klassischen 3-stufigen Adams-Bashforth-Verfahrens her.

$$\eta_{m+1} = \eta_m + \frac{h}{12} (23f_m - 16f_{m-1} + 5f_{m-2}), \quad n = 2, 3, \dots, \quad \eta_0, \eta_1, \eta_2 \text{ gegeben.}$$

3. Man löse das AWP (Riccati-DGL)

$$y' = x - y^2, \quad y(0) = -0.5, \quad x \in [0, 1],$$

mittels MSV und vergleiche die Ergebnisse.

a) Klassisches Adams-Bashforth(3)-Verfahren bei einer Schrittweite $h = 0.1$.

Hinweis: Man nutze für die Anlaufrechnung das KRKV.

b) Prädiktor-Korrektor-Verfahren von Milne 6. Ordnung mit der Schrittweite $h = 0.1$

$$\bar{\eta}_{n+1} = \eta_{n-5} + \frac{h}{10}(33f_n - 42f_{n-1} + 78f_{n-2} - 42f_{n-3} + 33f_{n-4}),$$

$$\eta_{n+1} = \eta_{n-3} + \frac{h}{45}(14\bar{f}_{n+1} + 64f_n + 24f_{n-1} + 64f_{n-2} + 14f_{n-3}),$$

$$\bar{f}_{n+1} = f(x_{n+1}, \bar{\eta}_{n+1}), \quad n = 5, 6, \dots, \eta_0, \eta_1, \dots, \eta_5 \text{ gegeben}$$

4. Man berechne die Lösung des AWP $y' = x - y$, $y(0) = 1$, $x \in [0, 1]$,
 $y(x) = 2e^{-x} + x - 1$, mittels Adams-Bashforth(n)-Verfahren mit $n = 1, 2, 3, 4$ bei
 der Schrittweite $h = 0.1$.

Die benötigten Startwerte sind durch eine Anlaufrechnung vorgegeben.

x	0	0.1	0.2	0.3	0.4	0.5
y	1	0.910	0.837	0.782	0.741	0.713

5. Man bestimme für MSV das erste und zweite erzeugende Polynom

$$\rho(\xi) = \sum_{i=0}^r \alpha_i \xi^i, \quad \alpha_r = 1, \quad \sigma(\xi) = \sum_{i=0}^r \beta_i \xi^i$$

und das charakteristische Polynom $\phi(\xi) = \rho(\xi) - z\sigma(\xi)$, $z = h\lambda$.
 Deshalb wird das Verfahren auch (ρ, σ) -MSV genannt.

a) 4-stufiges Adams-Bashforth(4)-Verfahren

$$\eta_{n+1} = \eta_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}), \quad n = 3, 4, \dots$$

b) 3-stufiges Adams-Moulton(4)-Verfahren

$$\eta_{n+1} = \eta_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}), \quad n = 2, 3, \dots$$

6. Gegeben sind zwei lineare MSV.

4-stufiges explizites Milne-Verfahren 4. Ordnung

$$\eta_{n+1} = \eta_{n-3} + \frac{h}{3}(8f_n - 4f_{n-1} + 8f_{n-2}), \quad n = 3, 4, \dots,$$

2-stufiges implizites Adams-Störmer-Verfahren 4. Ordnung (Milne-Simpson(4),
 Simpson-Regel)

$$\eta_{n+1} = \eta_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}), \quad n = 1, 2, \dots$$

a) Man bestimme die erzeugenden Polynome $\rho(\xi)$ bzw. $\sigma(\xi)$.

b) Man bestimme die Konsistenzordnungen.

c) Man bestimme das charakteristische Polynom $\phi(\xi)$ und die Konsistenzordnung
 des aus den Formeln bestehenden PC-Verfahrens von Milne.

d) Man überprüfe das folgende Wurzelkriterium:

Das erzeugende Polynom $\rho(\xi)$ hat keine Nullstellen ξ mit $|\xi| > 1$
 und alle Nullstellen mit $|\xi| = 1$ sind einfach.

Sind die Verfahren konvergent?

7. Das 2-stufige implizite Adams-Störmer-Verfahren 4. Ordnung (Milne-Simpson(4)) für $y' = f(x, y)$

$$\eta_{n+1} = \eta_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}), \quad n = 2, 3, \dots,$$

ist ein Differenzenschemaverfahren, das auf der Simpson-Regel der numerischen Integration beruht. Es gilt

$$\begin{aligned} \eta_{n+1} - \eta_{n-1} &= \int_{x_{n-1}}^{x_{n+1}} y' dx = \int_{x_{n-1}}^{x_{n+1}} f(x, y(x)) dx \\ &= \frac{2h}{6}(f(x_{n-1}, \eta_{n-1}) + 4f(x_n, \eta_n) + f(x_{n+1}, \eta_{n+1})) \\ &= \frac{h}{3}(6f_n + f_{n+1} - 2f_n + f_{n-1}) \\ &= h(2f_n + \frac{1}{3}\nabla^2 f_{n+1}), \end{aligned}$$

$$\eta_{n+1} = \eta_{n-1} + h(2f_n + \frac{1}{3}\nabla^2 f_{n+1}),$$

mit dem rückwärtigen Differenzenoperator

$$\nabla^k f_n = \nabla^{k-1} f_n - \nabla^{k-1} f_{n-1}, \quad \nabla^0 f_n = f_n.$$

Unter der Annahme, dass $\nabla^3 f_{n+1} = \nabla^3 f_n = \text{const}$ ist, folgt aus

$$\nabla^3 f_{n+1} = \nabla^2 f_{n+1} - \nabla^2 f_n$$

die Beziehung

$$\nabla^2 f_{n+1} = \nabla^2 f_n + \nabla^3 f_n.$$

Somit erhält man das explizite, aber 4-stufige Schema der Gestalt

$$\eta_{n+1} = \eta_{n-1} + h(2f_n + \frac{1}{3}(\nabla^2 f_n + \nabla^3 f_n)) = \eta_{n-1} + \frac{h}{3}(8f_n - 5f_{n-1} + 4f_{n-2} - f_{n-3})$$

(explizites Nyström(4)), für das in einer Anlaufrechnung mittels RKV die Anfangswerte $\eta_0, \eta_1, \eta_2, \eta_3$ bereitgestellt werden müssen.

Pro Schritt ist aber nur ein Funktionswert $f(x, y)$ zu berechnen.

Die Berechnungen kann man günstig als Tableau notieren.

k	x_k	η_k	f_k	∇f_k	$\nabla^2 f_k$	$\nabla^3 f_k$
$n-3$			
$n-2$		
$n-1$	
n
$n+1$

Man löse das AWP $y' = x - y$, $y(0) = 1$, $x \in [0, 1]$, $h = 0.1$, exakte Lösung ist $y(x) = 2e^{-x} + x - 1$, $y(1) = \frac{2}{e} = 0.735758882\dots$, mit folgendem Ausgangstableau und vervollständige es.

n	x_n	η_n	f_n	∇f_n	$\nabla^2 f_n$	$\nabla^3 f_n$
0	0	1	-1			
1	0.1	0.9096	-0.8096	0.1904		
2	0.2	0.8374	-0.6374	0.1722	-0.0182	
3	0.3	0.7816	-0.4816	0.1558	-0.0164	0.0018
4	0.4					
5	0.5					
6	0.6					
7	0.7					
8	0.8					
9	0.9					
10	1.0					

Man überprüfe die Annahme $\nabla^3 f_n = \text{const.}$

8. Das Adams-Störmer-Verfahren 4. Ordnung mit der Modifikation aus Aufgabe 7 ist die Basis für folgendes Iterationsverfahren.

$$\eta_{n+1}^{(0)} = \eta_{n-1} + h \left(2f_n + \frac{1}{3}(\nabla^2 f_n + \nabla^3 f_n) \right) \quad (\text{Nyström(4)}),$$

$$\eta_{n+1}^{(k+1)} = \eta_{n-1} + h \left(2f_n + \frac{1}{3}\nabla^2 f_{n+1}^{(k)} \right), \quad k = 0, 1, \dots, \quad (\text{Milne-Simpson(4)}).$$

Man kontrolliere, wie viele Iterationen für jedes n auszuführen sind, bis sich in der angegebenen Stellenzahl die Iterierten nicht mehr ändern und ergänze die folgende Tabelle.

n	x_n	η_n	f_n	∇f_n	$\nabla^2 f_n$	$\nabla^3 f_n$
0	0	1	-1			
1	0.1	0.909675	-0.809675	0.190325		
2	0.2	0.837462	-0.637462	0.172213	-0.018112	
3	0.3	0.781637	-0.481637	0.155825	-0.016388	0.001724
4	0.4	$\eta_4^{(0)}$ $\eta_4^{(1)}$ $\eta_4^{(2)}$ η_4	0.740646			
5	0.5	$\eta_5^{(0)}$ $\eta_5^{(1)}$ $\eta_5^{(2)}$ η_5				

9. Man löse das AWP (lineare homogene gDGI 2. Ordnung)

$$y'' + 12y' + 20y = 0, \quad y(0) = \frac{1}{2}, \quad y'(0) = -1,$$

$$\text{Lösung } y(x) = \frac{1}{2}e^{-2x},$$

mit dem modifizierten Verfahren von Adams-Störmer (Nyström(4))

$$\eta_{n+1} = \eta_{n-1} + h \left(2f_n + \frac{1}{3}(\nabla^2 f_n + \nabla^3 f_n) \right) = \eta_{n-1} + \frac{h}{3} (8f_n - 5f_{n-1} + 4f_{n-2} - f_{n-3}).$$

- a) Man bestimme die exakte Lösung des AWP mittels des Ansatzes $y(x) = c e^{\lambda x}$.
 b) Man überführe die gDGL in das SysgDGL 1. Ordnung

$$\begin{aligned} y' &= f(x, y, z), & y(0) &= y_0, \\ z' &= g(x, y, z), & z(0) &= z_0. \end{aligned}$$

- c) Man finde die Darstellung mit den Inkrementfunktionen Φ_i .

$$\begin{aligned} \eta_{n+1} &= \eta_{n-1} + h\Phi_1(\eta'_n, \eta'_{n-1}, \eta'_{n-2}, \eta'_{n-3}), \\ \eta'_{n+1} &= \eta'_{n-1} + h\Phi_2(\eta_n, \eta_{n-1}, \eta_{n-2}, \eta_{n-3}, \eta'_n, \eta'_{n-1}, \eta'_{n-2}, \eta'_{n-3}). \end{aligned}$$

- d) Folgende Anfangswerte sind gegeben.

x	0	0.1	0.2	0.3
y	0.5	0.4094	0.3352	0.2744
y'	-1	-0.8187	-0.6703	-0.5488

Man ergänze die Tabelle.

n	x_n	η_n	$\eta'_n = f_n$	∇f_n	$\nabla^2 f_n$	$\nabla^3 f_n$	g_n	∇g_n	$\nabla^2 g_n$	$\nabla^3 g_n$
0	0	0.5	-1				2			
1	0.1	0.4094	-0.8187	0.1813			1.6375	-0.3625		
2	0.2	0.3352	-0.6703	0.1484	-0.0329		1.3406	-0.2969	0.0656	
3	0.3	0.2744	-0.5488	0.1215	-0.0269	0.0060	1.0976	-0.2430	0.0539	-0.0117
4	0.4									
5	0.5									

10. Prädiktor-Korrektor-Verfahren (PC, PCV, PECE) bestehen i. Allg. aus einer expliziten Mehrschrittformel (Prädiktor) und einer nachfolgenden impliziten Mehrschrittformel (Korrektor).

In jedem Schritt wird der Prädiktor einmal berechnet und der Korrektor ist eine Iterationsformel, die mit dem Prädiktorwert startet und nur wenige Male ausgeführt wird (oft genügt hier eine Iteration).

Modifizierte PC-Verfahren sind solche, wo jeweils noch nach der Formel ein sogenannter Modifikator "nachgeschaltet" ist. Beim ersten Schritt entfällt der erste Modifikator, weil es dort die Größe $\eta_n^{(0)}$ noch nicht gibt.

Im iterativen Korrektor wird auf der rechten Seite zunächst $f(x_{n+1}, \bar{\eta}_{n+1}^{(0)})$ benutzt und in den evtl. weiteren Iterationen dann $f(x_{n+1}, \eta_{n+1}^{(k)})$, $k > 0$.

- a) Man diskutiere die nachfolgenden modifizierten PC-Verfahren 4. Ordnung.

Aus welchen Prädiktor- und Korrektor-Formeln sind sie zusammengesetzt?

- Milne, $err = -\frac{1}{90}h^4 f^{(5)}$

$$\eta_{n+1}^{(0)} = \eta_{n-3} + \frac{4}{3}h(2f_n - f_{n-1} + 2f_{n-2}), \quad \text{Milne-Prädiktor,}$$

$$\bar{\eta}_{n+1}^{(0)} = \eta_{n+1}^{(0)} + \frac{28}{29}(\eta_n - \eta_n^{(0)}), \quad \text{1. Modifikator (Evaluation),}$$

$$\eta_{n+1} \leftarrow \eta_{n+1}^{(k+1)} = \eta_{n-1} + \frac{1}{3}h(f(x_{n+1}, \eta_{n+1}^{(k)}) + 4f_n + f_{n-1}), \quad k = 0, 1, \dots, K, \\ \text{Korrektor,}$$

$$\bar{\eta}_{n+1}^{(1)} = \eta_{n+1}^{(K+1)} + \frac{1}{29}(\eta_{n+1}^{(0)} - \eta_{n+1}^{(K+1)}), \quad \text{2. Modifikator (Evaluation).}$$

– Hamming, $err = -\frac{1}{40}h^4 f^{(5)}$

$$\begin{aligned}\eta_{n+1}^{(0)} &= \eta_{n-3} + \frac{4}{3}h(2f_n - f_{n-1} + 2f_{n-2}), \\ \bar{\eta}_{n+1}^{(0)} &= \eta_{n+1}^{(0)} + \frac{112}{121}(\eta_n - \eta_n^{(0)}), \\ \eta_{n+1} \leftarrow \eta_{n+1}^{(k+1)} &= \frac{1}{8}(9\eta_n - \eta_{n-2}) + \frac{3}{8}h(f(x_{n+1}, \eta_{n+1}^{(k)}) + 2f_n - f_{n-1}), \quad k = 0(1)K, \\ \bar{\eta}_{n+1}^{(1)} &= \eta_{n+1}^{(K+1)} + \frac{9}{121}(\eta_{n+1}^{(0)} - \eta_{n+1}^{(K+1)}).\end{aligned}$$

– Adams-Bashforth-Moulton

$$\begin{aligned}\eta_{n+1}^{(0)} &= \eta_n + \frac{1}{24}h(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}), \\ \bar{\eta}_{n+1}^{(0)} &= \eta_{n+1}^{(0)} + \frac{251}{270}(\eta_n - \eta_n^{(0)}), \\ \eta_{n+1} \leftarrow \eta_{n+1}^{(k+1)} &= \eta_n + \frac{1}{24}h(9f(x_{n+1}, \eta_{n+1}^{(k)}) + 19f_n - 5f_{n-1} + f_{n-2}), \quad k = 0(1)K, \\ \bar{\eta}_{n+1}^{(1)} &= \eta_{n+1}^{(K+1)} + \frac{19}{270}(\eta_{n+1}^{(0)} - \eta_{n+1}^{(K+1)}).\end{aligned}$$

– Hermite, $err = -\frac{31}{720}h^4 f^{(5)}$

$$\begin{aligned}\eta_{n+1}^{(0)} &= \eta_n + \frac{1}{2}h(-f_n + 3f_{n-1}) + \frac{1}{12}h^2(17\eta_n'' + 7\eta_{n-1}''), \\ \bar{\eta}_{n+1}^{(0)} &= \eta_{n+1}^{(0)} + \frac{31}{30}(\eta_n - \eta_n^{(0)}), \\ \eta_{n+1} \leftarrow \eta_{n+1}^{(k+1)} &= \eta_n + \frac{1}{2}h(f(x_{n+1}, \eta_{n+1}^{(k)}) + f_n) + \frac{1}{12}h^2(-(\eta_{n+1}^{(k)})'' + \eta_n''), \quad k = 0(1)K, \\ \bar{\eta}_{n+1}^{(1)} &= \eta_{n+1}^{(K+1)} - \frac{1}{30}(\eta_{n+1}^{(0)} - \eta_{n+1}^{(K+1)}).\end{aligned}$$

Dieses Verfahren braucht noch die zweite Ableitung

$$y'' = \frac{df(x, y)}{dx} = f_x + f_y y' = f_x + f_y f,$$

die für die gegebene gDGL möglichst einfach zu berechnen sein sollte.
Für das nachfolgende AWP ist

$$f_y = -\frac{2 \tan(y)}{1 + \tan^2(y)} \begin{cases} < 0, & \text{falls } 0 \leq y < \pi/2, \\ = 0, & \text{falls } y = \pi/2, \\ > 0, & \text{falls } \pi/2 < y < \pi, \end{cases} \quad \text{Periode ist } \pi,$$

$$|f_y| \leq K = 1, \quad y'' = -\frac{2 \tan(y)}{(1 + \tan^2(y))^2}.$$

b) Man löse mit den PC-Verfahren mit der Schrittweite $h = 0.1$ das AWP

$$y' = \frac{1}{1 + \tan^2(y)}, \quad y(0) = 0, \quad x \in [0, 20], \quad \text{Lösung } y(x) = \arctan(x).$$

Man führe nur eine Korrekteriteration aus.

Die Anfangswerte, berechnet mittels RKV(4,4) mit minimalen Diskretisierungsfehler, sind

x	0	0.1	0.2	0.3	0.4
y	0	0.099 668 674	0.197 395 591	0.291 456 812	0.380 506 356

11. Man bestimme für das BDF(2)-Verfahren

$$\eta_{n+1} = \frac{4}{3}\eta_n - \frac{1}{3}\eta_{n-1} + \frac{2}{3}hf_{n+1}$$

die erzeugenden Polynome $\rho(\xi)$, $\sigma(\xi)$ und die Konsistenzordnung.

Ist das Verfahren konvergent?

Hinweis: Wurzelkriterium.

12. Die iterative Lösung des impliziten Adams-Störmer-Verfahrens 4. Ordnung (Milne-Simpson(4)) erfolgt gemäß

$$\eta_{n+1}^{(k)} = \eta_{n-1} + \frac{h}{3}(f_{n+1}^{(k-1)} + 4f_n + f_{n-1}), \quad k = 0, 1, \dots, \quad f_{n+1}^{(k-1)} = f(x_{n+1}, \eta_{n+1}^{(k-1)}),$$

$\eta_{n+1}^{(0)}$ gegeben.

Man untersuche, unter welcher Bedingung an h das Verfahren konvergiert.

Man vergleiche das Ergebnis mit der Forderung aus der Schrittkenzahl des KRKV $\kappa = h|f_y| = 0.1 \dots 0.2$.

Hinweis: Man schätze die Differenz $|\eta_{n+1}^{(k+1)} - \eta_{n+1}^{(k)}|$ unter Anwendung der Lipschitz-Bedingung ab und nutze die Kontraktionseigenschaft.

13. Eine vierte Möglichkeit der Notation von impliziten MSV ist die Form

$$\eta_{n+1} = \sum_{j=1}^r a_j \eta_{n+1-j} + h \sum_{j=0}^r b_j f(x_{n+1-j}, \eta_{n+1-j}), \quad n \geq r-1, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben,}$$

bzw.

$$\eta_{n+1} = hb_0 f(x_{n+1}, \eta_{n+1}) + \sum_{j=1}^r (a_j \eta_{n+1-j} + hb_j f_{n+1-j}).$$

Die Bestimmung von η_{n+1} erfolgt iterativ bei gegebenem $\eta_{n+1}^{(0)}$

$$\eta_{n+1}^{(k+1)} = hb_0 f(x_{n+1}, \eta_{n+1}^{(k)}) + C = g(\eta_{n+1}^{(k)}), \quad k = 0, 1, \dots$$

a) Man gebe die Fixpunktgleichung an, die dieser Iteration zugrunde liegt.

b) Man untersuche die Konvergenzbedingung für das allgemeine Iterationsverfahren (Fixpunktiteration, Picard-Verfahren).

Hinweis: Man prüfe $|g'(y)| < 1$.

c) Man formuliere das Verfahren als Nullstellengleichung $F(y) = 0$ und wende darauf das Newton-Verfahren an.

Es ist bekannt, dass das Newton-Verfahren bei hinreichend guter Startnäherung quadratisch konvergiert. Man überprüfe hier seine Konvergenzbedingung.

Inwieweit hat die Wahl der Schrittweite darauf Einfluss?

d) Man teste die Ergebnisse mit dem AWP

$$y' = \frac{y}{x} - y^2, \quad y(1) = 2, \quad \text{Lösung } y(x) = \frac{2}{x}.$$

14. Kann im Wurzelkriterium auch $|\xi_i| < 1$ für alle i gelten, falls das lineare MSV konsistent ist?

Begründung.

15. Man schreibe eine Funktion oder Prozedur für das klassische Adams-Bashforth(3)-Verfahren.

16. Man ermittle die Koeffizienten α_i, β_i der Adams-Bashforth(n)-Verfahren für $n = 1, 2, \dots, 10$.

Hinweis: Man nutze ein CAS.

Serie 7 - Stabilität

1. Sind folgende gDGL bzw. SysgDGL steif?

Man wähle geeignete AB sowie Parameter und löse das AWP.

a) $y'' + 2000y' + 10^6y = 0, \quad 0 \leq x \leq 1$

Lösung $y(x) = (c_1 + c_2x)e^{-1000x}$

b) $y'' + 100001y' + 10^5y = \sin(x), \quad 0 \leq x \leq 1$

Lösung $y(x) = c_1e^{-x} + c_2e^{-100000x} - \frac{100001}{20000000002} \cos(x) + \frac{99999}{20000000002} \sin(x)$

c) $\ddot{x} + 100.1\dot{x} + 1000x = x^2, \quad x = x(t), \quad -100 \leq t \leq 100$

Lösung der homogenen gDGL $x_{hom}(t) = c_1e^{-t(1001-3\sqrt{66889})/20} + c_2e^{-t(1001+3\sqrt{66889})/20}$
 $= c_1e^{-11.25563830t} + c_2e^{-88.84436170t}$

d) $\ddot{x} + 100.1\dot{x} + 1000x = -x^2, \quad x = x(t), \quad -1 \leq t \leq 1$

Lösung der homogenen gDGL $x_{hom}(t) = c_1e^{-t(1001-3\sqrt{66889})/20} + c_2e^{-t(1001+3\sqrt{66889})/20}$

e) $\ddot{x} - 100.1\dot{x} + 1000x = t^3, \quad x = x(t), \quad -100 \leq t \leq 100$

Lösung $x(t) = c_1e^{t(1001+3\sqrt{66889})/20} + c_2e^{t(1001-3\sqrt{66889})/20}$
 $+ \frac{2408409003}{5000000000000} + \frac{2706003}{50000000000}t + \frac{3003}{10000000}t^2 + \frac{1}{1000}t^3$

f) $y' = \begin{pmatrix} 1 & 1 \\ \omega^2 & 1 \end{pmatrix} y, \quad \omega > 0$

Lösung $y_1(t) = c_1e^{(1+\omega)t} + c_2e^{(1-\omega)t}, \quad y_2(t) = \omega(c_1e^{(1+\omega)t} - c_2e^{(1-\omega)t})$

g) $y' = \begin{pmatrix} 0 & 1 \\ -1 & -\frac{1}{100} \end{pmatrix} y$

Lösung $y_1(t) = e^{-t/200} [c_1 \sin(t\sqrt{39999}/200) + c_2 \cos(t\sqrt{39999}/200)]$
 $y_2(t) = 1/200 e^{-t/200} [c_1(\sqrt{39999} \cos(t\sqrt{39999}/200) - \sin(t\sqrt{39999}/200))$
 $- c_2(\sqrt{39999} \sin(t\sqrt{39999}/200) + \cos(t\sqrt{39999}/200))]$

h) $y' = \begin{pmatrix} 0 & 1 \\ -1 & -\frac{1}{100} \end{pmatrix} y, \quad y_1(0) = 0, \quad y_2(0) = 1$

Lösung $y_1(t) = 200/\sqrt{39999} e^{-t/200} \sin(t\sqrt{39999}/200)$
 $= 1.0000125 e^{-0.005t} \sin(0.9999875t)$
 $y_2(t) = e^{-t/200} [\cos(t\sqrt{39999}/200) - 1/\sqrt{39999} \sin(t\sqrt{39999}/200)]$
 $= e^{-0.005t} [\cos(0.9999875t) - 0.005000062501 \sin(0.9999875t)]$

i) $y' = \begin{pmatrix} 0 & 1 \\ -1 & -k \end{pmatrix} y, \quad k \geq 0$

Lösung $y_1(t) = c_1 e^{t(-k+\sqrt{k^2-4})/2} + c_2 e^{t(-k-\sqrt{k^2-4})/2}$
 $y_2(t) = c_1 (-k + \sqrt{k^2-4})/2 e^{t(-k+\sqrt{k^2-4})/2}$
 $+ c_2 (-k - \sqrt{k^2-4})/2 e^{t(-k-\sqrt{k^2-4})/2}$

Fallunterscheidung $k < 2, k = 2, k > 2$

2. Man bestimme mittels der Test-DGI $y' = \lambda y$, $\lambda \in \mathbb{C}$, die Stabilitätsfunktion $F(z) = F(h\lambda)$ und den Stabilitätsbereich $\mathcal{E} = \{z = h\lambda \in \mathbb{C} : |F(\lambda, h)| \leq 1\}$ folgender Verfahren.

- Euler-Cauchy-Verfahren (Rechteckregel links, PZV),
- Heun-Verfahren (Sehnentrapezregel, HV, VPZV),
- Collatz-Verfahren (Tangententrapezregel, MPZV),
- (3,3)-Verfahren von Kutta,
- KRKV.

Man folgere daraus, dass das charakteristische Polynom des (r, p) -RKV (r Stufenzahl, p Ordnung) die Form hat

$$\phi(\xi) = \xi - F(z) = \xi - \left(1 + \sum_{i=1}^r z^i \varepsilon_i\right), \quad \varepsilon_i \text{ eindeutig bestimmte Konstanten.}$$

Man vergleiche ihre Bereiche absoluter Stabilität.

3. Für das AWP $y' = -\lambda y$, $y(x_0) = y_0$, $\lambda > 0$, kann man im n -ten Schritt folgendes PC-Verfahren definieren.

$$\bar{\eta}_{n+1} = \eta_n + hf(x_n, \eta_n) = (1 - h\lambda)\eta_n \quad (\text{PZV}),$$

$$\eta_{n+1}^{(k+1)} = \eta_n + \frac{h}{2}[f(x_n, \eta_n) + f(x_{n+1}, \eta_{n+1}^{(k)})], \quad \eta_{n+1}^{(0)} = \bar{\eta}_{n+1}, \quad k = 0, 1, \dots, \quad (\text{HV}).$$

Für $k = 0$ erhalten wir die Verstärkungsfunktion (Stabilitätsfunktion)

$$F_0 = F(h\lambda) = 1 - h\lambda + \frac{(h\lambda)^2}{2},$$

aus der sich die Stabilitätsbedingung $|h\lambda| < 2$ ergibt.

a) Wird durch weitere Iterationen der Korrekturformel diese Stabilitätsforderung abgeschwächt?

Hinweis: Man untersuche dazu die Verstärkungsfunktion

$$F_k(z) = F_k(h\lambda) = \frac{1 - \frac{z}{2} - 2\left(-\frac{z}{2}\right)^{k+3}}{1 + \frac{z}{2}}, \quad z = h\lambda.$$

b) Was ist zur Konvergenz des Iterationsverfahrens zu sagen?

4. Man bestimme die Stabilitätsfunktion $F(z)$ und den Stabilitätsbereich \mathcal{E}

a) der impliziten (1,2)-Gauß-Formel

$$\eta_{n+1} = \eta_n + hk_1, \quad k_1 = f\left(x_n + \frac{h}{2}, \eta_n + \frac{h}{2}k_1\right),$$

b) der impliziten (2,4)-Gauß-Formel

$$\eta_{n+1} = \eta_n + \frac{h}{2}(k_1 + k_2),$$

$$k_1 = f\left(x_n + \frac{3 - \sqrt{3}}{6}, \eta_n + h\left(\frac{1}{4}k_1 + \frac{3 - 2\sqrt{3}}{12}k_2\right)\right),$$

$$k_2 = f\left(x_n + \frac{3 + \sqrt{3}}{6}, \eta_n + h\left(\frac{3 + 2\sqrt{3}}{12}k_1 + \frac{1}{4}k_2\right)\right).$$

5. Die Lösung des AWP (lineare homogene gDGL 2. Ordnung)

$$y'' + 12y' + 20y = 0, \quad y(0) = 4, \quad y'(0) = 0,$$

ist $y(x) = -e^{-10x} + 5e^{-2x}$, asymptotisch stabil, und es gilt $\lim_{x \rightarrow \infty} y(x) = 0$.

a) Man überführe die gDGL in ein SysgDGL 1. Ordnung

$$\begin{aligned} y' &= f(x, y, z), & y(0) &= y_0, \\ z' &= g(x, y, z), & z(0) &= z_0. \end{aligned}$$

b) Man notiere das PZV mit Schrittweite h komponentenweise sowie in Matrixdarstellung

$$\eta_{n+1} = M\eta_n, \quad \eta_n = (\eta_{1n}, \eta_{2n})^T \text{ mit der Transformationsmatrix } M.$$

c) Man zeige, dass das PZV genau dann stabil ist, wenn $h < 0.2$ gilt.

Hinweis: Für die angenommene diskrete Lösung der Form $u_n = c\xi^n$ heißt Stabilität $|\xi| < 1$.

d) Man berechne die Eigenwerte λ der Matrix M und leite daraus die Stabilität ab.

e) Man rechne einige Schritte des PZV mit $h = \frac{1}{6}$.

Wie ist die Stabilitätsbedingung zu verändern, um beim PZV Oszillationen in der Lösung zu vermeiden?

6. Gegeben ist das lineare AWP

$$\begin{aligned} \dot{y}_1 &= -y_1 - 24y_2, & y_1(0) &= 2, \\ \dot{y}_2 &= -25y_2, & y_2(0) &= 1, \\ \dot{y}_3 &= 125y_2 - 150y_3, & y_3(0) &= 0. \end{aligned}$$

a) Man bestimme die exakte Lösung.

b) Man transformiere die Aufgabe auf das entkoppelte SysgDGL

$$\dot{z}_i = \lambda_i z_i, \quad z_i(0) = z_{i0}, \quad i = 1, 2, 3.$$

c) Man ermittle eine Schrittweitenschranke $h_0 > 0$ für das PZV und für das KRKV, sodass bei allen Schrittweiten $h < h_0$ das jeweilige Verfahren absolut stabil ist.

7. Die sogenannten θ -Verfahren sind parametrische ESV der Form

$$\eta_{n+1} = \eta_n + h[(1 - \theta)f(x_n, \eta_n) + \theta f(x_{n+1}, \eta_{n+1})], \quad \theta \geq 0,$$

oder als One-leg-Verfahren

$$\eta_{n+1} = \eta_n + hf((1 - \theta)x_n + \theta x_{n+1}, (1 - \theta)\eta_n + \theta\eta_{n+1}), \quad 0 \leq \theta \leq 1.$$

Sei $f \in \mathcal{C}^2(\mathbb{R}^2)$.

a) Man bestimme die Stabilitätsfunktion $F(z) = F(h\lambda)$ der Verfahren.

b) Das Verfahren heißt A-stabil (unbeschränkt absolut stabil), falls $\mathcal{E} \supseteq \mathbb{C}^- = \{z \mid \Re z < 0\}$ bzw. \mathcal{E} einen genügend großen unbeschränkten Teil von \mathbb{C}^- enthält.

Für welche Werte θ sind die Verfahren A-stabil?

8. Das linear-implizite Euler-Verfahren für SysgDGL lautet

$$\eta_{n+1} = \eta_n + h \left[I - h \frac{\partial f}{\partial y}(x_n, \eta_n) \right]^{-1} f(x_n, \eta_n), \quad n = 0, 1, 2, \dots,$$

mit der Einheitsmatrix I .

Man bestimme die Stabilitätsfunktion $F(z) = F(h\lambda)$.

Ist das Verfahren A-stabil?

9. Man betrachte das SysgDGL $y' = f(x, y) = Ay$ gemäß

$$\begin{aligned} y_1' &= y_2, & y_1(0) &= 0, \\ y_2' &= -y_1 - \frac{1}{100}y_2, & y_2(0) &= 1. \end{aligned}$$

- Man berechne die beiden Eigenwerte der Matrix $A = \begin{pmatrix} \frac{\partial f}{\partial y} \end{pmatrix}$, die allgemeine Lösung und diskutiere die Steifheit des SysgDGL.
- Man löse das AWP für $x \in [0, 1]$.
- Man berechne mit dem PZV und $h = 0.1, 0.05, 0.02, 0.01$ eine Näherung. Welche Stabilitätsbedingung muss eingehalten werden?
- Man wende das A-stabile implizite Euler-Verfahren an, indem das entstehende Gleichungssystem nach den Unbekannten $(\eta_{1,n+1}, \eta_{2,n+1})$ aufgelöst wird.
- Man vergleiche die Ergebnisse.

10. Man bestimme für die beiden impliziten MSV

$$\begin{aligned} \eta_{n+1} &= \frac{4}{3}\eta_n - \frac{1}{3}\eta_{n-1} + \frac{2}{3}hf_{n+1}, \\ \eta_{n+1} &= \eta_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}), \end{aligned}$$

- das charakteristische Polynom $\phi(\xi)$,
- den Stabilitätsbereich \mathcal{E} .
- Welche Stabilitätseigenschaft besitzen diese Verfahren?

11. Für die Test-DGL $y' = -\lambda y$, $y(x_0) = y_0$, $\lambda > 0$, untersuche man das implizite MSV von Milne-Simpson

$$\begin{aligned} \eta_{n+1} &= \eta_{n-1} - \frac{h\lambda}{3}(\eta_{n+1} + \eta_n + \eta_{n-1}) \quad \text{bzw.} \\ (1+z)\eta_{n+1} + 4z\eta_n + (z-1)\eta_{n-1} &= 0, \quad z = \frac{h\lambda}{3}, \end{aligned}$$

auf seine Stabilität.

Man benutze dazu die Ansatzfunktion $\eta_n = c\xi^n$ für die Differenzgleichung mit der Forderung $|\xi| < 1$.

Index

- Anfangswertproblem, 8, 363
 - Anfangsbedingung, 2, 8, 358, 363
- Begleitmatrix, 532
- Bessel-Funktionen, 297, 629
- Differentialgleichung
 - n -ter Ordnung, 355
 - Abkühlung, 324
 - Abrollen, 12
 - Airy, 56
 - autonome, 356
 - Bernoulli, 319
 - Bessel, 12, 297
 - Bewegung, 11
 - Bewegung im Gravitationsfeld, 367
 - Clairaut, 328
 - Doppelpendel, 16
 - Duffing, 3, 4, 11
 - Eigenschwingung, 12
 - Emden, 11
 - explizite, 355
 - Fadenpendel, 11, 13
 - Ferroresonanzstabilisator, 4
 - Fluss, 364, 392
 - freier Fall, 8, 9
 - Frequenzteilerschaltung, 4
 - gedämpfte Schwingung, 12
 - gewöhnliche, 1, 355
 - harmonische Schwingung, 25
 - Hermite, 326
 - implizite, 355
 - Kettenlinie, 331
 - Kummer, 326
 - Kurven mittels Abstandsbedingung, 21
 - Legendre, 305
 - Liouville, 12
 - logistische, 30
 - Mathieu, 12
 - Pendel, 5, 10
 - Pendel-Diagramm, 12
 - Relaxationsschwingung, 3
 - Riccati, 133, 303, 470, 710
 - Rollpendel, 56
 - Schaltkreis, 1
 - Schwingkreis, 2
 - Schwingung, 4, 10, 357
 - System, 5, 356
 - Ökosystem, 31
 - Abrollen, 330
 - Ameisen-Kampf, 29
 - Atwood-Maschine, 13
 - chaotisches, 13
 - Diabetes, 33
 - Drei-Wellen-Wechselwirkung, 4
 - Dreikörperproblem, 5
 - Epidemiologie, 33
 - gekoppeltes Netzwerk, 3
 - Hénon-Heiles-Modell, 12
 - Hamilton, 12, 13
 - Hopf, 49
 - Immunsystem, 33
 - Jäger-Beute-Modell, 29
 - Kennlinie des Asynchronmotors, 4
 - Lorenz-Modell, 7
 - Lotka-Volterra, 28, 278
 - Populationsmodell, 285
 - Räuber-Beute-Modell, 258, 268, 270, 357
 - Rössler, 4
 - Ressourcen-Modell, 272
 - steifes, 48, 52, 391, 606
 - Van der Pol, 27, 282
 - Verhulst, 49
 - Zweisppezimodell, 28
 - Test-, 550
 - asymptotisch stabil, 550
 - stabil, 550
 - turbulente Strömung, 9

- Van der Pol, 3, 10, 24, 27
 Wachstum, 30, 356, 366
 Wurf, 253
 Zellwachstum, 34
- Differentialoperatoren, 354, 408, 410
- Differenz, 333
 - dividierte, 336
 - Schema, 336
 - Rückwärtsdifferenz, 333, 501
 - Steigung, 336
 - Vorwärtsdifferenz, 333, 501
- Differenzgleichung, 501, 523
 - homogene, 525
 - Stabilitätskriterium, 527
 - Wachstumsbeschränkung, 526
 - Wurzelkriterium, 527
- Diskretisierungsfehler
 - globaler, 393, 431, 528
 - lokaler, 392, 396, 398, 399, 544
- Diskretisierungsverfahren, 379
- Einschrittverfahren, 395
 - (2,2)-IRKV, 685
 - A-stabil, 552
 - Anstieg, 396, 398
 - explizites, 395, 398
 - implizites, 673
 - Parameterschemata, 682
 - Rosenbrock-Wanner, 697
 - ROW-Verfahren, 699
 - W-Methoden, 699
 - implizites Euler-Verfahren, 558, 675
 - Konsistenz, 399
 - Konsistenzordnung, 399, 407
 - L-stabil, 552
 - linear-implizites Euler-Verfahren, 679
 - One-leg-Verfahren, 687
 - Parameterschema, 401, 418, 421
 - (1,1)-IEV, 684
 - (1,2)-Gauß-Legendre-Verfahren, 685
 - (1,2)-IRKV, 685
 - (2,2)-RKV, 421
 - (2,3)-IRKV, 689
 - (2,4)-IRKV, 689
 - (3,2)-RKV, 421
 - (3,2)-RKV, Heun-Formel, 421
 - (3,3)-Kutta-Verfahren, 421
 - (3,3)-RKV, 421
 - (3,3)-RKV, Kuntzmann-Formel, 422
 - (3,3)-RKV, Nystrom-Formel, 421
 - (3,3)-RKV, RK-Fehlberg-Formel, 421
 - (4,4)-RKV, 422
 - (4,4)-RKV mit minimalem Fehler, 422
 - (4,4)-RKV, Ceschino-Formel, 422
 - (4,4)-RKV, England-Formel, 422
 - (4,4)-RKV, Gill-Modifikation, 422
 - (4,4)-RKV, Kuntzmann-Formel, 422
 - (5,4)-RKV, Scraton-Formel, 423
 - (5,5)-RKV, Shanks-Formel, 424
 - (6,5)-RKV, Butcher-Verfahren, 423
 - (6,5)-RKV, Fehlberg-Verfahren, 426
 - (6,5)-RKV, Lawson-Verfahren, 426
 - (6,5)-RKV, Luther-Verfahren, 426
 - (6,5)-RKV, Nyström-Formel, 424
 - (6,5)-RKV, Sarafyan-Formel, 424
 - (6,6)-RKV, Shanks-Formel, 424
 - (7,5)-RKV, Dormand-Prince-Verfahren, 425
 - (7,6)-RKV, Butcher-Verfahren, 425
 - (8,6)-RKV, Huta-Formel, 425
 - eingebettetes CSIRK(13,8), 460
 - eingebettetes CSIRK(6,5), 459
 - eingebettetes CSIRK(8,6), 459
 - eingebettetes DOPRI5(4), 458
 - eingebettetes Fehlberg5(6), 457
 - eingebettetes Fehlberg5(6) mod., 457
 - eingebettetes Prince-Dormand5(6), 458
 - eingebettetes RKF1(2), 456
 - eingebettetes RKF2(3), 456
 - eingebettetes RKF4(5), 456
 - eingebettetes RKF7(8), 460
 - eingebettetes RKV von Ceschino, 456
 - eingebettetes RKV von Sarafyan, 456
 - eingebettetes Verner5(6), 457
 - eingebettetes Verner5(6) mod., 458
 - Gauß-Verfahren, 689
 - Gear(1)-Verfahren, 684
 - Heun-Verfahren, 421
 - klassisches RKV 4. Ordnung, 422
 - Kutta-3/8-Formel, 422
 - Lobatto-IIIC-Verfahren, 690
 - modifiziertes Polygonzugverfahren, 421
 - Polygonzugverfahren, 421
 - Radau-IA-Verfahren, 690

- Radau-II-A(1)-Verfahren, 684
 - Polygonzugverfahren, 15, 22, 378
 - Runge-Kutta-Fehlberg-Verfahren
 - eingebettetes, 442, 454
 - Runge-Kutta-Verfahren
 - s -stufiges, 401, 404
 - eingebettetes, 454
 - Heun-Verfahren, 397
 - klassisches 4. Ordnung, 408
 - Kutta-3/8-Formel, 408
 - modifiziertes Polygonzugverfahren, 395
 - Polygonzugverfahren, 379
 - Stabilität, 551
 - Stabilitätsgebiet, 551
 - stark stabil, 552
 - unbeschränkt absolut stabil, 552
 - Verfahrensfunktion, 398, 399
 - Verfahrenskonstruktion, 407
 - Zuwachs, 396
- Fixpunktgleichung, 509, 678
 - Fixpunktiteration, 509, 678
- Gleichgewichtslage
 - instabile, 366, 476
 - stabile, 443, 481
- Gleitkommaarithmetik, 345
- Gleitkommaformat, 346
- Gompertzsche Wachstumskurve, 34
- Hermite-Polynom, 326
- Heunsche-Funktion, 298
- Isokline, 358
- Jordansche Normalform, 532
- Konsistenzordnung, 399
- Kummer-Funktionen, 326
- Lagrange-Interpolation, 498
- Legendre-Polynome, 305
- Lipschitz-Stetigkeit, 364
- Mehrschrittverfahren, 497
 - Adams-Bashforth, 501
 - $(r + 1)$ -stufiges, 503
 - n -stufiges, 504
 - Adams-Moulton, 501, 507
 - s -stufiges, 508
 - klassisches, 508
- Adams-Störmer, 501
- charakteristische Polynome, 528
- D-Stabilität, 527
- Erste Dahlquist-Barriere, 548
- erzeugende Polynome, 528
- explizites, 498
- Gear-BDF, 501, 522
- implizites, 498
- Konsistenz, 528
- Konvergenz, 528
- lineares, 497
- lineares r -stufiges, 500
- Milne-Simpson, 501
- nullstabil, 527
- Nullstabilität, 527
- Nyström, 501
- Ordnung, 517
- schwach stabil, 527
- Stabilität, 526, 568
- Stabilitätsgebiet, 569
- stark stabil, 527
- Zweite Dahlquist-Barriere, 527
- Newton-Interpolation, 337, 498
- Nichtlineare Gleichung, 508
 - Newtonsches Iterationsverfahren, 509, 678, 680, 684, 697
- Orbit, 13
- Pascalsches Dreieck, 334
- Phasenkurve, 6, 24, 356
- Phasenraum, 24, 356
- Prädiktor-Korrektor-Verfahren, 511
 - Korrektor, 511
 - Ordnung, 511
 - Prädiktor, 511
- Quadraturformel
 - Newton-Côtes-Formeln, 498
 - Simpson-Regel, 499, 519
- Richtungsfeld, 23, 69, 357
 - Linienelemente, 23, 358
 - Richtungselemente, 23, 358
- Satz von Picard-Lindelöf, 363
- Schrittweitensteuerung, 376

- adaptive, 440, 464
- asymptotische Fehlerschätzung, 464, 484
- Einbettung, 454
- Extrapolation, 464
- Fehlberg-Trick, 440
- Grob- und Feinrechnung, 464, 467
- Kontrollverfahren, 440
- Runge-Prinzip, 464
- Schrittkenzahl, 461
- variable, 484
- Stabilität, 548
 - absolute, 549
 - Einschrittverfahren
 - A-stabil, 552
 - absolut stabil, 552
 - Grenze des Stabilitätsgebiets, 551
 - L-stabil, 552
 - Stabilitätsgebiet, 551
 - Stabilitätsintervall, 566
 - stark stabil, 552
 - Mehrschrittverfahren
 - $A(\alpha)$ -stabil, 591
 - $A(\gamma)$ -stabil, 591
 - $A(0)$ -stabil, 591
 - absolut stabil, 568
 - Stabilitätsgebiet, 568, 569
 - steif-stabil, 591
 - Wachstum, 382
 - Wachstumsfunktion, 382, 551
- Steifheit, 606, 674
- Taylor-Reihen, 353
- Trajektorie
 - Lösung, 364, 378
 - explodierende, 366
 - globale, 366
 - kollabierende, 368
 - orthogonale, 23, 358

Literaturverzeichnis

- [1] ABELL, M.L. und J.P. BRASELTON: *Differential Equations with Mathematica*. AP Professional, A Division of Harcourt Brace & Company Boston 1993.
- [2] ABRAHAM, R.H. und C.D. SHAW: *Dynamics - The Geometry of Behavior*. Vol. 1, 2, 3 (Periodic, Chaotic, Global Behavior) The Visual Mathematics Library. Aerial Press, Inc. Santa Cruz 1983.
- [3] ABRAMOWITZ, M. und I. STEGUN, EDITORS: *Handbook of Mathematical Functions*. Dover Publication Inc. New York 1965.
- [4] AINSWORTH, M., LEVESLEY, J., LIGHT, W.A. und M. MARLETTA: *Theory of Numerics of Ordinary and Partial Differential Equations*. Oxford University Press Oxford 1995.
- [5] ALBRECHT, P.: *Die numerische Behandlung gewöhnlicher Differentialgleichungen*. Akademie-Verlag Berlin 1979.
- [6] ARGYRIS, J. und H.-P. MLEJNIK: *Die Methode der Finiten Elemente in der elementaren Strukturmechanik*. Band III. Einführung in die Dynamik (bes. Kap. 23-25). Friedr. Vieweg & Sohn Braunschweig/Wiesbaden 1988.
- [7] ARNOLD, V.I.: *Obyknovennyye differentsialnyye uravneniya*. Izd. Nauka Moskva 1971. Dt. Ausgabe : *Gewöhnliche Differentialgleichungen*. Springer-Verlag Berlin 1980.
- [8] AULBACH, B.: *Gewöhnliche Differentialgleichungen*. Spektrum Akademischer Verlag 2004.
- [9] BABOVSKY, H.: *Numerische Mathematik III*. Numerik gewöhnlicher Differentialgleichungen. Vorlesungsskript IfMath, TU Ilmenau 2006.
- [10] BENKER, H.: *Differentialgleichungen mit MATHCAD und MATLAB*. Springer-Verlag Berlin 2005.
- [11] BERESIN, I. S. und N.P. SHIDKOW: *Numerische Methoden*. Bd. 1,2. DVW Berlin 1970, 1971.
- [12] BODEWIG, E.: *Matrix Calculus*. North-Holland Publishing Company Ferce sur Sarthe 1959.
- [13] BRAUN, M.: *Differentialgleichungen und ihre Anwendungen*. Springer-Verlag Berlin 1979, 1991, 1994.
- [14] BRÄUNING, G.: *Gewöhnliche Differentialgleichungen*. Fachbuchverlag Leipzig 1964.
- [15] BURRAGE, K.: *Parallel and Sequential Methods for Ordinary Differential Equations*. Oxford University Press Oxford 1995.
- [16] BUTCHER, J.C.: *The numerical analysis of ordinary differential equations. Runge-Kutta and general linear methods*. John Wiley& Sons, New York 1987-

- [17] CHERKASOVA, M.P.: *Collected Problems in Numerical Methods*. Akademie-Verlag Berlin 1972.
- [18] COLLATZ, L.: *Funktionalanalysis und numerische Mathematik*. Springer-Verlag Berlin 1964.
- [19] COLLATZ, L.: *The Numerical Treatment of Differential Equations*. 3rd Edition. Springer-Verlag Berlin 1960.
- [20] DEKKER, K.: und J.G. VERWER: *Stability of Runge-Kutta methods for stiff nonlinear differential equations*. North-Holland Amsterdam 1984.
- [21] DEMAILLY, J.-P.: *Gewöhnliche Differentialgleichungen*. Theoretische und numerische Aspekte, Lehrbuch Angewandte Mathematik. Vieweg Verlag Braunschweig 1994.
- [22] DEMUTH, M. und B.-W. SCHULZE (Hrsg.): *Differential Equations, Asymptotic Analysis, and Mathematical Physics*. Reihe Mathematical Research. Wiley-VCH Berlin 1996.
- [23] DEUFLHARD, P. und F. BORNEMANN: *Numerische Mathematik*. 2. Integration gewöhnlicher Differentialgleichungen. Walter de Gruyter Berlin 1994.
- [24] DEUFLHARD, P. und H. HOHMANN: *Numerische Mathematik*. Walter de Gruyter Berlin 1991.
- [25] DEV-PASCAL. Bloodshed Dev-Pascal, Booldshed Software - Providing Free Software to the internet Community.
<http://www.bloodshed.net>
- [26] ENGELN-MÜLLGES, G. und F. REUTTER:
1. *Formelsammlung zur Numerischen Mathematik mit FORTRAN 77-Programmen*. Bibliogr. Institut Mannheim 1988.
2. *Formelsammlung zur Numerischen Mathematik mit Turbo Pascal-Programmen*. BI-Wissenschaftsverlag Mannheim 1991.
3. *Numerik-Algorithmen mit ANSI C-Programmen*. (auch für TP, FORTRAN). BI-Wissenschaftsverlag Mannheim 1993.
- [27] FAIRES, J.D. und R.L. BURDEN: *Numerische Methoden. Näherungsverfahren und ihre praktische Anwendung*. Spektrum Akademischer Verlag Heidelberg 1994.
- [28] FORSYTHE, J., M. MALCOLM and C. MOLER: *Computer Methods for Mathematical Computations*. Prentice Hall 1979.
- [29] FORTUNA, Z., B. MACUKOW und J. WĄSOWSKI: *Metody numeryczne*. Podręczniki akademickie eit. WNT Warszawa 1982.
- [30] FRIED, I.: *Numerical Solution of Differential Equations*. Computer Science an Applied Mathematics. Academic Press New York 1979.
- [31] FURLAN, P.: *Das gelbe Rechenbuch 3 für Ingenieure, Naturwissenschaftler und Mathematiker*. Verlag Martina Furlan Dortmund 1997.
- [32] GANTMACHER, F.R.: *Teorija matricz*. Moskwa 1954, 1967.
- [33] GOLUB, G.H. und J.M. ORTEGA: *Wissenschaftliches Rechnen & Differentialgleichungen*. Heldermann Verlag Berlin 1995. Hrsg. der dt. Übersetzung: Grigorieff, R. D.
- [34] GOLUBITSKY, M. und M. DELLNITZ: *Linear algebra and differential equations using MATLAB*. Brooks/Cole Pub. Co Pacific Grove 1999.
- [35] GOERING, H., FELGENHAUER, A., LUBE, G., ROOS, H.-G. und L. TOBISKA: *Singularly perturbed differential equations*. Mathematical Research 13. Akademie-Verlag Berlin 1983.

- [36] GRADSTEIN, I.S. und I.M. RYSHIK: *Tablicy integralow, summ, rjadow i proiswedenij*. Izdatelstwo Nauka Moskwa 1971.
- [37] GRIGORIEFF, R.D.: *Numerik gewöhnlicher Differentialgleichungen*. Bd. I : Einschrittverfahren. Teubner Stuttgart 1972.
Bd.II : Mehrschrittverfahren. Teubner Stuttgart 1977.
- [38] GUCKENHEIMER, J. und P. HOLMES: *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. Springer-Verlag New York 1983, 1986, 1990.
- [39] HAIRER, E., NORSETT, S.P. und G. WANNER: *Solving Ordinary Differential Equations*. Vol. 1. Nonstiff problems. Springer-Verlag Berlin 1987.
- [40] HAIRER, E. und G. WANNER: *Solving Ordinary Differential Equations*. Vol. 2. Stiff and differential-algebraic problems. Springer-Verlag Berlin 1991.
- [41] HÄMMERLIN G. und K.-H. HOFFMANN: *Numerische Mathematik*. Grundwissen Mathematik 7. Springer-Verlag Berlin 1991.
- [42] HANKE-BOURGEOIS, M.: *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Teubner Stuttgart 2002.
- [43] HEERMANN, D.W.: *Theoretische Physik I. Klassische Mechanik*. Vorlesungsmaterial Universität Heidelberg Institut für Theoretische Physik 2004.
- [44] HEMKER, P.W.: *A numerical study of stiff two-point boundary problems*. Akademische Dissertation. Mathematisches Zentrum Amsterdam 1977.
- [45] HENNING, K. und S. KUTSCHA: *Informatik im Maschinenbau*, 4. Auflage. Springer-Verlag Berlin 1994.
- [46] HERMANN, M.: *Numerische Mathematik*, 3. Auflage. R. Oldenbourg Verlag München 2011.
- [47] HERMANN, M.: *Numerik gewöhnlicher Differentialgleichungen, Anfangs- und Randwertprobleme*. R. Oldenbourg Verlag München 2004.
- [48] HINZE, J.; ED.: *Numerical Integration of Differential Equations and Large Linear Systems*. Proceeding of two Workshops at the University of Bielefeld 1980. Lecture Notes in mathematics, Vol. 968, Springer-Verlag 1982.
- [49] HOLMES, P. and J. GUCKENHEIMER: *Nonlinear oscillations, dynamical systems, and bifurcation of vector fields*. Springer-Verlag 1983, 1990.
- [50] ISAACSON, E. und H. B. KELLER: *Analyse numerischer Verfahren*. Edition Leipzig 1972.
- [51] JANKOWSKA, J. und M. JANKOWSKI: *Przegląd metod i algorytmów numerycznych*. Bd. 1. WNT Warszawa 1981.
- [52] KALITKIN, N.N.: *Tshislelnye metody*. Nauka Moskwa 1978.
- [53] KAMKE, E.: *Differentialgleichungen*. Lösungsmethoden und Lösungen I. B. G. Teubner Stuttgart 1977.
- [54] KANTOROWITSCH, L. W. und G. P. AKILOW: *Funktionalanalysis in normierten Räumen*. Akademie-Verlag Berlin 1978.
- [55] KIELBASIŃSKI, A. und H. SCHWETLICK: *Numerische lineare Algebra*. DVW Berlin 1988.
- [56] KINCAID, D. und W. CHENEY: *Numerical Analysis*. Brooks/Cole Publ. Comp. Pacific Grovw 1991.

- [57] KNOBLOCH, H.W. und F. KAPPEL: *Gewöhnliche Differentialgleichungen*. B. G. Teubner Stuttgart 1974.
- [58] KNOBLOCH, J. und J. STEIGENBERGER: *Gewöhnliche Differentialgleichungen*. Skript IfMath TU Ilmenau 1996.
- [59] KORSCH, H.J.: *Chaos: a program collection for the PC*. Springer-Verlag Berlin 1998.
- [60] KORSCH, H.J.: *Chaos: A program collection for the PC*. With 250 fig., many numerical experiments, and two 3 1/2MS-DOS-diskettes. Springer-Verlag Berlin 1994.
- [61] KOSE, K., SCHRÖDER, R. und K. WIELICZEK: *Numerik sehen und verstehen*. Ein kombiniertes Lehr- und Arbeitsbuch mit Visualisierungssoftware. Vieweg Braunschweig 1992.
- [62] KÖCKLER, N.: *Numerische Algorithmen in Softwaresystemen*: unter besonderer Berücksichtigung der NAG-Bibliothek. B. G. Teubner Stuttgart 1990.
- [63] KUBICEK, M. and M. MAREK: *Computational Methods in Bifurcation Theory and Dissipative Structures*. Springer Series in Comput. Physics 1983.
- [64] KUYPERS, F.: *Klassische Mechanik*. 5. Auflage, Wiley-VCH Verlag GmbH Weinheim 1997.
- [65] LORENZ, E.N.: *The essence of chaos*. Paperback University of Washington Press 1995.
- [66] MAESS, G.: *Vorlesungen über Numerische Mathematik*. Bd. I, II. Akademie-Verlag Berlin 1984, 1988.
- [67] MAPLE. Maplesoft, a division of Waterloo Maple Inc. Waterloo Canada, 1981-2011
<http://www.maplesoft.com>
- [68] MATLAB. The Language of Technical Computing. The MathWorks Inc. 1984-2010
<http://www.mathworks.com>
- [69] MATWIEJEW, N.M.: *Zadania z równań różniczkowych zwyczajnych*. PWN Warszawa 1976, Übers. aus dem Russ.
- [70] MICHLIN, S.G. und C.L. SMOLICKI: *Metody przybliżone rozwiązywania równań różniczkowych i całkowych*. PWN Warszawa 1970, Übers. aus dem Russ.
- [71] MIRANKER, W.L.: *Numerical Methods for Stiff Equations and Singular Perturbation Problems*. Mathematics and Its Applications 5. D. Reidel Publishing Comp. Dordrecht 1981.
- [72] MOHR, R.: *Numerische Methoden in der Technik*. Ein Lehrbuch mit Matlab-Routinen. Vieweg Verlag Wiesbaden 1998.
- [73] NEUNDORF, W.: *Numerische Mathematik*. Vorlesungen, Übungen, Algorithmen und Programme. Shaker Aachen 2002.
- [74] NEUNDORF, W.: *Gewöhnliche Differentialgleichungen*. Beispiele, Modelle, Verfahren, Software. Preprint M11/98 IfMath TU Ilmenau 1998.
- [75] NEUNDORF, W.: *Grundlagen der numerischen linearen Algebra*. Preprint M04/04 IfMath TU Ilmenau 2004.
- [76] NEUNDORF, W.: *Lösungsmethoden mit Maple*. Betrachtung von Fehlern und Kondition sowie Darstellungsmöglichkeiten. Preprint M08/03 IfMath TU Ilmenau 2003.
- [77] NEUNDORF, W. und B. WALTHER: *Praktikumsaufgaben für die Fachgebiete Algorithmen und Programmierung, Grundlagen der Informatik und Informatik und Programmierung*. Skriptum TU Ilmenau 1997.

- [78] OEVEL, W.: *Einführung in die Numerische Mathematik*. Spektrum Akademischer Verlag Heidelberg 1996.
- [79] OPFER, G.: *Numerische Mathematik für Anfänger*. Vieweg Studium Grundkurs Mathematik Wiesbaden 1993, 3. überarbeitete und erw. Auflage 2001.
- [80] PARKER, T.S. und L.O. CHUA: *Practical Numerical Algorithms for Chaotic Systems*. Springer-Verlag New York Berlin 1989.
- [81] PHILIPPOW, E.: *Grundlagen der Elektrotechnik*. Verlag Technik 1988.
- [82] PHILIPPOW, E. und W. BÜNTIG: *Analyse nichtlinearer dynamischer Systeme der Elektrotechnik*. Carl Hanser Verlag München 1992.
- [83] PLATO, R.: *Numerische Mathematik kompakt*. Grundlagenwissen für Studium und Praxis. Vieweg Verlag Wiesbaden 2000.
- [84] QUARTERONI, A., R. SACCO und F. SALERI: *Numerische Mathematik 2*. Springer-Verlag New York Berlin 2002.
- [85] RALSTON, A.: *A First Course in Numerical Analysis*. McGraw-Hill New York 1965.
- [86] RICE, J.R.: *Numerical Methods, Software and Analysis*. 2nd Edition. Academic Press Inc. Boston 1993.
- [87] ROOS, H.-G. und H. SCHWETLICK: *Numerische Mathematik*. Das Grundwissen für jedermann. B. G. Teubner Stuttgart 1999.
- [88] ROSS, SHEPLEY L.: *Differential equations*. 3 Ed. John Wiley & Sons, Inc. New York 1984.
- [89] RUTISHAUSER, H.: *Vorlesungen über Numerische Mathematik*. Bd. 1,2. Birkhäuser Verlag Basel 1976.
- [90] SCHABACK, R. und H. WERNER: *Numerische Mathematik*. Springer-Verlag Berlin 1993.
- [91] SCHILDER, F.: *Ein portables C++ Programm zu numerischen Integration von Anfangswertproblemen*. Preprint M 1/97 TU Ilmenau März 1997.
- [92] SCHWARZ, H.R.: *Numerische Mathematik*. B. G. Teubner Stuttgart 1988.
- [93] SCHWARZ, H.R., H. RUTISHAUSER und E. STIEFEL: *Numerik symmetrischer Matrizen*. B. G. Teubner Stuttgart 1972.
- [94] SCHWETLICK, H. und H. KRETZSCHMAR: *Numerische Verfahren für Naturwissenschaftler und Ingenieure*. Fachbuchverlag Leipzig 1991.
- [95] SEYDEL, R.: *Practical bifurcation and stability analysis*. Springer-Verlag New York 1994.
- [96] SEYDEL, R.: *From equilibrium to chaos*. Elsevier New York 1988.
- [97] SHAMPINE, L.F. und M.K. GORDON: *Computerlösung gewöhnlicher Differentialgleichungen*. Vieweg & Sohn Braunschweig 1984.
- [98] SPÄTH, H.: *Numerik*. Vieweg Wiesbaden 1994.
- [99] STEEB, W.-H.: *Chaos and Fractals. Algorithms and Computation*. B.I. Wissenschaftsverlag Mannheim 1992.
- [100] STOER, J.: *Numerische Mathematik I*. Heidelberger Taschenbücher 105. Springer-Verlag Berlin 1993.

- [101] STOER, J. und R. BULIRSCH: *Einführung in die Numerische Mathematik II*. Heidelberger Taschenbücher 114. Springer-Verlag Berlin 1990.
- [102] STRAMPP, W. und V. GRANZHA: *Differentialgleichungen mit Mathematica*. Vieweg-Verlag Wiesbaden 1995.
- [103] STREHMEL, K. und R. WEINER: *Numerik gewöhnlicher Differentialgleichungen*. Teubner Studienbücher Mathematik. B. G. Teubner Stuttgart 1995.
- [104] STUART, A. und A. HUMPHRIES, A.: *Dynamical Systems and Numerical Analysis*. Cambridge Monographs on Applied and Computational Mathematics 2. Cambridge Univ. Press 1996.
- [105] STUMMEL, F. und K. HAINER: *Praktische Mathematik*. B. G. Teubner Stuttgart 1982.
- [106] TODA, M.: *Theory of Nonlinear Lattices*. 2nd enl. edition, Springer Berlin 1989.
- [107] THOMPSON, W.J.: *Non-Linearity and Chaos in Engineering Dynamics*. John Wiley & Sons 1994
- [108] THOMPSON, W.J.: *Atlas for Computing Mathematical Functions*. An Illustrated Guide for Practitioners with Programs in FORTRAN 90 resp. C and Mathematica. John Wiley & Sons 1997.
- [109] TROGER, H. und A. STEINDL: *Nonlinear stability and bifurcation theory*. Springer-Verlag 1991.
- [110] TUFILLARO, N.B.; ABBOTT, T.A. and D.J. GRIFFITHS: *Swinging Atwoods machine*. Amer. Journal Phys. 52 (10) October 1984, pp. 895-903.
- [111] UHLMANN, H.: *Grundlagen der elektrischen Modellierung und Simulationstechnik*. Akad. Verlagsgesellschaft Geest & Portig. Wiss. Monographien der ET, Bd.5 1977.
- [112] UNGLAUB, F.: *Anwendungen des Lagrange-Formalismus an Beispielen der Oberstufenphysik*. GSI Darmstadt theory.gsi.de/~vanhees/faq/lagrange/ 2003.
- [113] VERNER, J.H.: *Explicit Runge-Kutta methods with estimates of the local truncation error*. SIAM J. Numer. Anal. 15 (1978) 772-790.
- [114] VOGT, W.: *Zur Numerik gewöhnlicher Differentialgleichungen*. Teil 1, Anfangswertprobleme. Preprint M02/09 IfMath TU Ilmenau 2002.
- [115] WALTER, W.: *Gewöhnliche Differentialgleichungen*. Eine Einführung. Heidelberger Taschenbücher Bd. 110. Springer-Verlag Berlin 1972.
- [116] WALTER, W.: *Gewöhnliche Differentialgleichungen*. 7. Auflage. Springer-Verlag Berlin 2000.
- [117] WENZEL, H.: *MINÖL 7/1,2*. Gewöhnliche Differentialgleichungen, Band 1, 2. B. G. Teubner Verlagsgesellschaft 1973.
- [118] WERNER, H. und H. ARNDT: *Gewöhnliche Differentialgleichungen*. Eine Einführung in Theorie und Praxis. Springer-Verlag Berlin 1986.
- [119] WERNER, W.: *Mathematik lernen mit Maple*. Bd. 1,2. Ein Lehr- und Arbeitsbuch für das Grundstudium. dpunkt Heidelberg 1996, 1998.
- [120] WESTERMANN, T.: *Mathematik für Ingenieure mit Maple*. Springer 1996.
- [121] ZWILLINGER, D.: *Handbook of Differential Equations*. 3rd ed., Academic Press Boston 1997.

