

Preprint No. M 11/06

Numerik gewöhnlicher
Differentialgleichungen -
Anfangswertprobleme

Werner Neundorf

Februar 2011

Impressum:

Hrsg.: Leiter des Instituts für Mathematik

Weimarer Straße 25

98693 Ilmenau

Tel.: +49 3677 69-3621

Fax: +49 3677 69-3270

<http://www.tu-ilmenau.de/math/>

Technische Universität Ilmenau
Fakultät für Mathematik
und Naturwissenschaften
Institut für Mathematik
<http://www.tu-ilmenau.de/math/>

Postfach 10 05 65
D - 98684 Ilmenau
Germany
Tel.: 03677/69 3267
Fax: 03677/69 3272
Telex: 33 84 23 tuil d.
email:
werner.neundorf@tu-ilmenau.de

Preprint No. M 06/11

Numerik
gewöhnlicher Differentialgleichungen
Anfangswertprobleme
Werner Neundorf

Februar 2011

Zusammenfassung

Die vorliegende Arbeit ist aus Vorlesungen und Seminaren für die Mathematik- und Ingenieurstudenten an der TU Ilmenau hervorgegangen. Sie widmet sich der numerischen Behandlung von Anfangswertproblemen gewöhnlicher Differentialgleichungen, im notwendigen Maße der Theorie, mehr jedoch den Fragen der Algorithmisierung sowie der Nutzung von Software. Dabei werden bezüglich der Implementierung die CAS MATLAB und Maple verwendet.

Es ist klar, dass in einer solchen Arbeit nur ein Bruchteil der mit dieser Problematik verbundenen Aspekte dargestellt werden kann. Im Literaturverzeichnis gibt es weiterführende Fachliteratur.

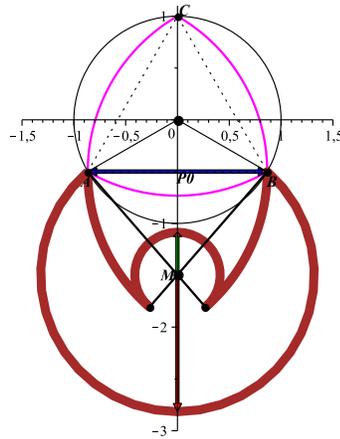
Im Skript findet der Leser zahlreiche Hinweise, Beispiele und Illustrationen zur numerischen Behandlung von Differentialgleichungen im Studium und in der Praxis. Die Übungsaufgaben dienen zur Vertiefung der Erkenntnisse.

Inhaltsverzeichnis

1	Gewöhnliche Differentialgleichungen - Einführung	1
1.1	Grundbegriffe	1
1.2	Das Eulersche Polygonzugverfahren (PZV)	18
1.3	Erste Fehlerbetrachtungen	32
2	Einschrittverfahren	35
2.1	Das Konzept der Einschrittverfahren (ESV)	35
2.2	Runge-Kutta-Verfahren (RKV)	40
2.2.1	Parameterschemata für explizite ESV	47
2.3	Konvergenz von ESV	50
2.4	Schrittweitensteuerung	55
2.4.1	Schrittweitenkontrolle mit lokalem Diskretisierungsfehler	55
2.4.2	Schrittweitenkontrolle mit Schrittkenzahl	65
2.4.3	Schrittweitensteuerung mit Grob- und Feinrechnung	68
3	Lineare Mehrschrittverfahren	85
3.1	Das Konzept der linearen Mehrschrittverfahren (MSV)	85
3.1.1	Adams-Bashforth-Verfahren	89
3.1.2	Adams-Moulton-Verfahren	94
3.1.3	Prädiktor-Korrektor-Verfahren	96
3.2	Die Ordnung linearer MSV	103
3.3	Homogene lineare Differenzgleichungen	107
3.4	Konsistenz und Konvergenz von MSV	110
4	Stabilität von Ein- und Mehrschrittverfahren	120
4.1	Absolute Stabilität	120
4.1.1	Stabilitätsgebiet von ESV	122
4.2	Stabilität von MSV	138
4.2.1	Stabilitätsgebiet von MSV	139
4.3	Integration steifer Systeme	169
4.3.1	Systeme mit ausgeprägter Steifheit	206
4.3.2	Steife Systeme und Linearisierung	230
5	Übungsserien	236
	Literaturverzeichnis	271

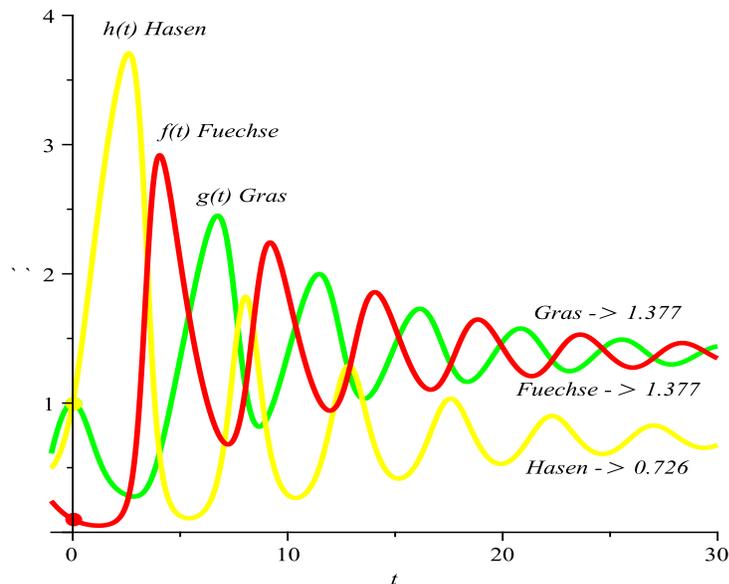
Eric Temple Bell, 1883-1960

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



Archimedes

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



1 Gewöhnliche Differentialgleichungen - Einführung

1.1 Grundbegriffe

Die Arbeit befasst sich mit der numerischen Lösung gewöhnlicher Differentialgleichungen. Diese sind wie folgt definiert.

[1.1] Definition

(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

$$x'(t) = f(t, x(t)) \quad (1.1)$$

oder kurz $x' = f(t, x)$, heisst **explizite gewöhnliche Differentialgleichung 1. Ordnung**, kurz **gDGL**. Die Gleichung

$$F(t, x(t), x'(t)) = 0 \quad (1.2)$$

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

$$x^{(n)} = f(t, x, x', \dots, x^{(n-1)}) \quad (1.3)$$

heißt **explizite gewöhnliche Differentialgleichung n -ter Ordnung**. Die Gleichung

$$F(t, x, x', \dots, x^{(n-1)}, x^{(n)}) = 0 \quad (1.4)$$

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

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t)) \quad (1.5)$$

heißt **System expliziter gewöhnlicher Differentialgleichungen 1. Ordnung**, kurz **SysgDGL**, für die vektorwertige Funktion $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$.

Seine implizite Form ist

$$\mathbf{F}(t, \mathbf{x}(t), \mathbf{x}'(t)) = \mathbf{0}. \quad (1.6)$$

(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 $\dot{x}(t) = f(t, x(t))$, $y'(x) = f(x, y(x))$ usw.

[1.2] Bemerkungen (a) Eine Funktion $x(t)$ (bzw. $\mathbf{x}(t)$) auf einem Intervall I ist Lösung der gDGL (des SysgDGL), falls für jedes $t \in I$ die entsprechende Gleichung aus (1.1), ..., (1.6) erfüllt ist.

(b) Mit der Lösung der gDGL $x(t)$ bzw. $\mathbf{x}(t)$ des SysgDGL ist die **Phasenkurve** (Bahnkurve, Integralkurve) $(t, x(t))$ bzw. $(t, \mathbf{x}(t))$ definiert.

Das Gebiet Ω wird auch als **Phasenraum** bezeichnet.

(c) gDGL höherer Ordnung können stets als SysgDGL geschrieben werden, indem die Gleichung (1.3) 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 (1.3) ist äquivalent zur Lösung des Systems

$$\mathbf{z}'(t) = \begin{pmatrix} z_2(t) \\ z_3(t) \\ z_4(t) \\ \vdots \\ z_n(t) \\ f(t, \mathbf{z}(t)) \end{pmatrix} = \begin{pmatrix} z_2(t) & & & & & \\ & z_3(t) & & & & \\ & & z_4(t) & & & \\ & & & \ddots & & \\ & & & & z_n(t) & \\ f(t, z_1(t), z_2(t), z_3(t), \dots, z_n(t)) & & & & & \end{pmatrix} = \tilde{\mathbf{f}}(t, \mathbf{z}(t)). \quad (1.7)$$

(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 (1.3) ist äquivalent zur Lösung des SysgDGL

$$\tilde{\mathbf{x}}' = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}) \quad \text{mit} \quad \tilde{\mathbf{f}}(\tilde{\mathbf{x}}) = (1, f(x_1, x_2))^T. \quad (1.8)$$

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.

[1.3] Beispiele

(a) Die explizite gDGL erster Ordnung

$$x' = -\lambda x \quad \text{mit} \quad \lambda > 0 \quad (1.9)$$

beschreibt ein exponentiell abklingendes System, wie es zum Beispiel beim Zerfall eines radioaktiven Stoffes der Fall ist. Die allgemeine Lösung ist

$$x(t) = C e^{-\lambda t} \quad (1.10)$$

mit einer frei zu wählenden Konstante C .

(b) Die implizite gDGL

$$m x''(t) + b x'(t) + D x(t) = 0 \tag{1.11}$$

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 Dämpfung b ist die allgemeine Lösung gegeben durch

$$x(t) = C e^{-\delta t} \sin(\omega_d t + \phi) \tag{1.12}$$

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 Systeme gewöhnlicher Differentialgleichungen ist das sog. **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 gDGL

$$x' = ax - bxy \quad \text{und} \quad y' = -cy + dxy \tag{1.13}$$

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 [1.2](b) kann die gDGL zweiter Ordnung aus Teil (b) umformuliert werden in ein System erster 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

$$\mathbf{x}'(t) = \begin{pmatrix} 0 & 1 \\ -D/m & -b/m \end{pmatrix} \mathbf{x}(t) \tag{1.14}$$

mit der rechten Seite $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$. ◀

Wie wir in den vorangegangenen 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 (1.1) 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 gegeben durch den Tangentenvektor $(1, f(t_0, c))^T$. Diese Information erlaubt es uns, den ungefähren Verlauf der Lösung mit Hilfe des Richtungsfeldes grafisch zu skizzieren und motiviert die folgende Definition.

[1.4] Definition Das zur gDGL $x' = f(t, x)$ definierte Vektorfeld

$$R(t, x) = \frac{1}{\sqrt{1 + f^2(t, x)}} \begin{pmatrix} 1 \\ f(t, x) \end{pmatrix} \tag{1.15}$$

heißt das **Richtungsfeld** der gDGL.

Um das Richtungsfeld in jedem Gebiet Ω_1 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. 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 Richtungsfelds in MATLAB definiert man eine Funktion als MATLAB-Datei `dfield.m`.

```
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 die Funktionsprozedur als MATLAB-Datei `f0.m`

```
function y = f(t,x)
    y = x;
```

und ihre Aufrufe gemäß

```
% 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 die gDGL mit ihren allgemeinen Lösungen

$$\begin{aligned}
 x' &= x, & x(t) &= Ce^t, \\
 x' &= 10(x - t^2/(1+t^2)) + 2t/(1+t^2)^2, & x(t) &= Ce^{10t} + t^2/(1+t^2), \\
 x' &= \sin(1/x) - 2, \\
 x' &= x(1-x), & x(t) &= Ce^t/(1+C(e^t-1)).
 \end{aligned}
 \tag{1.16}$$

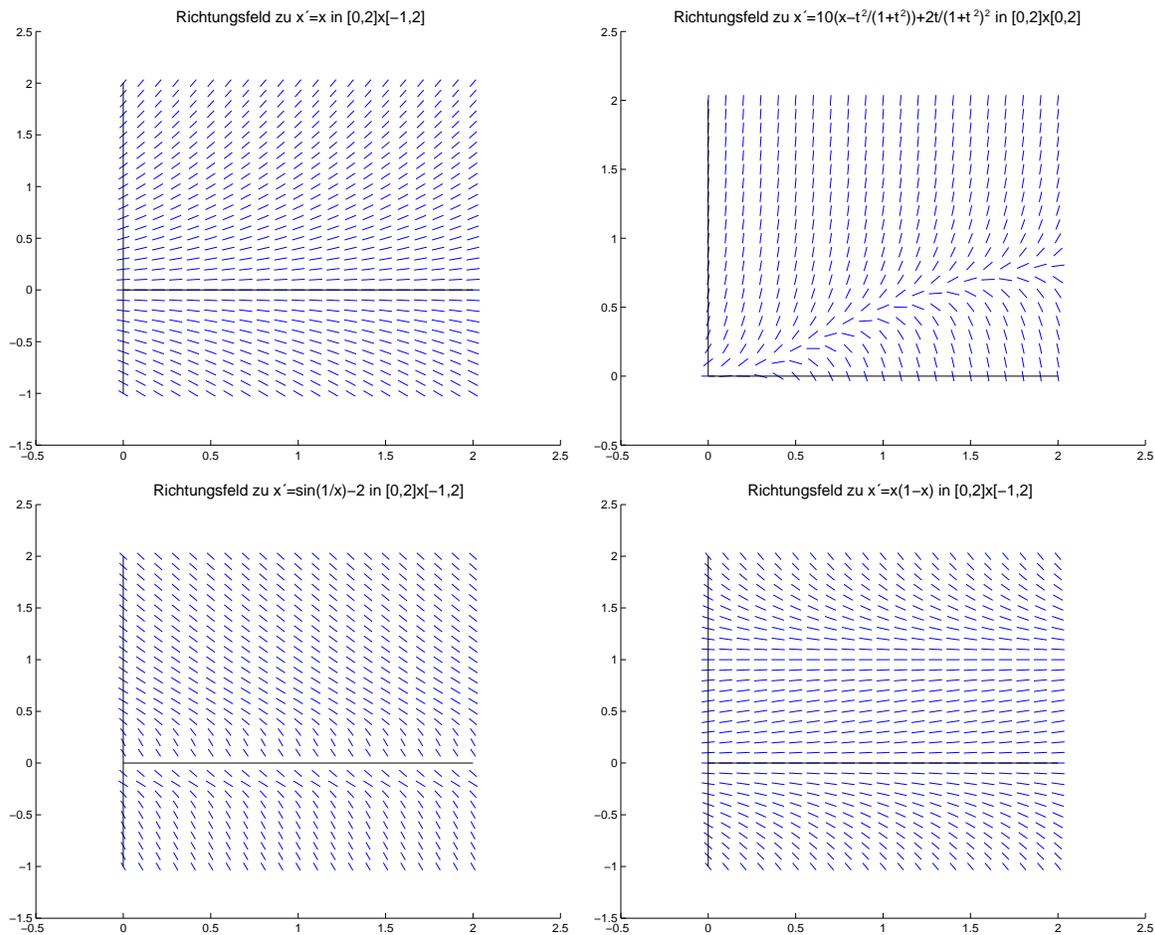


Abb. 1.1: Dateien *r_f0.ps*, *r_f1.ps*, *r_f2.ps*, *r_f4.ps*, Richtungsfelder zu den 4 gDGL

Natürlich kann man auch das MATLAB-Kommando `quiver` verwenden.

Wir notieren es im Rahmen von 2 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]');

```

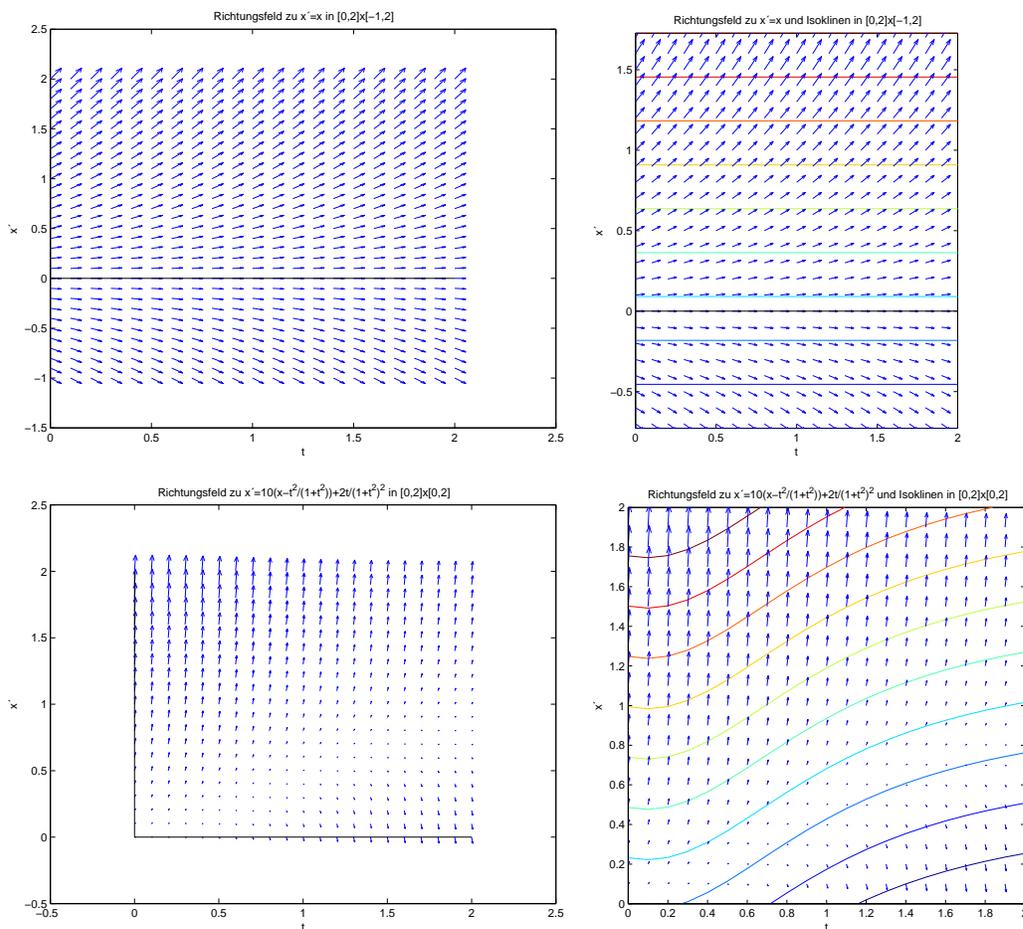
```

xlabel('t'); ylabel('x');
hold on
plot([0 2],[0 0],'k-',[0 0],[-1 2],'k-');
print r_f0b.ps -dpasc
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 -dpasc
hold off

```

Die Grafik der Variante 3 hat bezüglich der Achsen den gleichen Maßstab und enthält zusätzlich 10 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.



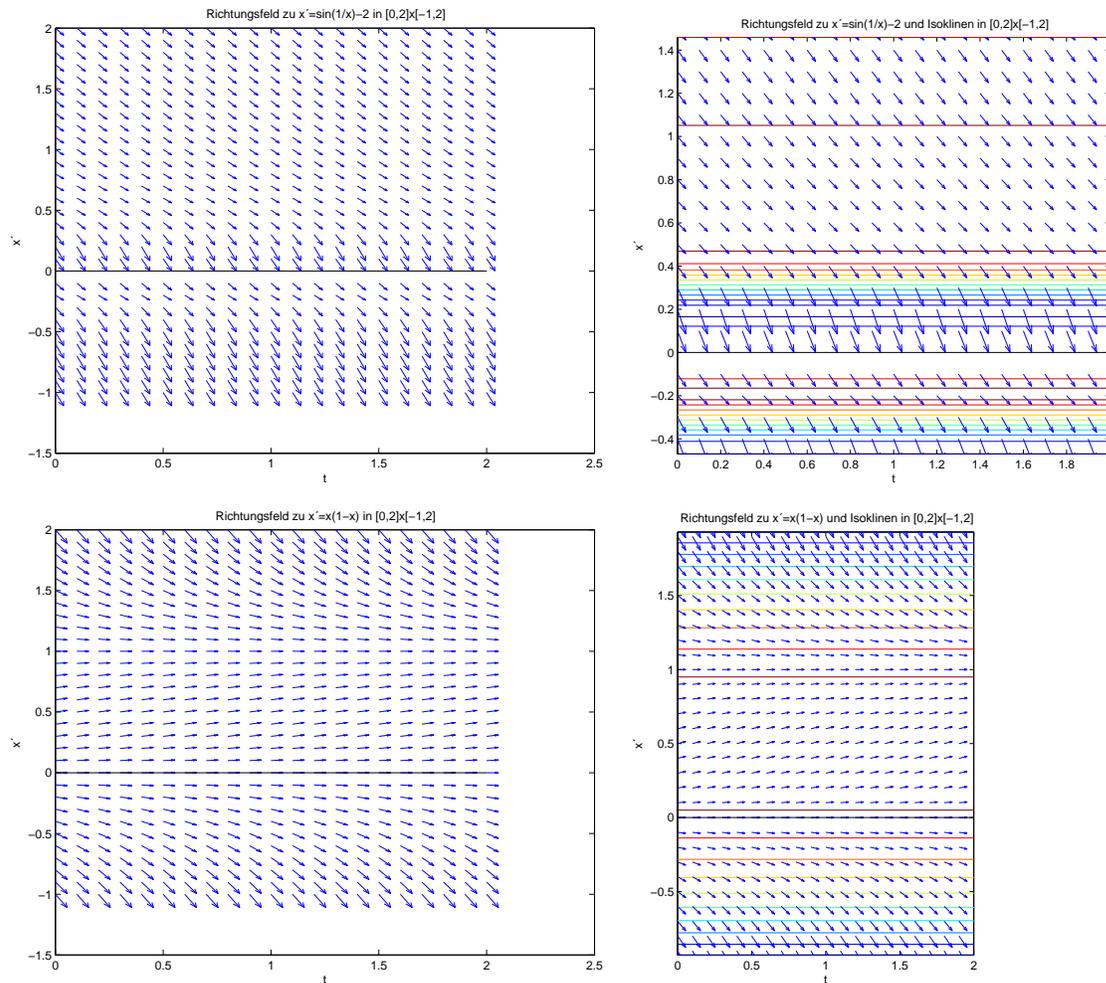


Abb. 1.2: Dateien *r_f0b.ps*, *r_f0c.ps*, *r_f1bb.ps*, *r_f1cc.ps*,
r_f2b.ps, *r_f2c.ps*, *r_f4b.ps*, *r_f4c.ps*,
 Richtungsfelder (mit Isoklinen) zu den vier gDGL

Wenn die Anstiege der Linienelemente sehr unterschiedlich sind, hilft auch der mögliche Skalierungsparameter in `quiver` nicht. Die eigene Definition `dfield` ist teilweise anschaulicher.

Ähnlich kann man in Maple verfahren.

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 dem Package 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,linecolor=[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);
```

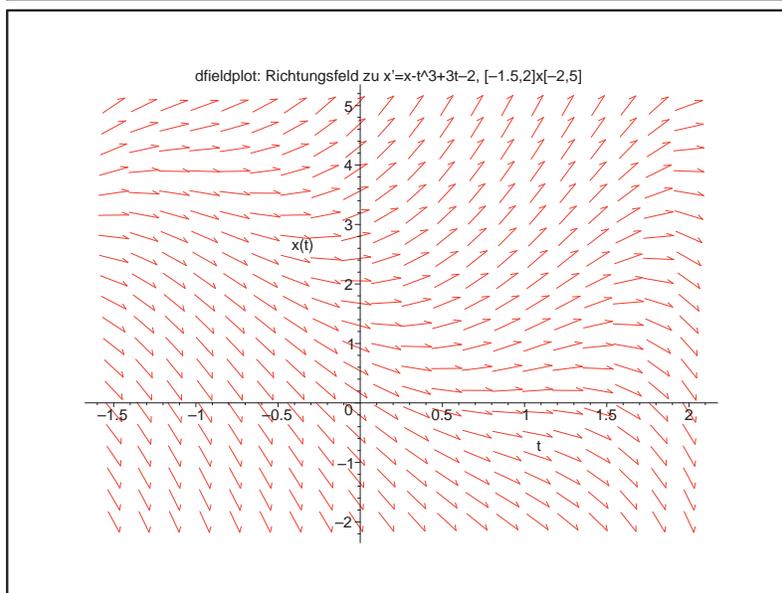
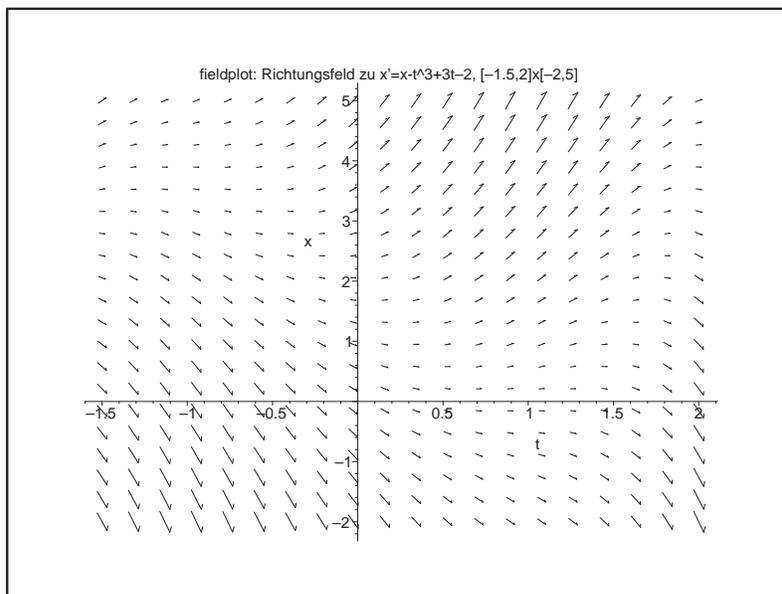


Abb. 1.3:

Dateien

richt1.ps, richt2.ps,

Richtungsfelder zur gDGL

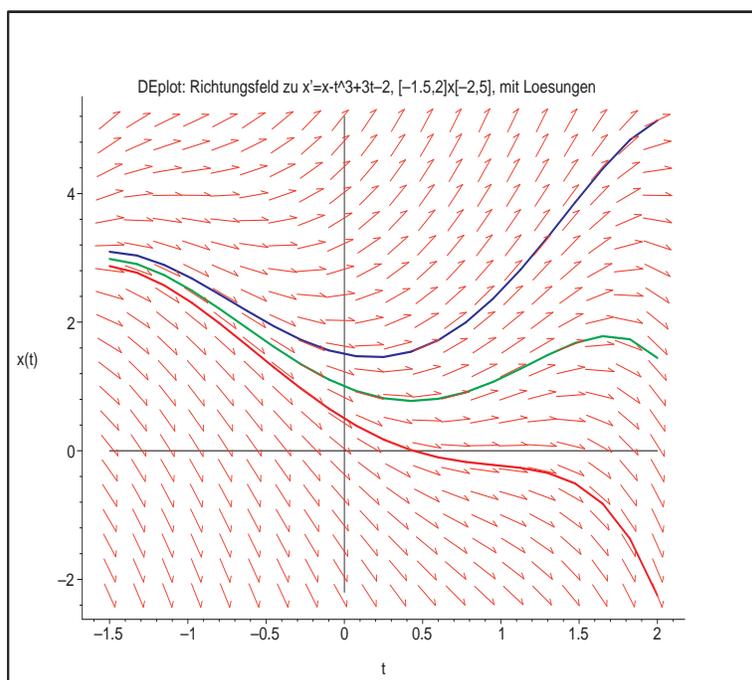
$$x' = f(t, x)$$

$$= x - t^3 + 3t - 2,$$

Maple-Kommandos

`fieldplot`

`dfieldplot`

**Abb. 1.4:**Datei *richt3.ps*,

Richtungsfeld zur gDGL

$$x' = f(t, x)$$

$$= x - t^3 + 3t - 2,$$

sowie Näherungslösungen

mit 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 (1.5) ist unter gewissen Bedingungen an \mathbf{f} *eindeutig*, wenn sie an einem Punkt t_0 vorgegeben ist. Dies führt auf die folgende Definition.

[1.5] Definition Unter einem **Anfangswertproblem (AWP)** verstehen wir ein SysgDGL der Form (1.5), welches ergänzt ist durch eine Bedingung der Form

$$\mathbf{x}(t_0) = \mathbf{c} = (c_1, c_2, \dots, c_n)^T \quad \text{für ein } (t_0, \mathbf{c})^T \in \Omega \quad (1.17)$$

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 gewöhnlichen Differentialgleichungen, 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.

[1.6] Satz 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 (1.5) sei in folgendem 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 gilt die Abschätzung

$$\|\mathbf{f}(t, \mathbf{x}_1) - \mathbf{f}(t, \mathbf{x}_2)\| \leq L \|\mathbf{x}_1 - \mathbf{x}_2\|. \quad (1.18)$$

Hierbei ist $\|\cdot\|$ eine beliebige Vektornorm in \mathbb{R}^n . Dann hat das AWP (1.5), (1.17) eine eindeutige Lösung $\mathbf{x}(t)$. Diese lässt sich bis zum Rand von Ω fortsetzen.

Im Folgenden sei, um die Eindeutigkeit des AWP zu gewährleisten, die Lipschitz-Stetigkeit (1.18) immer vorausgesetzt. Das SysgDGl (1.5) erzeugt eine ganze Schar von Lösungen, welche **der zu (1.5) gehörige Fluss** genannt wird und welche wir mit dem Symbol Φ bzw. $\Phi^{\cdot,\cdot}$ bezeichnen wollen. Hierbei beschreibt

$$\mathbf{x}(t) = \mathbf{x}(t; \tau, \tilde{\mathbf{x}}) = \Phi^{t,\tau} \tilde{\mathbf{x}} \quad (1.19)$$

diejenige spezielle Lösung (*Trajektorie*), welche zum Zeitpunkt τ den Wert $\tilde{\mathbf{x}}$ annimmt, also

$$\mathbf{x}(\tau) = \Phi^{\tau,\tau} \tilde{\mathbf{x}}.$$

Insbesondere notiert man die Lösung des AWP (1.5), (1.17) in der Kurzform

$$\mathbf{x}(t) = \Phi^{t,t_0} \mathbf{c}. \quad (1.20)$$

Wir wollen den Fluss im skalaren Fall (1.1) grafisch illustrieren.

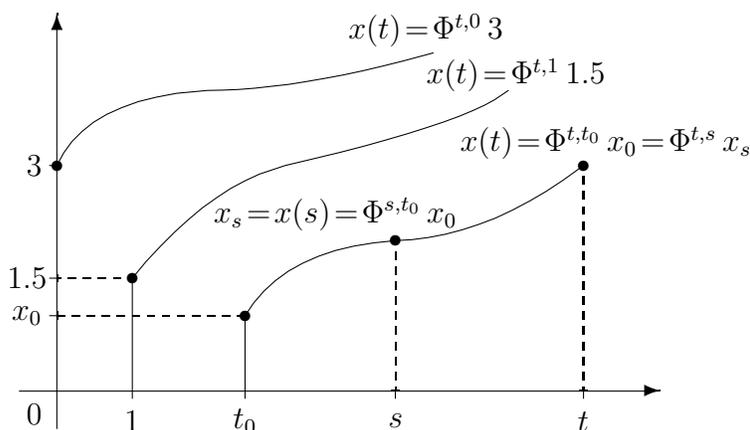


Abb. 1.5: Datei *fluss1.pic*, Lösungen $x(t)$ eines AWP als $\Phi^{t,\tau} \tilde{x}$ (1.19)

Wie sieht der Fluss eines SysgDGl aus? Sei

$$\mathbf{x}'(t) = A\mathbf{x}(t), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (1.21)$$

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 $\mathbf{x}'(t) = Ae^{A(t-t_0)} \mathbf{x}_0 = A\mathbf{x}(t)$.

Der Fluss ist also die Matrixfunktion

$$\Phi^{t,t_0} = e^{A(t-t_0)}. \quad (1.22)$$

Besonders einfach ist die Darstellung, wenn A eine Diagonalmatrix ist.

Aus $A = \text{diag}(a_1, a_2, \dots, a_n)$ und bei $\mathbf{x}(t_0) = \mathbf{x}_0 = \mathbf{c} = (c_1, c_2, \dots, c_n)^T$ folgt

$$\Phi^{t,t_0} = e^{A(t-t_0)} = \text{diag}(e^{a_1(t-t_0)}, e^{a_2(t-t_0)}, \dots, e^{a_n(t-t_0)})$$

und

$$\mathbf{x}(t) = \Phi^{t,t_0} \mathbf{c} = (c_1 e^{a_1(t-t_0)}, c_2 e^{a_2(t-t_0)}, \dots, c_n e^{a_n(t-t_0)})^T.$$

[1.7] Satz 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.

$$(a) \quad \Phi^{t_0,t_0} = I, \quad (1.23)$$

$$(b) \quad \Phi^{t,s} \Phi^{s,t_0} = \Phi^{t,t_0}. \quad (1.24)$$

Beweis:

(a) Die Formel (1.23) 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:

Beide stellen eine Lösung des AWP

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t)), \quad \mathbf{x}(s) = \mathbf{x}_s,$$

dar, d. h. ihre Werte stimmen für $t \geq s$ überein. Somit folgt

$$\begin{aligned} \mathbf{x}(t; t_0, \mathbf{x}_0) &= \mathbf{x}(t; s, \mathbf{x}_s), \\ \Phi^{t,t_0} \mathbf{x}_0 &= \Phi^{t,s} \mathbf{x}_s = \Phi^{t,s} \Phi^{s,t_0} \mathbf{x}_0, \quad \text{also} \\ \Phi^{t,t_0} &= \Phi^{t,s} \Phi^{s,t_0}. \end{aligned}$$

□

[1.8] Bemerkung Dass Existenz und Eindeutigkeit nicht für allgemeine rechte Seiten erfüllt sind, wollen wir an mehreren Beispielen demonstrieren.

(a) Gegeben sei das AWP auf $\Omega = \mathbb{R}^2$

$$x'(t) = x(t) - t^3 + 3t - 2, \quad x(t_0) = x_0 = c. \quad (1.25)$$

In den Abb. 1.3, 1.4 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 gDGL erhält man

$$x(t) = Ce^t + t^3 + 3t^2 + 3t + 5. \quad (1.26)$$

Mit der AB $x(0) = 1$ folgt $C = -4$ und die zugehörige Trajektorie lautet

$$x(t) = -4e^t + t^3 + 3t^2 + 3t + 5. \quad (1.27)$$

Damit lässt sich auch das Verhalten der Funktion für betragsgroße Werte t deuten.

(b) Gegeben sei die gDGL auf $\Omega = \mathbb{R}^2$

$$x'(t) = x^2(t). \quad (1.28)$$

Einerseits hat man die einzige globale Lösung $x(t) \equiv 0$. Dies stellt eine instabile Gleichgewichtslage dar. Wenn x nicht Null ist, findet man gemäß $x'/x^2 = 1$ durch Integration die allgemeine Lösung

$$x(t) = -\frac{1}{t+C} = \frac{1}{-C-t}. \quad (1.29)$$

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 fortgesetzte, 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 keinen Sinn, sie in einem größeren Intervall $[t_0, T]$ zu suchen.

Gegeben sei (1.28) auf $I = [a, b] = [0, 2]$ als AWP

$$x'(t) = x^2(t), \quad x(0) = 1. \quad (1.30)$$

Die eindeutige Lösung

$$x(t) = \frac{1}{1-t} \quad (1.31)$$

existiert auf $[0, 1)$, auch auf $(-\infty, 1)$, aber nicht auf dem Intervall $[0, 2]$.

Für einen noch so kleinen positiven Startwert c explodiert die Lösung immer nach langer, aber endlicher Zeit.

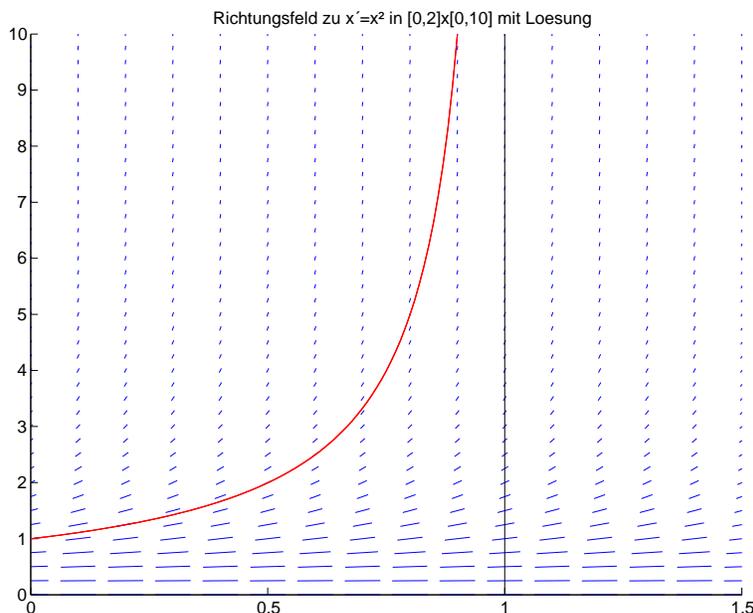


Abb. 1.6:
 Datei *r_f6.ps*,
 Richtungsfeld zur gDGL
 $x' = f(t, x) = x^2$
 sowie bei $t = 1$
 explodierende Lösung
 $x(t) = 1/(1 - t)$
 mit der AB $x(0) = 1$

(c) Gegeben sei auf $I = [a, b] = [0, 1]$ das AWP

$$x'(t) = 2\sqrt{x(t)}, \quad x(0) = 0. \tag{1.32}$$

Die allgemeine Lösung ist gegeben durch

$$x(t) = \begin{cases} (t - s)^2 & \text{für } t \geq s \geq a, \\ 0 & \text{sonst,} \end{cases} \tag{1.33}$$

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

$$x'(t) = -\frac{1}{\sqrt{x(t)}}, \quad x(0) = 1. \tag{1.34}$$

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

$$x(t) = \left(1 - \frac{3}{2}t\right)^{2/3}. \tag{1.35}$$

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 führen würde.

Der Gültigkeitsbereich der gDGL 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

$$\lim_{t \nearrow T} \text{dist}((t, x(t)), \partial\Omega) = 0.$$

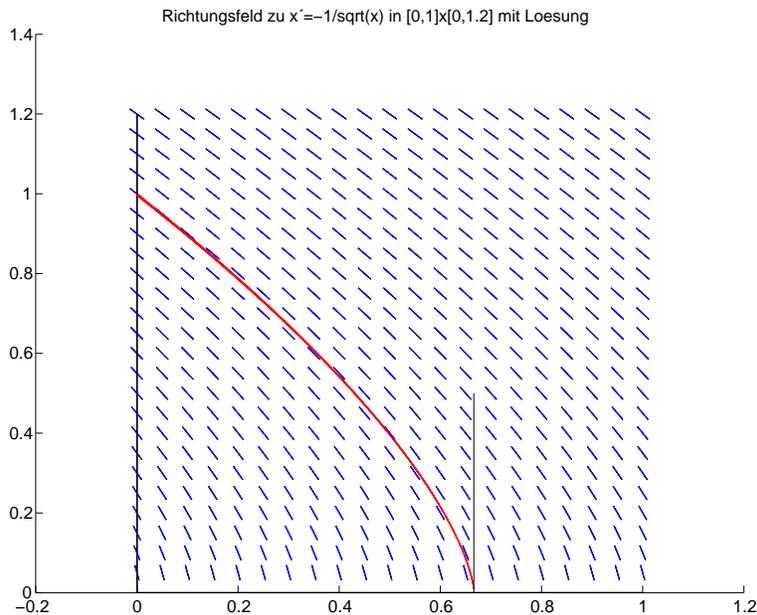


Abb. 1.7:

Datei *r_f5.ps*,

Richtungsfeld zur gDGL

$$x' = f(t, x) = -\frac{1}{\sqrt{x(t)}}$$

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

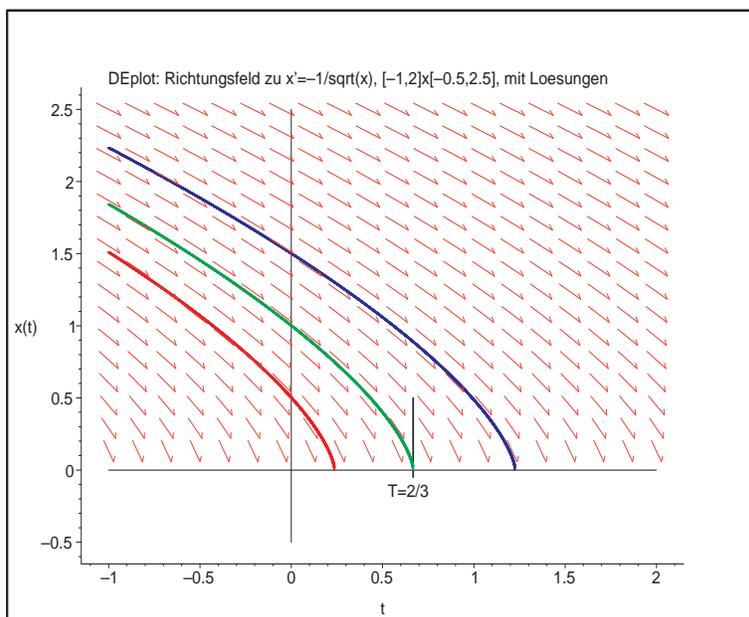


Abb. 1.8:

Datei *r_f52.ps*,

Richtungsfeld zur gDGL

$$x' = f(t, x) = -\frac{1}{\sqrt{x(t)}}$$

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 errors(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) In der Abb. 1.1 zeigten wir das Richtungsfeld der autonomen gDGL

$$x' = \sin(1/x) - 2 \quad (1.36)$$

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 gDGL ist auf dem Gebiet $\Omega = [0, +\infty) \times (0, +\infty)$ beliebig oft differenzierbar. Bei einer allgemeinen Anfangsbedingung $x(t_0) = c$ 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.

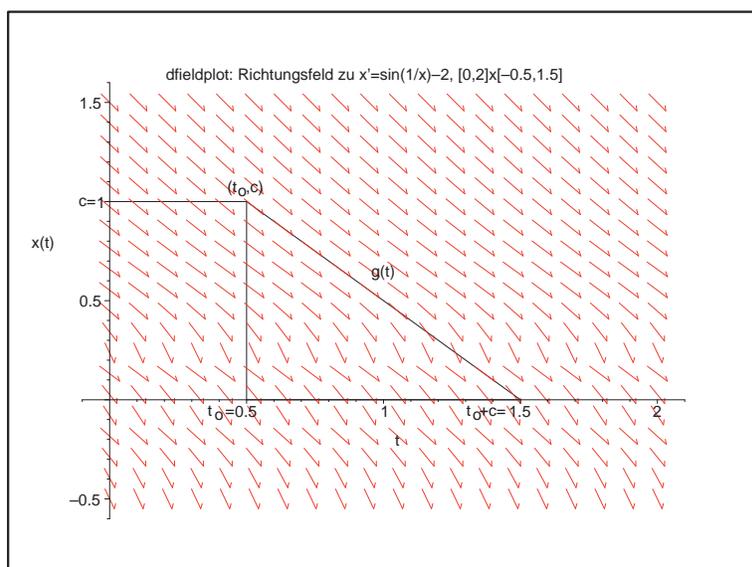


Abb. 1.9:

Datei *r_f71.ps*,
 Richtungsfeld zur gDGL
 $x' = f(t, x) = \sin(1/x) - 2$
 sowie obere Grenze
 $g(t) = c - (t - t_0)$ für
 eine durch den Punkt (t_0, c)
 verlaufende Lösung $x(t)$

Für die Anfangsbedingung $x(0) = 1$ heißt das $x(t) \leq 1 - t$ und $T \leq t_s = 1$.

Die Grenze T bestimmt man gemäß

$$\frac{dx}{dt} = \sin(1/x) - 2,$$

$$dt = \frac{dx}{\sin(1/x) - 2},$$

$$\int_0^T dt = \int_1^0 \frac{dx}{\sin(1/x) - 2},$$

$$T = \int_0^1 \frac{dx}{2 - \sin(1/x)} = 0.767\,410\,604. \tag{1.37}$$

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 $[-3, -1]$, so dass sich schwer eine Lösung aus dem Aneinanderfügen solcher Elemente erzeugen lässt.

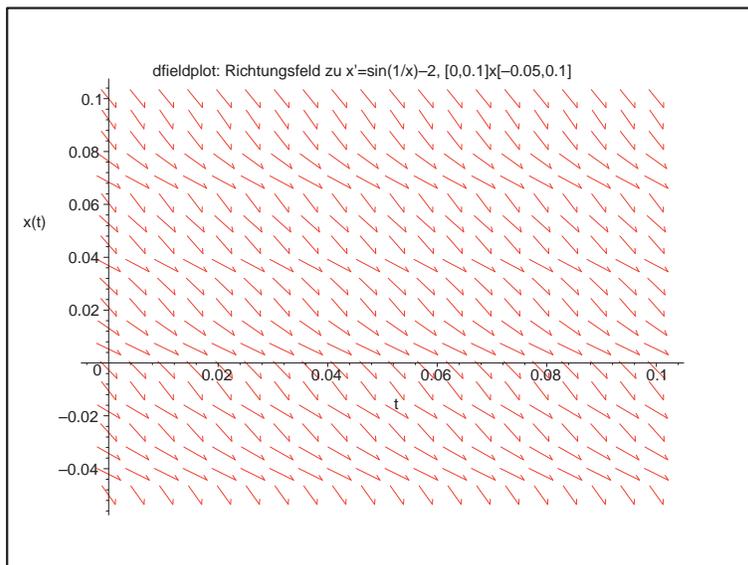


Abb. 1.10:

Datei *r_f72.ps*,

Richtungsfeld zur gDGL

$$x' = f(t, x) = \sin(1/x) - 2$$

nahe $x = 0$

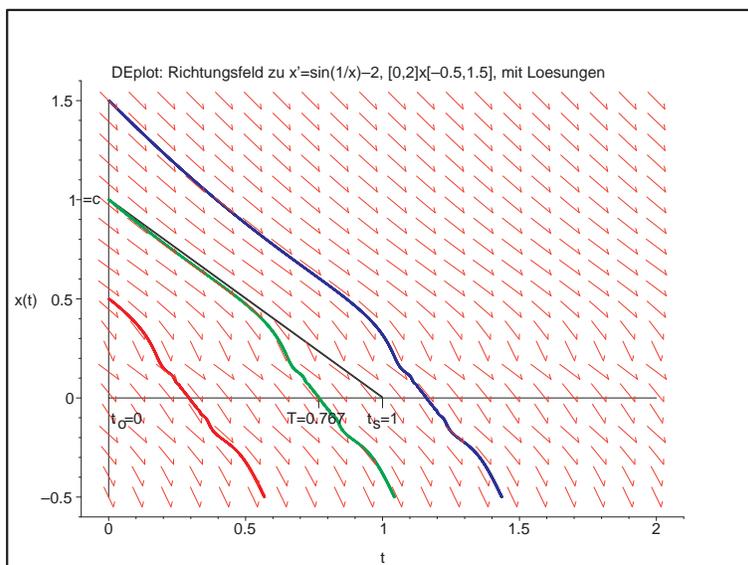


Abb. 1.11:

Datei *r_f73.ps*,

Richtungsfeld zur gDGL

$$x' = f(t, x) = \sin(1/x) - 2$$

sowie bei $t = T$ eigentlich

kollabierende Lösungen $x(t)$

mit den AB $x(0) = \frac{1}{2}, 1, \frac{3}{2}$

Dass die Trajektorien über die kritische Linie $x = 0$ hinaus berechnet werden, wie in Abb. 1.11 zu sehen ist, liegt an dem im Tool DEplot implementierten Einschrittverfahren (Standard ist das klassische Runge-Kutta-Verfahren 4. Ordnung). 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.

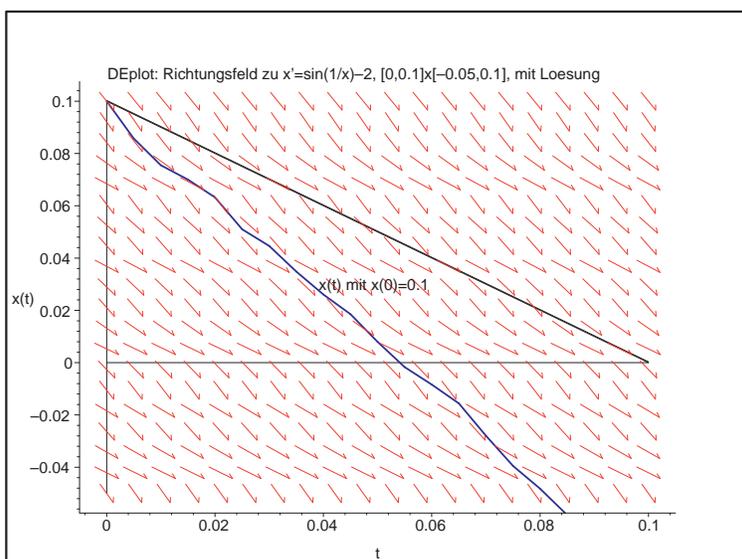


Abb. 1.12:

Datei `r_f74.ps`,

Richtungsfeld zur gDGL
 $x' = f(t, x) = \sin(1/x) - 2$
sowie Näherungslösung
mit der AB $x(0) = 0.1$,
ohne `stepsize` in DEplot

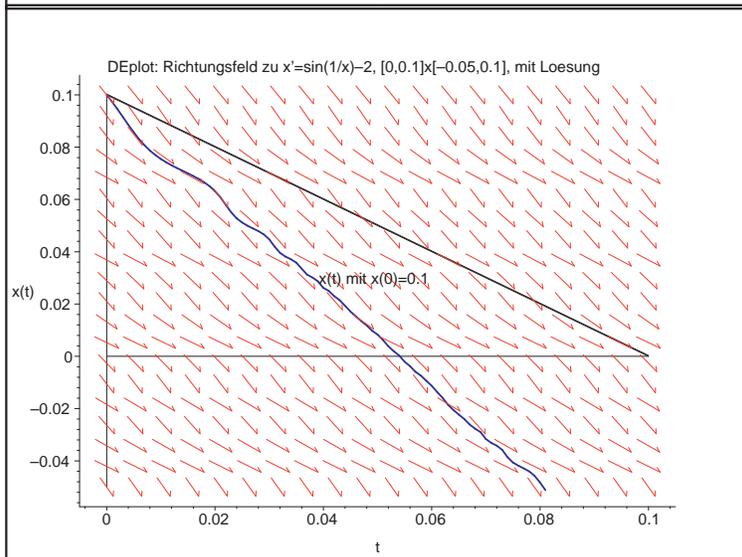


Abb. 1.13:

Datei `r_f75.ps`,

Richtungsfeld zur gDGL
 $x' = f(t, x) = \sin(1/x) - 2$
sowie Näherungslösung
mit der AB $x(0) = 0.1$,
`stepsize=0.001` in DEplot

[1.9] Übung Weisen Sie nach, dass die in (1.33) und (1.35) definierten Funktionen Lösungen der AWP (1.32) bzw. (1.34) sind. Zeigen Sie, dass die rechten Seiten der Differentialgleichungen auf $[a, b]$ nicht Lipschitz-stetig sind und damit nicht die Voraussetzungen des Satzes [1.6] erfüllen.

1.2 Das Eulersche Polygonzugverfahren (PZV)

Die Formel

$$\mathbf{x}(t) = \Phi^{t,\tau} \mathbf{x}_\tau \tag{1.38}$$

bezeichnet diejenige Lösung (Trajektorie) des SysgDGL, welche zu einem fest vorgegebenen Zeitpunkt 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

$$\mathbf{x}(t) \approx \mathbf{x}_\tau + (t - \tau) \mathbf{f}(\tau, \mathbf{x}_\tau). \tag{1.39}$$

Wir stellen die Situation im skalaren Fall dar.

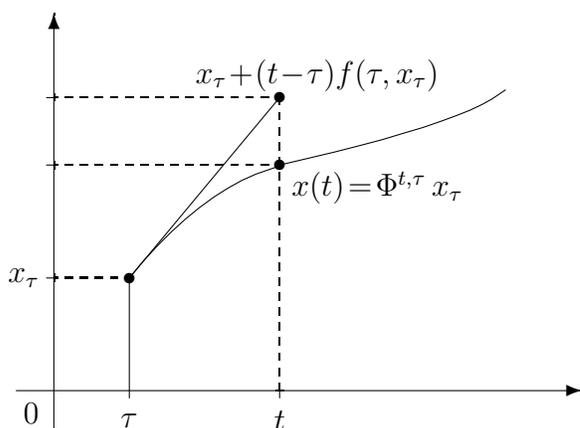


Abb. 1.14: Datei *fluss2.pic*, Approximation von $x(t)$ durch $x_\tau + (t - \tau) f(\tau, x_\tau)$

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

$$\begin{aligned} \mathbf{x}(t) &= \mathbf{x}(\tau) + (t - \tau) \mathbf{f}(\tau, \mathbf{x}(\tau)) + \frac{(t - \tau)^2}{2} \mathbf{x}''(\tau + \theta(t - \tau)) \\ &= \mathbf{x}(\tau) + (t - \tau) \mathbf{f}(\tau, \mathbf{x}(\tau)) + \mathcal{O}((t - \tau)^2) \end{aligned} \tag{1.40}$$

für ein $\theta \in (0, 1)$. Diese Ideen können wir benutzen zur Konstruktion einer ersten Näherung der Lösung des AWP (1.5), (1.17) für $t \geq t_0$. Hierzu definieren wir einen kleinen Zeitschritt $h = \Delta t$ sowie äquidistante Knoten (Stützstellen)

$$t_i = t_0 + i h, \quad i = 1, 2, 3, \dots \tag{1.41}$$

⁴Brook Taylor (1685-1731), englischer Mathematiker, arbeitete auf den Gebieten der Funktionenreihen, DGL, projektiven Geometrie

und approximieren die Lösung durch die folgende stetige, stückweise lineare Funktion.

Initialisierung: $\eta_0 = \mathbf{x}_h(t_0) := \mathbf{c}$,
Iteration, Rekursion: Ist $\eta_i = \mathbf{x}_h(t_i)$ gegeben, so definiere $\mathbf{x}_h(t)$ 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 folgende schrittweise bzw. **Diskretisierungsverfahren**.

[1.10] Algorithmus Eulersches Polygonzugverfahren

Wähle einen kleinen Zeitschritt 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

$$\eta_0 = \mathbf{c}, \tag{1.42}$$

$$\eta_{i+1} = \eta_i + h k, \quad k = k_1 = \mathbf{f}(t_i, \eta_i), \quad i = 0, 1, 2, \dots \tag{1.43}$$

Folgende Bezeichnungen für das Verfahren findet der Leser in der Literatur: Polygonzugverfahren, 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.

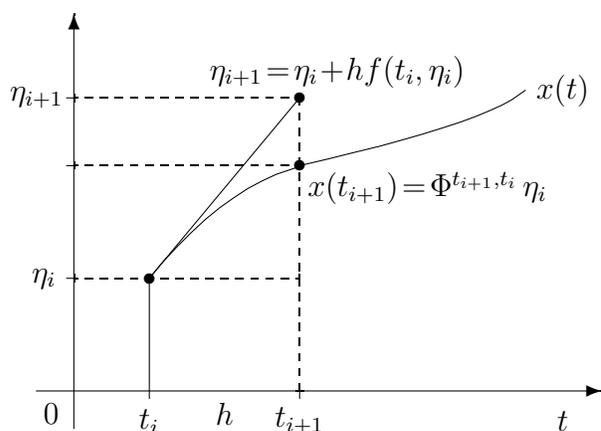


Abb. 1.15: Datei *pzv1.pic*,
 PZV $\eta_{i+1} = \eta_i + h f(t_i, \eta_i)$
 zum AWP $x'(t) = f(t, x(t))$

[1.11] Beispiele

(a) Betrachten wir als AWP die lineare homogene gDGL

$$x' = \lambda x, \quad x(0) = 1, \quad \lambda \in \mathbb{C}. \tag{1.44}$$

Die exakte Lösung des AWP ist $x(t) = e^{\lambda t}$. Sie soll mit dem PZV im Intervall $[0, T]$ approximiert werden. Hierzu wählen wir eine Zahl $N \in \mathbb{N} \setminus \{0\}$ und den Zeitschritt $h = T/N$.

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 [1.10]

$$\begin{aligned}\eta_0 &= 1, \\ \eta_{i+1} &= \eta_i + h \lambda \eta_i = (1 + h \lambda) \eta_i.\end{aligned}$$

Durch Induktion folgt

$$\eta_i = (1 + h \lambda)^i, \quad i = 0, 1, 2, \dots \quad (1.45)$$

Nun bemerkt man Folgendes.

(1) Aus der Analysis kennt man die Formel

$$\left(1 + \frac{1}{r}\right)^r \rightarrow e \quad \text{für } r \rightarrow \infty.$$

Damit gilt

$$\begin{aligned}\eta_N &= \left(1 + \lambda \frac{T}{N}\right)^N = \left(\left(1 + \frac{1}{N/(\lambda T)}\right)^{N/(\lambda T)}\right)^{\lambda T} \\ &\rightarrow e^{\lambda T} = x(T) \quad \text{für } N \rightarrow \infty.\end{aligned} \quad (1.46)$$

η_N konvergiert damit für große N gegen den korrekten Wert $x(T)$.

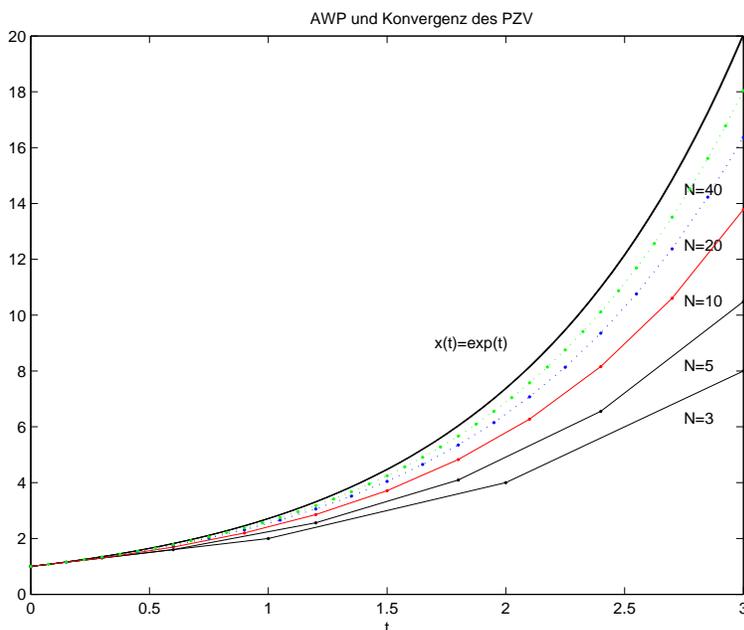


Abb. 1.16: Datei *pzv_1.ps*,
AWP $x' = \lambda x$, $x(0) = 1$,
 $0 \leq t \leq 3$, $\lambda = 1$,
mit exakter Lösung $x(t) = e^t$
und Konvergenz
des PZV $\eta_{i+1} = (1 + h \lambda) \eta_i$,
 $h = 3/N$, $N = 3, 5, 10, 20, 40$

(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 der Zeitschritt 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

$$|\eta_i| = |1 + h \lambda|^i \rightarrow \infty \quad \text{für } i \rightarrow \infty. \quad (1.47)$$

Als notwendiges Kriterium für die Konvergenz gegen 0 für $N \rightarrow \infty$ ergibt sich damit die Bedingung

$$h < \frac{2}{|\lambda|}. \tag{1.48}$$

Will man vermeiden, dass sogar $1 + h\lambda < 0$ wird, also keine Oszillationen, sondern monoton abnehmenden Iterationsverlauf hat, so erhält man die strengere Beschränkung der Zeitschrittweite $h < \frac{1}{|\lambda|}$.

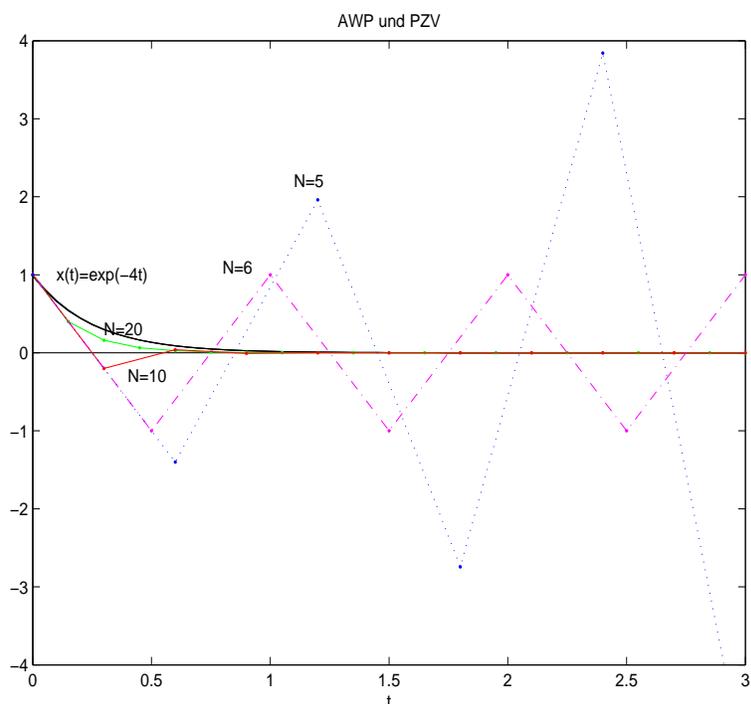


Abb. 1.17: Datei *pzv_2.ps*,
 AWP $x' = \lambda x$, $x(0) = 1$,
 $0 \leq t \leq 3$, $\lambda = -4$,
 mit exakter Lösung $x(t) = e^{-4t}$
 und PZV $\eta_{i+1} = (1 + h\lambda)\eta_i$,
 $h = 3/N$, $N = 5, 6, 10, 20$

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

$$\mathbf{x}' = \begin{pmatrix} -\alpha & \beta \\ \beta & -\alpha \end{pmatrix} \mathbf{x} = A \mathbf{x}, \quad \mathbf{x}(t_0) = \mathbf{x}(0) = \mathbf{x}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{1.49}$$

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

$$\lambda_1 = -\alpha + \beta \quad \text{und} \quad \lambda_2 = -\alpha - \beta, \tag{1.50}$$

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 Zusammenfügen der Eigenvektoren zu einer Matrix, $AV = V\Lambda$ 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 Anfangsbedingung $\mathbf{x}(0) = (1, 0)^T$ 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 (1.49) ist

$$\mathbf{x}(t) = \frac{1}{2} \left[\begin{pmatrix} e^{(-\alpha+\beta)t} + e^{(-\alpha-\beta)t} \\ e^{(-\alpha+\beta)t} - e^{(-\alpha-\beta)t} \end{pmatrix} \right]. \quad (1.51)$$

Das PZV liefert zur gewählten Schrittweite $h = \Delta t$ 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

$$\eta_i = \frac{1}{2} \begin{pmatrix} [1 + h(-\alpha + \beta)]^i + [1 + h(-\alpha - \beta)]^i \\ [1 + h(-\alpha + \beta)]^i - [1 + h(-\alpha - \beta)]^i \end{pmatrix}, \quad i = 0, 1, 2, \dots \quad (1.52)$$

Damit erklärt sich die Approximation von $\mathbf{x}(t_i)$ gemäß (1.51) durch die Größen η_i (1.52) wegen des Zusammenhangs von

$$\begin{aligned} e^{(-\alpha+\beta)t_i} &= e^{(-\alpha+\beta)ih} && \iff [1 + h(-\alpha + \beta)]^i \\ &= 1 + (-\alpha + \beta)ih && = 1 + (-\alpha + \beta)ih \\ &\quad + \frac{1}{2}(-\alpha + \beta)^2 i^2 h^2 + \mathcal{O}((ih)^3) && \quad + \frac{1}{2}(-\alpha + \beta)^2 i(i-1)h^2 + \mathcal{O}((ih)^3) \\ e^{(-\alpha-\beta)t_i} &= e^{(-\alpha-\beta)ih} && \iff [1 + h(-\alpha - \beta)]^i \\ &= 1 + (-\alpha - \beta)ih && = 1 + (-\alpha - \beta)ih \\ &\quad + \frac{1}{2}(-\alpha - \beta)^2 i^2 h^2 + \mathcal{O}((ih)^3) && \quad + \frac{1}{2}(-\alpha - \beta)^2 i(i-1)h^2 + \mathcal{O}((ih)^3) \end{aligned}$$

Man hat also vergleichbares Wachstumsverhalten.

Seien nun $\alpha > 0$ und $\beta > 0$. Wenn zusätzlich $\alpha > \beta$ ist, 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.

$$0 < h < \frac{2}{\alpha + \beta} = \frac{2}{|\lambda_2|}. \quad (1.53)$$

Will man auch Oszillation vermeiden und monotonen 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 in Teil (1).

Führt man eine Testrechnung durch für $\alpha = \beta = 1$ mit einer Schrittweite $h = 0.1$, so erhält man die in Tabelle 1.1 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. 1.1: Lösung $\eta_i = (\eta_{1i}, \eta_{2i})$ für $h = 0.1$, $\alpha = \beta = 1$

Testrechnungen für $\alpha = \beta = 20$ sind aus Tabelle 1.2 abzulesen. Es zeigt sich dass die Schrittweite $h = 0.1$ hier völlig unbrauchbar ist. Die numerische Lösung ist offenbar instabil. Wie lässt sich dies erklären?

t	num., $h=0.1$		num., $h = 0.01$		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	0.49999998969
1.0	29525	-29524	0.500000000000	0.500000000000	0.500000000000	0.500000000000

Tab. 1.2: Lösung $\eta_i = (\eta_{1i}, \eta_{2i})$ für $h = 0.1, 0.01$, $\alpha = \beta = 20$

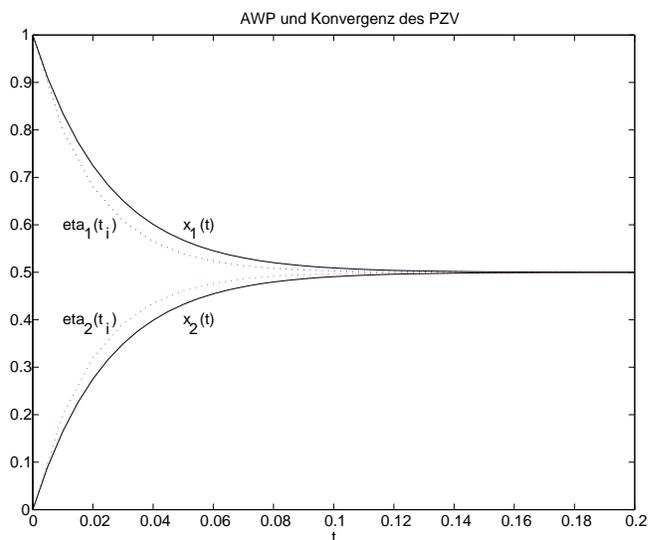


Abb. 1.18: Datei *pzv_4.ps*, AWP (1.49) mit Lösung und PZV (1.52) bei $\alpha = \beta = 20$, $h = 1/N$, $N = 100$, in $t \in [0, 0.2]$

Der Grund für unbrauchbare Näherungslösungen sind die Schrittweite h sowie die Eigenwerte $\lambda_{1,2}$ von A , die wir schon berechnet haben.

Im Fall $\alpha = \beta$ sind die Eigenwerte gleich $\lambda_1 = 0$ 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 gDGL

$$z'_k = \lambda_k z_k, \quad k = 1, 2.$$

Aus Teil (a) (vgl. (1.48)) bzw. (1.53) wissen wir, dass zur numerischen Lösung mit Hilfe des PZV eine Schrittweite gewählt werden muss, für die gilt $h < 2/|\lambda_k|$. Im Fall $\alpha = \beta = 1$ folgt nach diesem Kriterium als maximal zulässige Schrittweite $h < 1/\alpha = 1$. Im Fall $\alpha = \beta = 20$ ist dies $h < 1/20 = 0.05$.

(c) Wir kehren kurz zur gDGL $x' = \sin(1/x) - 2$ aus der Bemerkung [1.8] (e) zurück und wenden das PZV darauf an.

Zunächst werfen wir noch einen Blick auf Richtungsfelder, die erst bei feiner Auflösung und nahe $x = 0$ das "Zittern" der Linienelemente deutlich machen lassen. Diese wurden mit der MATLAB-Funktion `dfield` aus Abschnitt 1.1 erzeugt.

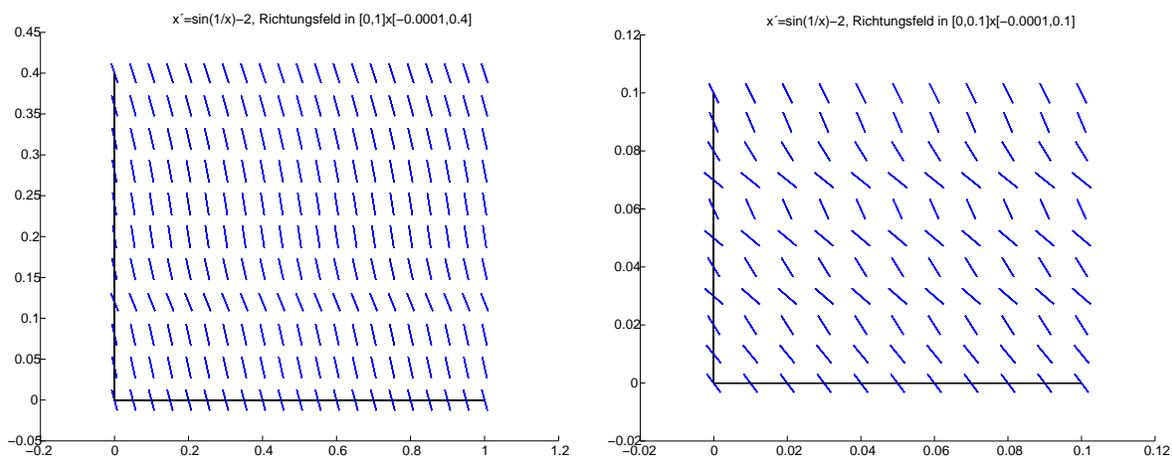


Abb. 1.19: Dateien `pzv_51.ps`, `pzv_52.ps`, Richtungsfelder zur gDGL $x' = \sin(1/x) - 2$ in den Bereichen $\Omega_1 = [0, 1] \times [-10^{-4}, 0.4]$ und $\Omega_2 = [0, 0.1] \times [-10^{-4}, 0.1]$

Nun verfolgen wir den Verlauf der Näherungslösung mit dem PZV (1.42), (1.43) 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 (1.37)

$$t = T = \int_0^c \frac{dx}{2 - \sin(1/x)} = 0.767\,410\,604,$$

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 Lösungsverläufe für verschiedene Schrittweiten h zeichnen wir vor dem Hintergrund eines groben Richtungsfelds.

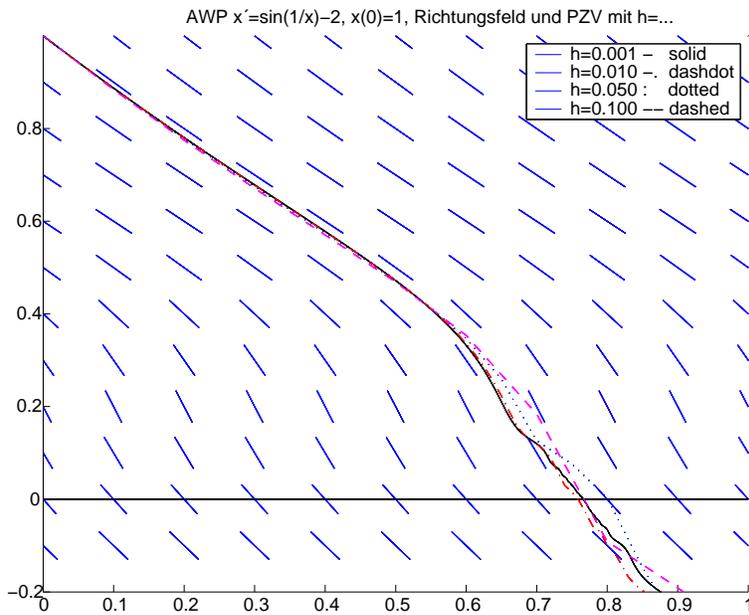


Abb. 1.20:

Datei *pzv_53.ps*,
 Richtungsfeld zur gDGL
 $x' = \sin(1/x) - 2$
 sowie Näherungslösungen
 mit PZV, AB $x(0) = 1$,
 bei $h = 0.001, 0.01, 0.05, 0.1$

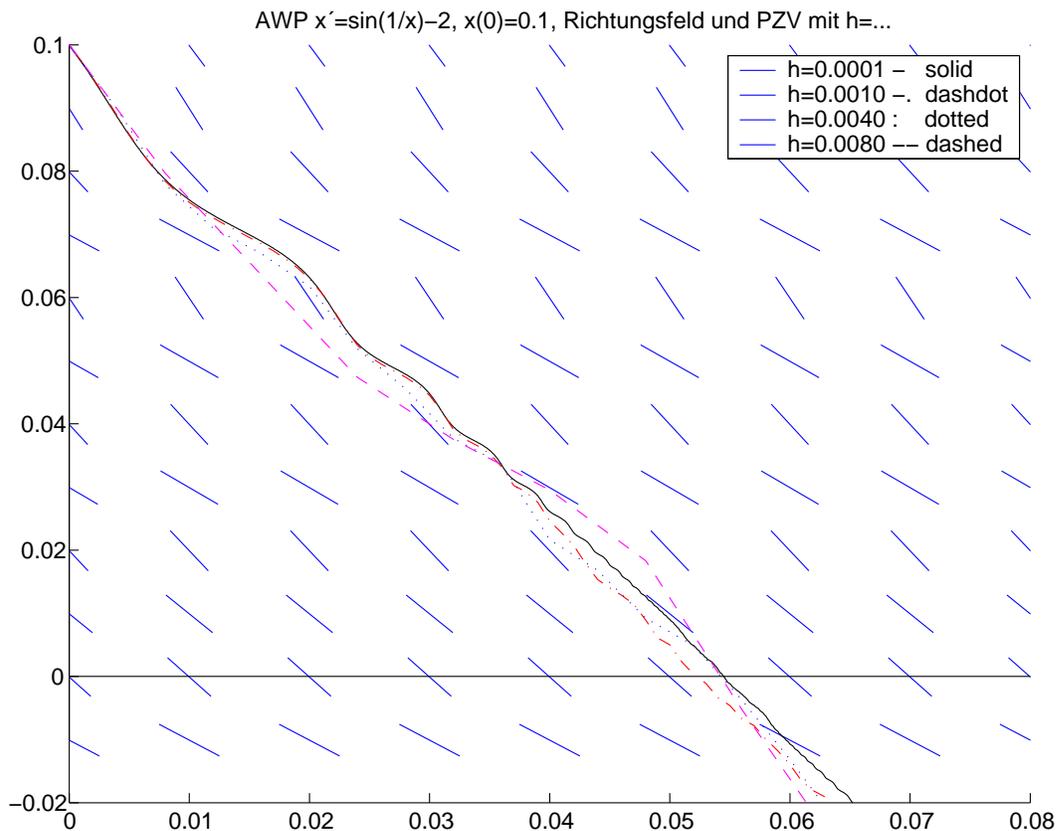


Abb. 1.21: Datei *pzv_54.ps*, Richtungsfeld zur gDGL $x' = \sin(1/x) - 2$
 sowie Näherungslösungen mit PZV, AB $x(0) = 0.1$,
 bei $h = 0.0001, 0.001, 0.004, 0.008$

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. 1.3: 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 für das AWP der RICATTI⁷-DGL

$$x' = t - x^2, \quad x(0) = 0, \quad t \in [0, 1], \tag{1.54}$$

die nicht exakt lösbar ist.

Man löse dieses 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.

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. 1.4: 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. 1.5: 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. beim Umgang mit Feldern) zu notieren.

MATLAB für den skalaren Fall

Die Implementation ist für das Beispiel [1.11] (a) $x' = \lambda x$, $x(0) = 1$, mit $\lambda = 1$, woraus die Abb. 1.16 folgt.

```
%-----
% Datei f0.m
function y = f(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 [1.11] (b), Tab. 1.2. und Abb. 1.18.

```

%-----
% Datei f3s.m
function y = fs(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('Naeherungsloesung 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 -dpasc
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 [1.11] (a),(b) sollen die folgende allgemeine Merkregel motivieren.

[1.12] Merkregel 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 ein Zeitschritt

$$h = \Delta t \ll \frac{1}{|\lambda_{max}|} \quad (1.55)$$

gewählt werden.

Lineare SysgDGl, bei denen Eigenwerte mit sehr unterschiedlichen negativen Realteilen auftreten, heißen auch **steife Differentialgleichungssysteme**. Hier erzwingt der betragsgrößte negative Realteil eine Obergrenze für die Wahl der Schrittweite.

[1.13] Übungen

- Das Beispiel [1.3] (c) soll für sehr kleine Werte von z_1 und z_2 mit Hilfe des PZV gelöst werden. Geben Sie eine Obergrenze für die Zeitschrittweite an.
- Das Beispiel [1.3] (d) soll mittels des PZV gelöst werden. Geben Sie eine obere Schranke für die Schrittweitenwahl an.
- Wie verhält sich das PZV bei großen Schrittweiten im Falle positiver Eigenwerte?

1.3 Erste Fehlerbetrachtungen

Gegeben sei das AWP

$$x' = f(t, x), \quad x(t_0) = c. \tag{1.56}$$

Die exakte Lösung kann mit Hilfe des Flusses beschrieben werden in der Form

$$x(t) = \Phi^{t,t_0} c. \tag{1.57}$$

Zu gegebener Schrittweite $h = \Delta t$ und den zugehörigen diskreten Zeiten $t_i = t_0 + i h$ liefert ein numerisches Verfahren, z. B. das Eulersche PZV, Näherungen η_i zu den gesuchten Werten $x(t_i)$. Der hierbei entstehende Approximationsfehler kann in zwei Anteile aufgeteilt werden.

[1.14] Fehlerarten

(a) Ausgehend von einem Näherungswert η_i für $\Phi^{t_i,t_0} c$ wird η_{i+1} konstruiert als Näherungswert der Trajektorie durch (t_i, η_i) zum nächsten Zeitpunkt t_{i+1} , 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

$$\tau_{i+1} = \tau(t_i, \eta_i, h) = \frac{1}{h} (\Phi^{t_i+h,t_i} \eta_i - \eta_{i+1}) \tag{1.58}$$

nennt man **lokalen Diskretisierungsfehler**. Zu seiner Abschätzung bedient man sich üblicherweise der Taylor-Entwicklung des Flusses $\Phi^{t,t_i} \eta_i$ um den Punkt (t_i, η_i) .

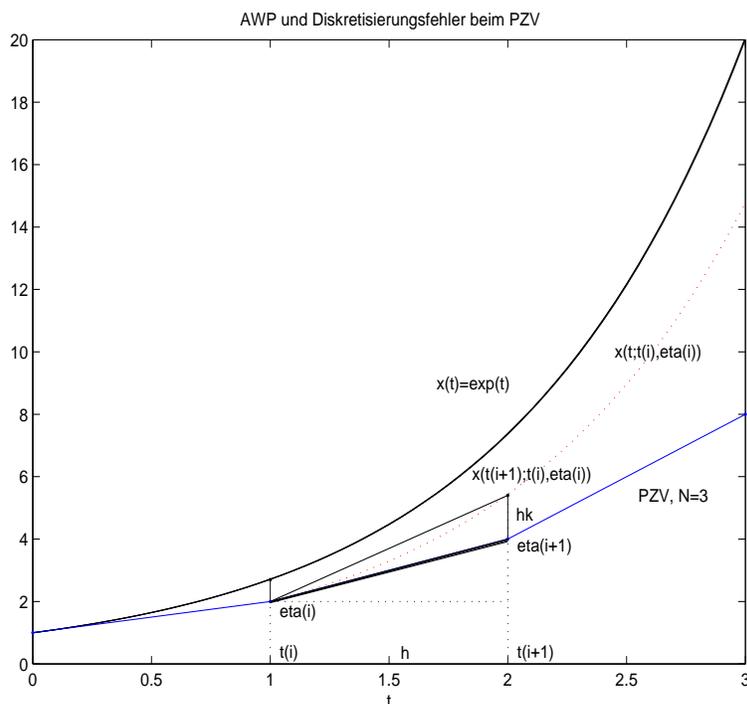


Abb. 1.22: Datei *pzv_3.ps*,
 Lokaler Diskretisierungsfehler $\tau(t_i, \eta_i, h)$
 zum PZV $\eta_{i+1} = (1 + h\lambda)\eta_i$,
 $h = 3/N$, $N = 3$,
 für das AWP $x' = \lambda x$,
 $x(0) = 1$, $0 \leq t \leq 3$, $\lambda = 1$,
 mit exakter Lösung $x(t) = e^t$

Welche geometrische Interpretation kann man dem lokalen Diskretisierungsfehler (1.57) $\tau(t_i, \eta_i, h)$ geben?

Gemäß Abb. 1.22 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

$$\begin{aligned} \tau_{i+1} = \tau(t_i, \eta_i, h) &= \frac{1}{h} (\Phi^{t_{i+1}, t_i} \eta_i - \eta_{i+1}) = \frac{1}{h} (\Phi^{t_{i+1}, t_i} \eta_i - \eta_i + \eta_i - \eta_{i+1}) \\ &= \frac{1}{h} (x(t_{i+1}; t_i, \eta_i) - \eta_i) - \frac{1}{h} (\eta_{i+1} - \eta_i). \end{aligned} \quad (1.59)$$

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, so dass 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 Zeitschritt neu entstehenden Anteile zum **globalen Diskretisierungsfehler** an der Stelle t_i

$$e_i = e(t_i, h) = \eta_i - x(t_i; t_0, x_0) = \eta_i - \Phi^{t_i, t_0} x_0 \quad (1.60)$$

auf, so dass unter Umständen die Teiltrajektorien von der gesuchten Lösungstrajektorie immer weiter "wegdriften". Diesen globalen Fehler versucht man mit so genannten 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 bez. t und x .

Ist (t_i, η_i) gegeben, so bezeichne

$$z(t) = \Phi^{t, t_i} \eta_i \quad (1.61)$$

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

$$\begin{aligned} z''(t_i) &= \left. \frac{d}{dt} z'(t) \right|_{t=t_i} = \left. \frac{d}{dt} f(t, z(t)) \right|_{t=t_i} \\ &= f_t(t_i, z(t_i)) + z'(t_i) f_x(t_i, z(t_i)) = f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i). \end{aligned}$$

Hierbei bezeichnen f_t und f_x die partiellen Ableitungen von f bez. der ersten und zweiten Variable. Aus

$$z(t_{i+1}) = \eta_i + h f(t_i, \eta_i) + \frac{h^2}{2} [f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i)] + \mathcal{O}(h^3) \quad (1.62)$$

und

$$\eta_{i+1} = \eta_i + h f(t_i, \eta_i) \quad (1.63)$$

ergibt sich hieraus der lokale Diskretisierungsfehler

$$\begin{aligned} \tau_{i+1} = \tau(t_i, \eta_i, h) &= \frac{1}{h}(z(t_{i+1}) - \eta_{i+1}) \\ &= \frac{h}{2}[f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i)] + \mathcal{O}(h^2) \\ &\doteq \frac{h}{2}[f_t(t_i, \eta_i) + f(t_i, \eta_i) f_x(t_i, \eta_i)] \\ &= \mathcal{O}(h). \end{aligned} \quad (1.64)$$

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_i den exakten Lösungswert $x(t_i)$ einsetzt. Das bedeutet

$$\begin{aligned} \eta_{i+1} &= \eta_i + h f(t_i, \eta_i), \\ 0 &= \frac{1}{h}(\eta_{i+1} - \eta_i) - f(t_i, \eta_i), \\ \tau_{i+1} &= \frac{1}{h}[x(t_{i+1}) - x(t_i)] - f(t_i, x(t_i)) \\ &= \frac{1}{h}[x(t_i) + hx'(t_i) + \frac{h^2}{2}x''(t_i) + \mathcal{O}(h^3) - x(t_i)] - f(t_i, x(t_i)) \\ &= x'(t_i) - f(t_i, x(t_i)) + \frac{h}{2}x''(t_i) + \mathcal{O}(h^2) \\ &= \frac{h}{2}x''(t_i) + \mathcal{O}(h^2) \\ &= \mathcal{O}(h). \end{aligned}$$

[1.15] Beispiel

Im Fall der gDGL $x'(t) = tx(t)$ ist $f(t, x) = tx$, $f_t(t, x) = x$ und $f_x(t, x) = t$.

Der lokale Diskretisierungsfehler ist daher gleich

$$\tau(t_i, \eta_i, h) \doteq \frac{h}{2}(t_i + t_i^2 \eta_i) = \frac{h}{2} t_i(1 + t_i \eta_i).$$

2 Einschrittverfahren

2.1 Das Konzept der Einschrittverfahren (ESV)

Der lokale Diskretisierungsfehler des PZV ist von der Ordnung $\mathcal{O}(h)$, $h = \Delta t$. Hierbei werden Näherungslösungen von gDGL konstruiert, indem in jedem kleinen Teilintervall eine lineare Approximation konstruiert wird, wobei als Steigung die *Steigung am linken Randpunkt* gewählt wird. Das Ziel dieses Kapitels ist es, Alternativen zum PZV zu entwickeln, welche günstiger sind bez. des lokalen Diskretisierungsfehlers. Die folgenden Beispiele versuchen, einen Wert für die Steigung zu finden, welcher dem Verlauf des gesamten Teilintervalls besser angepasst ist. Wie vorher sei $t_i = t_0 + ih$ mit einem vorgegebenen kleinen Zeitschritt h . Zur Berechnung des lokalen Diskretisierungsfehlers benötigen wir die Taylor-Entwicklung von $f(\cdot, \cdot)$ um den Punkt (t_i, η_i) .

Für $x(t) - \eta_i = (t - t_i)x'(\xi) = \mathcal{O}(h)$ und $\mathcal{O}(t - t_i) = \mathcal{O}(h)$ ist diese gegeben durch

$$\begin{aligned} f(t, x) &= f(t_i, \eta_i) + (t - t_i) f_t(t_i, \eta_i) + (x - \eta_i) f_x(t_i, \eta_i) + \frac{1}{2} (t - t_i)^2 f_{tt}(t_i, \eta_i) \\ &\quad + (t - t_i)(x - \eta_i) f_{tx}(t_i, \eta_i) + \frac{1}{2} (x - \eta_i)^2 f_{xx}(t_i, \eta_i) + \mathcal{O}(h^3) \\ &= f(t_i, \eta_i) + (t - t_i) f_t(t_i, \eta_i) + (x - \eta_i) f_x(t_i, \eta_i) + \mathcal{O}(h^2). \end{aligned} \tag{2.1}$$

[2.1] Beispiele

(a) Modifiziertes Polygonzug-Verfahren (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?

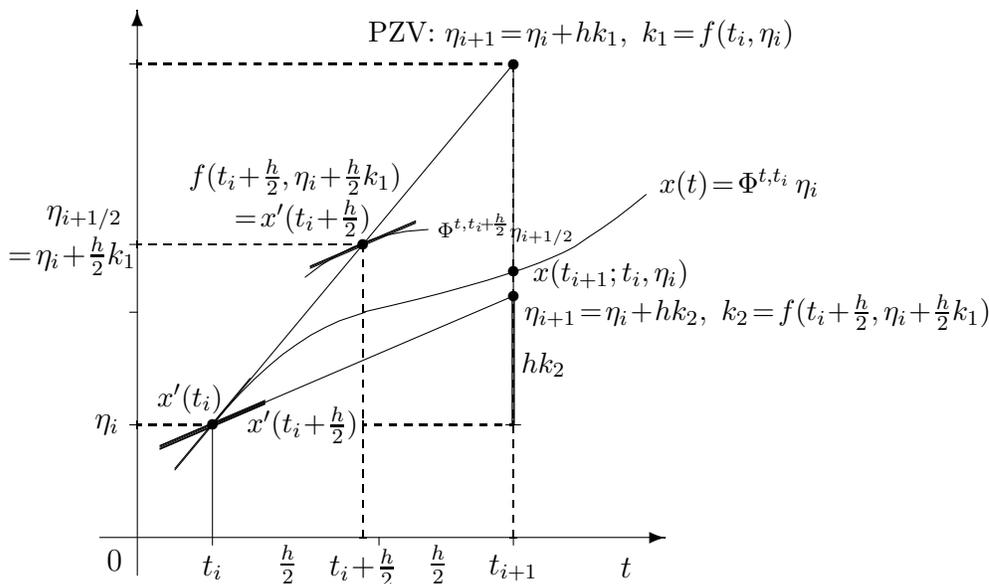


Abb. 2.1: Datei *mpzv1.pic*, Wahl des Anstiegs von einer Trajektorie “in der Mitte des Intervalls“ und Iterationsschritt beim MPZV

Der Iterationsschritt ist damit gegeben durch die Vorschrift

$$\eta_{i+1} = \eta_i + hf(t_i + \frac{h}{2}, \eta_i + \frac{h}{2}f(t_i, \eta_i)) \quad (2.2)$$

oder in der Form mit den Anstiegen (Steigungswerte) k_1 , k_2 bzw. Zuwächsen (Korrekturen) hk_1 , hk_2

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f(t_i + \frac{h}{2}, \eta_i + \frac{h}{2}k_1), \\ \eta_{i+1} &= \eta_i + hk, \quad k = k_2. \end{aligned} \quad (2.3)$$

Aus (2.2) und der Taylor-Entwicklung (2.1) folgt

$$\begin{aligned} \eta_{i+1} &= \eta_i + h[f(t_i, \eta_i) + \frac{h}{2}f_t(t_i, \eta_i) + \frac{h}{2}f(t_i, \eta_i)f_x(t_i, \eta_i) + \mathcal{O}(h^2)] \\ &= \eta_i + hf(t_i, \eta_i) + \frac{h^2}{2}[f_t(t_i, \eta_i) + f(t_i, \eta_i)f_x(t_i, \eta_i)] + \mathcal{O}(h^3). \end{aligned} \quad (2.4)$$

Ein Vergleich mit der Taylor-Entwicklung (1.62) für $z(t) = \Phi^{t,t_i}\eta_i$ ergibt als Ordnung für den lokalen Diskretisierungsfehler

$$\tau_{i+1} = \tau(t_i, \eta_i, h) = \frac{1}{h}(z(t_{i+1}) - \eta_{i+1}) = \mathcal{O}(h^2). \quad (2.5)$$

Der lokale Diskretisierungsfehler ist damit um eine Ordnung besser als im Fall des PZV. Allerdings sind pro Zeitschritt 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. $\mathbf{x}(t_0) = \mathbf{x}_0$ 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. HEUN⁸ (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.

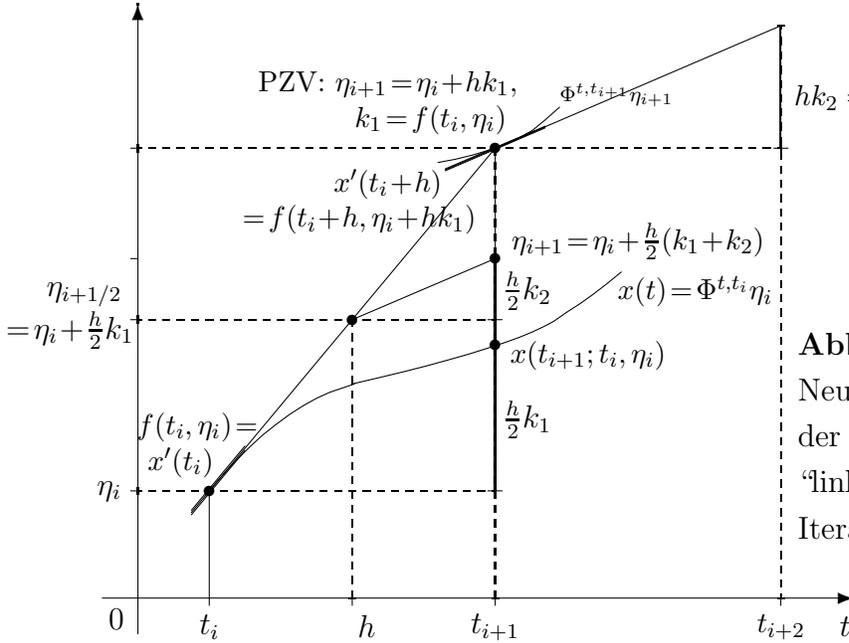


Abb. 2.2: Datei *heun1.pic*, Neuer Anstieg als Mittelwert der Anstiege der Trajektorien “links“ und “rechts“ sowie Iterationsschritt beim HV

Die Iterationsvorschrift lautet also

$$\eta_{i+1} = \eta_i + \frac{h}{2}[f(t_i, \eta_i) + f(t_i + h, \eta_i + hf(t_i, \eta_i))] \quad (2.6)$$

bzw. in der Form mit den Anstiegen k_1, k_2

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f(t_i + h, \eta_i + hk_1), \\ \eta_{i+1} &= \eta_i + hk, \quad k = \frac{1}{2}(k_1 + k_2). \end{aligned} \quad (2.7)$$

Notiert man das PZV als $\bar{\eta}_{i+1} = \eta_i + hk_1$, so ergibt sich für das HV die Formel

$$\eta_{i+1} = \eta_i + hk_1 + \frac{h}{2}(k_2 - k_1) = \bar{\eta}_{i+1} + \frac{h}{2}(k_2 - k_1),$$

so dass die Verbesserung (Korrekturterm) $\frac{h}{2}(k_2 - k_1)$ zum PZV sichtbar wird.

Die Taylor-Entwicklung (2.1) liefert für (2.7)

$$\begin{aligned} \eta_{i+1} &= \eta_i + \frac{h}{2}f(t_i, \eta_i) + \frac{h}{2}[f(t_i, \eta_i) + hf_t(t_i, \eta_i) + hf(t_i, \eta_i)f_x(t_i, \eta_i) + \mathcal{O}(h^2)] \\ &= \eta_i + hf(t_i, \eta_i) + \frac{h^2}{2}[f_t(t_i, \eta_i) + f(t_i, \eta_i)f_x(t_i, \eta_i)] + \mathcal{O}(h^3). \end{aligned}$$

Wie im Teil (a) ergibt sich damit ein lokaler Diskretisierungsfehler der Ordnung $\mathcal{O}(h^2)$.

⁸Karl Ludwig Wilhelm Max Heun (1859-1929), deutscher Mathematiker, arbeitete an mechanischen Problemen wie d’Alembertsches Prinzip, Eulersche Gleichungen, Lagrange-Formalismus

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 (2.3) ist analog die vektorielle Version zu notieren.

Eine allgemeine Klasse von Approximationsverfahren für AWP lässt sich wie folgt formulieren.

[2.2] Definition

(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

$$\eta_{i+1} = \eta_i + h \Psi(t_i, \eta_i, h), \quad i = 0, 1, \dots, \quad (2.8)$$

mit einer vorgegebenen, i. Allg. von $f(\cdot, \cdot)$ bzw. von $\mathbf{f}(\cdot, \cdot)$ abhängigen Funktion

$$\Psi = \Psi(t_i, \eta_i, h) = \Psi(t_i, \eta_i; h, \mathbf{f}), \quad (2.9)$$

die auch **Verfahrens-** oder **Inkrementfunktion** genannt wird.

(b) Der **lokale Diskretisierungsfehler** (Approximationsfehler) für das ESV ist definiert durch

$$\tau(t_i, \eta_i, h) = \frac{1}{h} (\Phi^{t_i+h, t_i} \eta_i - \eta_{i+1}) = \frac{1}{h} (\Phi^{t_i+h, t_i} \eta_i - \eta_i) - \Psi(t_i, \eta_i, h) \quad (2.10)$$

(man vergl. Definition (1.58) und Formel (1.59)).

(c) Das ESV heißt **konsistent** (verträglich), wenn gilt

$$\lim_{h \rightarrow 0} \tau(t_i, \eta_i, h) = 0. \quad (2.11)$$

Dies ist nach Gleichung (2.8) und der Definition von $\Phi^{t_i+h, t_i} \eta_i$ genau dann der Fall, wenn gilt

$$\lim_{h \rightarrow 0} \Psi(t_i, \eta_i, h) = f(t_i, \eta_i). \quad (2.12)$$

(d) Das ESV ist ein **Verfahren der Ordnung** $p > 0$, falls

$$\tau(t_i, \eta_i, h) = \mathcal{O}(h^p) \quad \text{bzw.} \quad |\tau(t_i, \eta_i, h)| \leq C h^p, \quad C \geq 0. \quad (2.13)$$

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

$$\Psi(t_i, \eta_i, h) = b_1 f(t_i, \eta_i) + b_2 f(t_i + ch, \eta_i + ah f(t_i, \eta_i)) \quad (2.14)$$

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 $\sum b_i = 1$ und hier auch $a = c$ gelten.

Damit hat bei Einbeziehung der Zuwächse $hk_1 = hf(t_i, \eta_i)$ und $hk_2 = hf(t_i + ch, \eta_i + ahk_1)$ die Verfahrensfunktion 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

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^T \end{array} \quad \text{bzw.} \quad \begin{array}{c|cc} c_1 = 0 & & \\ \hline c_2 & a_{21} & \\ \hline 1 & b_1 & b_2 \end{array}$$

Geeignete Koeffizienten ergeben sich aus folgendem Satz.

[2.3] Satz Ein ESV mit dem Ansatz (2.14) ist genau dann von (mindestens) der Konsistenzordnung 2, wenn

$$b_1 + b_2 = 1, \quad b_2 a = \frac{1}{2} \quad \text{und} \quad b_2 c = \frac{1}{2}. \quad (2.15)$$

Eine höhere als die Ordnung 2 ist mit diesem Ansatz i. Allg. nicht zu erreichen.

Beweis: Mit dem Ansatz (2.14) hat Ψ um (t_i, η_i) die Taylor-Entwicklung

$$\Psi(t_i, \eta_i, h) = (b_1 + b_2)f(t_i, \eta_i) + b_2 h [cf_t(t_i, \eta_i) + af(t_i, \eta_i)f_x(t_i, \eta_i)] + \mathcal{O}(h^2). \quad (2.16)$$

Die Bedingungen (2.15) folgen durch Einsetzen dieser Entwicklung in das ESV (2.8) und durch Vergleich mit der Taylor-Entwicklung (1.62) für $z(t_{i+1}) = \Phi^{t_{i+1}, t_i} \eta_i$, also aus

$$\begin{aligned} \tau(t_i, \eta_i, h) &= \frac{1}{h}(\Phi^{t_i+h, t_i} \eta_i - \eta_{i+1}) \\ &= \frac{1}{h}(z(t_{i+1}) - \eta_{i+1}) \\ &= \frac{1}{h}\{\eta_i + hf(t_i, \eta_i) + \frac{h^2}{2}[f_t(t_i, \eta_i) + f(t_i, \eta_i)f_x(t_i, \eta_i)] + \mathcal{O}(h^3) \\ &\quad - \eta_i - h[(b_1 + b_2)f(t_i, \eta_i) + b_2 h(cf_t(t_i, \eta_i) + af(t_i, \eta_i)f_x(t_i, \eta_i)) + \mathcal{O}(h^2)]\} \\ &= (1 - b_1 - b_2)f(t_i, \eta_i) + h(\frac{1}{2} - b_2 c)f_t(t_i, \eta_i) + h(\frac{1}{2} - b_2 a)f(t_i, \eta_i)f_x(t_i, \eta_i) + \mathcal{O}(h^2). \square \end{aligned}$$

2.2 Runge-Kutta-Verfahren (RKV)

Ein Iterationsschritt für ein explizites ESV mit dem Ansatz (2.14) kann in die folgenden Rechenschritte zerlegt werden.

[2.4] **Algorithmus** Iterationsschritt für ein explizites ESV mit Ansatz (2.14)

Gegeben sei (t_i, η_i) .

S1 Berechne $k_1 = f(t_i, \eta_i)$.

S2 Berechne $k_2 = f(t_i + ch, \eta_i + ahk_1)$.

S3 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 Berechne (t_{i+1}, η_{i+1}) mit $t_{i+1} = t_i + h$ und
 $\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h)$ bzw. $\eta_{i+1} = \eta_i + hk$

Verfahren dieser Art werden wegen der Berechnung von zwei Anstiegen k_i als **2-stufige Runge⁹-Kutta¹⁰-Verfahren** bezeichnet.

Dies stellt den Spezialfall eines **s-stufigen RKV** dar. Der Ansatz (2.14) soll nun verallgemeinert und analysiert werden, um eine höhere Ordnung zu erzielen.

[2.5] **Definition**

Ein **3-stufiges RKV** ist ein explizites ESV mit dem Ansatz

$$\begin{aligned} k_1 &= f(t_i + c_1 h, \eta_i), \\ k_2 &= f(t_i + c_2 h, \eta_i + h a_{21} k_1), \\ k_3 &= f(t_i + c_3 h, \eta_i + h (a_{31} k_1 + a_{32} k_2)), \\ \Psi(t_i, \eta_i, h) &= k = b_1 k_1 + b_2 k_2 + b_3 k_3, \end{aligned} \quad (2.17)$$

mit geeigneten Koeffizienten a_{ij} , b_i , c_i .

[2.6] **Beispiel** 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

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^T \end{array} = \begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ 1 & \frac{1}{3} & \frac{2}{3} \\ \hline 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{array} \quad (2.18)$$

⁹Carle David Tolmé Runge (1856-1927), deutscher Mathematiker, Arbeitsgebiete waren Differentialgeometrie, Lösung von algebraischen und gDGL

¹⁰Martin Wilhelm Kutta (1867-1944), deutscher Mathematiker, arbeitete auf dem Gebiet der Lösung von gDGL

Damit ergeben sich die Anstiege

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hf(t_i, \eta_i)), \\ k_3 &= f(t_i + h, \eta_i + h(\frac{1}{3}f(t_i, \eta_i) + \frac{2}{3}f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hf(t_i, \eta_i)))), \\ \Psi(t_i, \eta_i, h) &= \frac{1}{3}(k_1 + k_2 + k_3). \end{aligned}$$

bzw. in Kurzform

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hk_1), \\ k_3 &= f(t_i + h, \eta_i + h(\frac{1}{3}k_1 + \frac{2}{3}k_2)), \\ \Psi(t_i, \eta_i, h) &= k = \frac{1}{3}(k_1 + k_2 + k_3). \end{aligned} \tag{2.19}$$

Wir wenden dieses Verfahren an zur Lösung der autonomen gDGL

$$x' = f(x).$$

In diesem Fall lauten die Gleichungen zur Berechnung der k_i

$$\begin{aligned} k_1 &= f(\eta_i), \\ k_2 &= f(\eta_i + \frac{1}{2}hf(\eta_i)), \\ k_3 &= f(\eta_i + h(\frac{1}{3}f(\eta_i) + \frac{2}{3}f(\eta_i + \frac{1}{2}hf(\eta_i)))). \end{aligned}$$

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 $z(t) = \Phi^{t, t_i} \eta_i$ (vgl. (1.62))

$$z(t_{i+1}) = \eta_i + hf_i + \frac{1}{2}h^2 f_i f'_i + \mathcal{O}(h^3),$$

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. ◀

[2.7] **Übung** Zeigen Sie, dass für die autonome gDGL ein 3-stufiges RKV mindestens die Ordnung 2 hat, wenn gilt

$$b_1 + b_2 + b_3 = 1, \quad (2.20)$$

$$b_2 a_{21} + b_3 (a_{31} + a_{32}) = \frac{1}{2}. \quad (2.21)$$

Ähnliche Abschätzungen wie in Beispiel [2.6] können auch für nichtautonome gDGL durchgeführt werden, auch für höherstufige RKV-Verfahren, welche wie folgt definiert sind.

[2.8] **Definition** Ein s -stufiges explizites RKV für die gDGL $x' = f(t, x)$ ist ein ESV der Form (2.8) $\eta_{l+1} = \eta_l + h \Psi(t_l, \eta_l, h)$ mit

$$\Psi(t_l, \eta_l, h) = \sum_{i=1}^s b_i k_i \quad (2.22)$$

(Achtung: Umbezeichnung der Indizes) und

$$k_i = f\left(t_l + c_i h, \eta_l + h \sum_{j=1}^{i-1} a_{ij} k_j\right), \quad i = 1, 2, \dots, s. \quad (2.23)$$

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 rkvl.m

% Explizite Runge-Kutta-Verfahren, RKV fuer gDGL

```
function [t,x] = rkvl(A,b,c,f,t0,t1,x0,N)
    h = (t1-t0)/N;
    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(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;
```

Die Anweisungen der l -Schleife in `rkvl` 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.

```
% Datei rkv1a.m

function [t,x] = rkv1a(A,b,c,f,t0,t1,x0,N)
    h = (t1-t0)/N;
    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(l)+h*c(i),x(l)+h*k*A(i,:))';
        end;
        x(l+1) = x(l)+h*k*b;
    end;
```

Es soll noch auf eine Modifikation von rkv1 hingewiesen werden, die jedoch zu Fehlern führt. Das Problem, welches hier in der i -Schleife auftritt, ist entsprechend kommentiert.

```
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, RKV fuer SysgDGL

function [t,x] = rkv(A,b,c,f,t0,t1,x0,N)
    h = (t1-t0)/N;
    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 [2.6] 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] = rkvl(A,b,c,'f8',t0,t1,x0,N);
% Auswertung der Naehungsloesung
```

Analog kan man alle bisherigen ESV auf diese Weise notieren und lösen.

Durch die Taylor-Entwicklung lassen sich leicht Bedingungen für die Konsistenzordnung s -stufiger RKV angeben. Wir fassen zusammen.

[2.9] Satz Zwischen den Koeffizienten c_i und a_{ij} gelte zusätzlich die Beziehung¹¹

$$c_i = \sum_{j=1}^s a_{ij}. \quad (2.24)$$

Dann besitzt das s -stufige RKV genau dann (mindestens) die Konsistenzordnung

1, wenn

$$\mathbf{b}^T \mathbf{1} = \sum_{i=1}^s b_i = 1, \quad (2.25)$$

2, wenn zusätzlich gilt

$$\mathbf{b}^T \mathbf{c} = \sum_{i=1}^s b_i c_i = \frac{1}{2}, \quad (2.26)$$

3, wenn zusätzlich gilt

$$\sum_{i=1}^s b_i c_i^2 = \frac{1}{3}, \quad \mathbf{b}^T A \mathbf{c} = \sum_{i,j=1}^s b_i a_{ij} c_j = \frac{1}{6}, \quad (2.27)$$

4, wenn zusätzlich gilt

$$\sum_{i=1}^s b_i c_i^3 = \frac{1}{4}, \quad \sum_{i,j=1}^s b_i c_i a_{ij} c_j = \frac{1}{8}, \quad \sum_{i,j=1}^s b_i a_{ij} c_j^2 = \frac{1}{12}, \quad \mathbf{b}^T A^2 \mathbf{c} = \sum_{i,j,k=1}^s b_i a_{ij} a_{jk} c_k = \frac{1}{24}. \quad (2.28)$$

Die im Satz angegebenen Bedingungsgleichungen sind nichtlinear und deshalb stellt die Bestimmung geeigneter Koeffizienten, die sogenannte *Verfahrenskonstruktion*, eine schwierige Aufgabe dar.

¹¹Eine Begründung für diese Bedingung findet man z. B. in Abschnitt 4.2.1 von P. Deuffhard/ F. Bornemann, Numerische Mathematik II, de Gruyter, 1994.

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 $b_1, b_2, b_3, b_4, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}$, zu bestimmen, so dass sämtliche Bedingungen (2.25) bis (2.28) 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.

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

Dies besitzt eigentlich nur 3 Knoten. Deswegen wird der mittlere verdoppelt, so dass

$$\mathbf{c} = (0, \frac{1}{2}, \frac{1}{2}, 1)^T, \quad \mathbf{b} = (\frac{1}{6}, \frac{2}{6}, \frac{2}{6}, \frac{1}{6})^T.$$

Ein analoges Vorgehen wie unter (a) führt auf das **klassisches RKV (KRKV)** (1895).

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
Mindeststufen s	1	2	3	4	6	7	9
#Bedingungsgleichungen	1	2	4	8	17	37	85
#Koeffizienten	1	3	6	10	21	28	45

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
maximale Ordnung p	1	2	3	4	4	5	6	6

[2.10] **Beispiel** Das KRKV ist 4-stufig, d. h. $s = 4$, und gegeben durch

$$\mathbf{b} = \frac{1}{6} \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (2.29)$$

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. BUTCHER¹² eingeführten Parameterschema übersichtlich zusammengefasst.

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^T \\ \hline 1 & \frac{1}{6} \quad \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{6} \end{array} = \begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 1 & 0 & 0 & 1 \\ \hline 1 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Die explizite Formulierung des KRKV (2.8) lautet

$$\begin{aligned} k_1 &= f(t_i, \eta_i), \\ k_2 &= f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hk_1), \\ k_3 &= f(t_i + \frac{1}{2}h, \eta_i + \frac{1}{2}hk_2), \\ k_4 &= f(t_i + h, \eta_i + hk_3), \\ \Psi(t_i, \eta_i, h) &= k = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned} \quad (2.30)$$

Man überzeugt sich leicht, dass die Gleichungen (2.25), ..., (2.28) erfüllt sind. Es handelt sich also um ein Verfahren 4. Ordnung. ◀

```
% Datei krkv1.m
% Explizites Klassisches Runge-Kutta-Verfahren, KRKV fuer gDGL
```

```
function [t,x] = krkv1(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);
        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;
```

¹²John C. Butcher (1933-...), neuseeländischer Mathematiker, Arbeitsgebiete Numerik von DGL, Wissenschaftliches Rechnen

Genauso kann man die vektorielle Version definieren.

Die Einbeziehung des Parameterschemas ist nicht angebracht, weil die Matrix A nur drei Nichtnulleinträge enthält.

Die geometrische Illustration des KRKV mit den vier Anstiegen ist wie folgt.

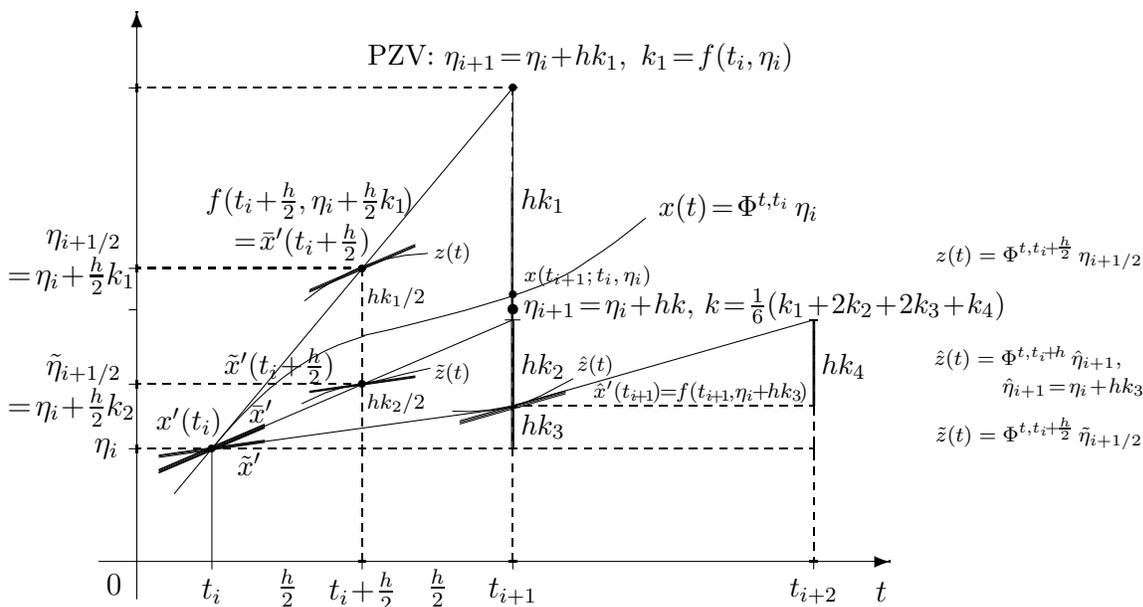


Abb. 2.3: Datei *krkv.pic*, Wahl des Anstiegs als Mittelwert der Anstiege von vier Trajektorien "links", "Mitte" und "rechts" sowie Schritt beim KRKV

[2.11] Übung Zeigen Sie Folgendes.

- (a) Das 3-stufige Verfahren aus Beispiel [2.6] 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.

2.2.1 Parameterschemata für explizite ESV

In einer Übersicht stellen wir ausgewählte Parameterschemata bzw. Butcher-Tableau 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

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^T \end{array} = \begin{array}{c|ccc} c_1 = 0 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ c_4 & a_{41} & a_{42} & a_{43} \\ \dots & \dots & \dots & \dots \\ c_s & a_{s1} & a_{s2} & a_{s3} & \dots & a_{s,s-1} \\ \hline 1 & b_1 & b_2 & b_3 & \dots & b_{s-1} & b_s \end{array} \quad (2.31)$$

Dabei lässt man den Koeffizienten $c_1 = 0$ oftmals weg.

(a) (1,1)-RKV, PZV (b) (2,2)-RKV, MPZV (c) (2,2)-RKV, HV (d) (2,2)-RKV

$c_1 = 0$	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$ $\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$
			0 1		$\frac{1}{2}$ $\frac{1}{2}$		$\frac{1}{4}$ $\frac{3}{4}$

(e) (3,2)-RKV (f) (3,2)-RKV (g) (3,3)-RKV (h) (3,3)-KUTTA-V.

$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
1	$\frac{1}{3}$ $\frac{2}{3}$	$\frac{27}{40}$	$-\frac{189}{800}$ $\frac{729}{800}$	$\frac{3}{4}$	0 $\frac{3}{4}$	1	-1 2
	$\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{3}$		$\frac{214}{891}$ $\frac{1}{33}$ $\frac{650}{891}$		$\frac{2}{9}$ $\frac{3}{9}$ $\frac{4}{9}$		$\frac{1}{6}$ $\frac{4}{6}$ $\frac{1}{6}$

(i) (4,4)-RKV, KRKV (j) (4,4)-RKV, 3/8-Formel (k) (4,4)-RKV, ENGLAND-Formel

$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	0 $\frac{1}{2}$	$\frac{2}{3}$	$-\frac{1}{3}$ 1	$\frac{1}{2}$	$\frac{1}{4}$ $\frac{1}{4}$
1	0 0 1	1	1 -1 1	1	0 -1 2
	$\frac{1}{6}$ $\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{6}$		$\frac{1}{8}$ $\frac{3}{8}$ $\frac{3}{8}$ $\frac{1}{8}$		$\frac{1}{6}$ 0 $\frac{2}{3}$ $\frac{1}{6}$

(l) (4,4)-RKV, KUNTZMANN-Formel (m) (4,4)-RKV, GILL-Modifikation, $c=1/\sqrt{2}$

$\frac{2}{5}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{3}{5}$	$-\frac{3}{20}$ $\frac{3}{4}$	$\frac{1}{2}$	$c - \frac{1}{2}$ $1 - c$
1	$\frac{19}{44}$ $-\frac{15}{44}$ $\frac{40}{44}$	1	0 $-c$ $1 + c$
	$\frac{55}{360}$ $\frac{125}{360}$ $\frac{125}{360}$ $\frac{55}{360}$		$\frac{1}{6}$ $\frac{1-c}{3}$ $\frac{1+c}{3}$ $\frac{1}{6}$

(n) (4,4)-RKV mit minimalem Diskretisierungsfehler

$$\alpha_2 = \frac{2}{5}, \quad \alpha_3 = \frac{7}{8} - \frac{3}{16}\sqrt{5} = 0.455737254218789,$$

$$\beta = \frac{1-\alpha_2}{(\alpha_3-\alpha_2)(6\alpha_2\alpha_3-4(\alpha_2+\alpha_3)+3)} = 16.047205832726471$$

$\frac{2}{5}$	$\frac{2}{5}$	$\alpha_3 - \frac{\alpha_3(\alpha_3-\alpha_2)}{2\alpha_2(1-2\alpha_2)}$	$\frac{\alpha_3(\alpha_3-\alpha_2)}{2\alpha_2(1-2\alpha_2)}$
$\frac{14-3\sqrt{5}}{16}$	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

(o) (5,4)-RKV, SCRATON-Formel

Ordnung $p = 5$ wird erreicht, falls für alle t entlang der Lösung gilt $f_t(t, x) \neq 0$.

$\frac{2}{9}$	$\frac{2}{9}$				
$\frac{1}{3}$	$\frac{1}{12}$	$\frac{3}{12}$			
$\frac{3}{4}$	$\frac{69}{128}$	$-\frac{243}{128}$	$\frac{270}{128}$		
$\frac{9}{10}$	$-\frac{3105}{10000}$	$\frac{18225}{10000}$	$-\frac{11016}{10000}$	$\frac{4896}{10000}$	
	$\frac{1445}{13770}$	0	$\frac{6561}{13770}$	$\frac{3264}{13770}$	$\frac{2500}{13770}$

(p) (6,5)-RKV, BUTCHER-Verfahren (1964)

Für $p \geq 5$ muss stets $s > p$ sein (BUTCHER 1964).

$\frac{1}{4}$	$\frac{1}{4}$				
$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$			
$\frac{1}{2}$	0	$-\frac{1}{2}$	1		
$\frac{3}{4}$	$\frac{3}{16}$	0	0	$\frac{9}{16}$	
1	$-\frac{3}{7}$	$\frac{2}{7}$	$\frac{12}{7}$	$-\frac{12}{7}$	$\frac{8}{7}$
	$\frac{7}{90}$	0	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$

(q) (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
	$\frac{11}{120}$	0	$\frac{81}{120}$	$\frac{81}{120}$	$-\frac{32}{120}$

(r) (6,5)-RKV, NYSTRÖM-Formel (1925)

(s) (8,6)-RKV, HUTA-Formel (1956)

Explizite ESV bieten sich für die näherungsweise Lösung von nicht steifen SysgDGl an.

2.3 Konvergenz von ESV

Bisher haben wir uns nur mit dem *lokalen Diskretisierungsfehler* befasst, dem Fehler also, der in jedem Zeitschritt $h = \Delta t$ neu entsteht dadurch, dass die zu approximierende Lösung eines AWP durch ein Geradenstück approximiert wird. Wir werden nun feststellen, dass dieser Fehler auch maßgeblich ist für die *globale Abweichung* der Näherung von der exakten Trajektorie.

Im Folgenden untersuchen wir im Zeitintervall $[t_0, T]$ das AWP

$$x' = f(t, x), \quad x(t_0) = x_0. \quad (2.32)$$

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 einer Funktion Ψ ein ESV der Form (2.8) 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

$$e(t, h) = \eta_N - x(t), \quad h = (t - t_0)/N, \quad (2.33)$$

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

$$\lim_{h \rightarrow 0} e(t, h) = 0 \quad \text{bzw.} \quad \lim_{N \rightarrow \infty} e(t, h) = 0. \quad (2.34)$$

Das folgende zentrale Ergebnis stellt den globalen Fehler in Relation zum lokalen Diskretisierungsfehler.

[2.12] Satz Die Funktion Ψ sei stetig und erfülle die Lipschitz-Bedingung

$$|\Psi(t, x_1, h) - \Psi(t, x_2, h)| \leq M |x_1 - x_2| \quad (2.35)$$

mit einer Konstanten $M \geq 0$.

Dann gilt für das durch Ψ definierte ESV (2.8):

Ist das ESV ein Verfahren der Konsistenzordnung p (vergl. Def. [2.2] (d)), so erfüllt mit $h = (t - t_0)/N$ und mit einem geeigneten $C > 0$ der globale Diskretisierungsfehler

eine Abschätzung der Form

$$|e(t, h)| \leq C h^p \frac{e^{M(t-t_0)} - 1}{M}. \quad (2.36)$$

Bevor dieser Satz bewiesen wird, soll seine Aussage kurz kommentiert werden.

[2.13] Bemerkungen

(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 (2.35) lässt sich in den oben vorgestellten Verfahren leicht aus der Lipschitz-Stetigkeit (1.18) 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 (2.6) 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 (2.36) mit zunehmenden Abstand vom Anfangspunkt t_0 exponentiell wächst.

Zum Beweis des Satzes [2.12] benötigen wir das folgende Hilfsergebnis.

[2.14] Lemma Genügt eine Zahlenfolge ξ_i , $i = 0, 1, 2, \dots$, einer Abschätzung der Form

$$|\xi_{i+1}| \leq (1 + \delta)|\xi_i| + B \quad (2.37)$$

mit den Konstanten $\delta > 0$ und $B \geq 0$, so gilt für alle $k \in \mathbb{N}$

$$|\xi_k| \leq e^{k\delta} |\xi_0| + B \frac{e^{k\delta} - 1}{\delta}. \quad (2.38)$$

Beweis von Lemma [2.14]:

Die iterative Anwendung der Abschätzung (2.37) ergibt mit 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 $1 + \delta \leq e^\delta$. \square

Beweis von Satz [2.12]: 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 (2.8) 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 (2.35) 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 ist (vgl. (2.10), (2.13)) 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

$$|e_{i+1}| \leq (1 + Mh) |e_i| + C h^{p+1}. \quad (2.39)$$

Wegen Lemma [2.14] folgt hieraus zunächst

$$\begin{aligned}
|e_N| &\leq e^{M Nh} |e_0| + C h^{p+1} \frac{e^{M Nh} - 1}{Mh} \\
&\leq e^{M(t-t_0)} |e_0| + C h^p \frac{e^{M(t-t_0)} - 1}{M}.
\end{aligned} \quad (2.40)$$

Bei genauer Berücksichtigung der AB ist $e_0 = 0$ und man erhält die gewünschte Abschätzung (2.36). \square

[2.15] Bemerkungen

(a) Um in jedem Fall die Konvergenz zu sichern, muss bezüglich der AB

$$\lim_{h \rightarrow 0} e_0 = 0 \quad \text{bzw.} \quad |e_0| \leq C_1 h^q, \quad q > 0, \quad C_1 \geq 0, \quad (2.41)$$

gelten. Meisten macht man jedoch bei der Approximation der AB keine Fehler.

(b) Falls $|e_0| \leq C_1 h^p$ ist, und insbesondere bei $e_0 = 0$, überträgt sich die Konsistenzordnung p auf die Konvergenz und damit 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.

[2.16] Beispiel Wir testen mehrere Verfahren unterschiedlicher Konvergenzordnung, das PZV, MPZV, HV, 3-stufige RKV 2. Ordnung (2.18) und KRKV, am AWP

$$x'(t) = t x(t), \quad x(0) = 1. \quad (2.42)$$

Die exakte Lösung ist gegeben durch $x(t) = e^{t^2/2}$. Die 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 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 $x(5) = 2.683\,372\,865\,208\dots\text{E}+05$ wird das Intervall $[0, T] = [0, 5]$ in $N = 50, 500$ und 5000 gleiche Teilintervalle unterteilt.

In der Tabelle 2.1 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. 2.1: Näherungswerte η_N der ESV im Vergleich zu $x(5) = 2.683\,372\,865\,208\dots\text{E}+05$ (Dezimalzahlen abgeschnitten)

Die globalen Fehler $|\eta_N - x(5)|$ sind in der Tabelle 2.2 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. 2.2: Fehler bei numerischer Berechnung zu $x(5) = 2.683\ 372\ 865\ 208\dots E+05$
(Dezimalzahlen abgeschnitten)

Es zeigt sich, dass das KRKV bezüglich der Fehlerrgüte die anderen deutlich übertrifft und ohne großen Aufwand mit vier Funktionswertberechnungen pro Schritt eine Genauigkeit liefert, welche nahe an rechnerinterne Darstellungsgenauigkeiten reeller Zahlen 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. ◀

Verschiedene Faktoren sind also bei der Schrittweitenwahl zu berücksichtigen. Die Abb. 2.4 gibt eine qualitative Darstellung des Gesamtfehlers im äquidistanten Fall. 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 eine untere Schranke, welche die Schrittweite nicht unterschreiten sollte.

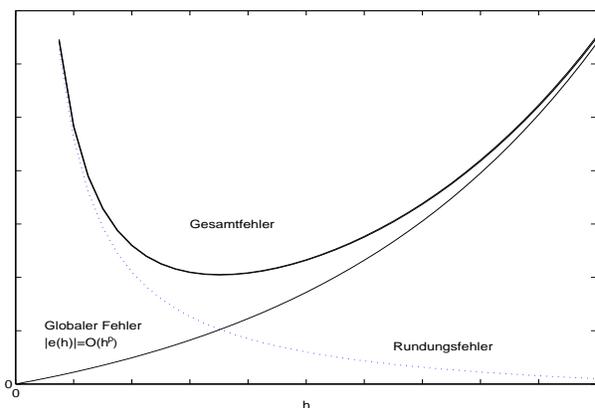


Abb. 2.4: Verhalten des globalen Fehlers und Rundungsfehlers in Abhängigkeit von h

2.4 Schrittweitensteuerung

Bei der Wahl einer geeigneten Schrittweite h muss ein Kompromiss gefunden werden zwischen hoher Rechengenauigkeit und moderatem Rechenaufwand. Hinzu kommt, dass sich die "ideale" oder "optimale" Schrittweite im Verlauf der Integration des AWP ändern kann. Das Ziel ist eine *adaptive Schrittweitensteuerung* (SWS, adaptive Gittersteuerung), welche im Verlauf der Rechnungen die Schrittweite korrigiert und optimiert. Einige Möglichkeiten sollen aufgezeigt werden.

2.4.1 Schrittweitenkontrolle mit lokalem Diskretisierungsfehler

Hierbei ist es nötig, den lokalen Diskretisierungsfehler zu messen. Dies geschieht durch die Auswertung eines anderen geeigneten ESV, des so genannten *Kontrollverfahrens*, welches eine höhere Ordnung als das eingesetzte Verfahren hat.

Seien ein ESV und ein Kontrollverfahren zu einem AWP für eine gDGL gegeben durch

$$\eta_{i+1} = \eta_i + h \Psi(t_i, \eta_i, h), \quad (2.43)$$

$$\hat{\eta}_{i+1} = \eta_i + h \hat{\Psi}(t_i, \eta_i, h). \quad (2.44)$$

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

$$\begin{aligned} \frac{1}{h} |\eta_{i+1} - \hat{\eta}_{i+1}| &= \frac{1}{h} |\eta_{i+1} - z(t_{i+1}) + z(t_{i+1}) - \hat{\eta}_{i+1}| \\ &\leq \frac{1}{h} |\eta_{i+1} - z(t_{i+1})| + \frac{1}{h} |\hat{\eta}_{i+1} - z(t_{i+1})| \\ &\leq C h^p + \hat{C} h^{p+1} \approx C h^p \end{aligned} \quad (2.45)$$

und daher

$$\frac{1}{h} |\eta_{i+1} - \hat{\eta}_{i+1}| \approx \frac{1}{h} |\eta_{i+1} - z(t_{i+1})| = |\tau(t_i, \eta_i, h)|. \quad (2.46)$$

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

$$\eta_{i+1} = \eta_i + h \sum_{i=1}^s b_i k_i \quad (2.47)$$

mit ein, welcher ja ohnehin bei akzeptierter Schrittweite berechnet werden muss.

Benutzt werden soll das KRKV der Ordnung $p = 4$ als Kontrollverfahren zur Steuerung eines Verfahrens 3. Ordnung. Das KRKV ist gegeben durch

$$\mathbf{b} = \frac{1}{6} \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (2.48)$$

mit der Iterationsvorschrift

$$\eta_{i+1} = \eta_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4). \quad (2.49)$$

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 [2.9] 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

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ 0 & \frac{1}{4} & \frac{1}{4} & 1 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \\ \hat{b}_4 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \frac{1}{6} \end{pmatrix}. \quad (2.50)$$

Dieses Gleichungssystem ist regulär und hat die eindeutige Lösung $\hat{\mathbf{b}} = \mathbf{b} = \frac{1}{6}(1, 2, 2, 1)^T$, so dass 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

$$k_5 = f(t_i + h, \eta_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)) \quad (2.51)$$

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, lautet nun

$$\tilde{A} \hat{\mathbf{b}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 & 1 \\ 0 & \frac{1}{4} & \frac{1}{4} & 1 & 1 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \\ \hat{b}_4 \\ \hat{b}_5 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \frac{1}{6} \end{pmatrix}. \quad (2.52)$$

Seine allgemeine parameterabhängige Lösung lautet

$$\hat{\mathbf{b}} = \frac{1}{6}(1, 2, 2, 1 - 6u, 6u), \quad u \in \mathbb{R}, \quad (2.53)$$

und es ist

$$\text{kern}(\tilde{A}) = \{\mathbf{z} : \tilde{A}\mathbf{z} = \mathbf{0}\} = \text{span}(0, 0, 0, 1, -1)^T. \quad (2.54)$$

Eine spezielle Lösung bei $u = 0$ ist

$$\hat{\mathbf{b}} = \frac{1}{6}(1, 2, 2, 1, 0)^T, \quad (2.55)$$

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

$$\hat{\mathbf{b}} = \frac{1}{6}(1, 2, 2, 0, 1)^T, \quad (2.56)$$

$$\hat{\mathbf{b}} = \frac{1}{6}(1, 2, 2, \frac{1}{2}, \frac{1}{2})^T. \quad (2.57)$$

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 $\hat{\eta}_{i+1}$ 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 FEHLBERG zurück und ist in den eingebetteten RUNGA-KUTTA-FEHLBERG-Verfahren (RKfV) realisiert.

[2.17] Beispiel (a) Gegeben sei das AWP

$$x' = f(t, x) = t \cos(x), \quad x(-2) = 1, \quad t \in [t_0, T] = [-2, 5]. \quad (2.58)$$

Die exakte Lösung ist

$$x(t) = \arctan \left(\frac{(1 + \sin(1)) e^{t^2-4} - 1 + \sin(1)}{(1 + \sin(1)) e^{t^2-4} + 1 - \sin(1)}, \frac{2 \cos(1) e^{t^2/2-2}}{(1 + \sin(1)) e^{t^2-4} + 1 - \sin(1)} \right), \quad (2.59)$$

wobei

$$\arctan(y, z) = -\imath \ln \left(\frac{z + \imath y}{\sqrt{z^2 + y^2}} \right). \quad \imath = \sqrt{-1}.$$

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 2.5 die exakte Lösung des AWP (2.58) als gestrichelte Kurve eingetragen. Seine Näherungslösung mit dem KRKV liegt nur knapp daneben.

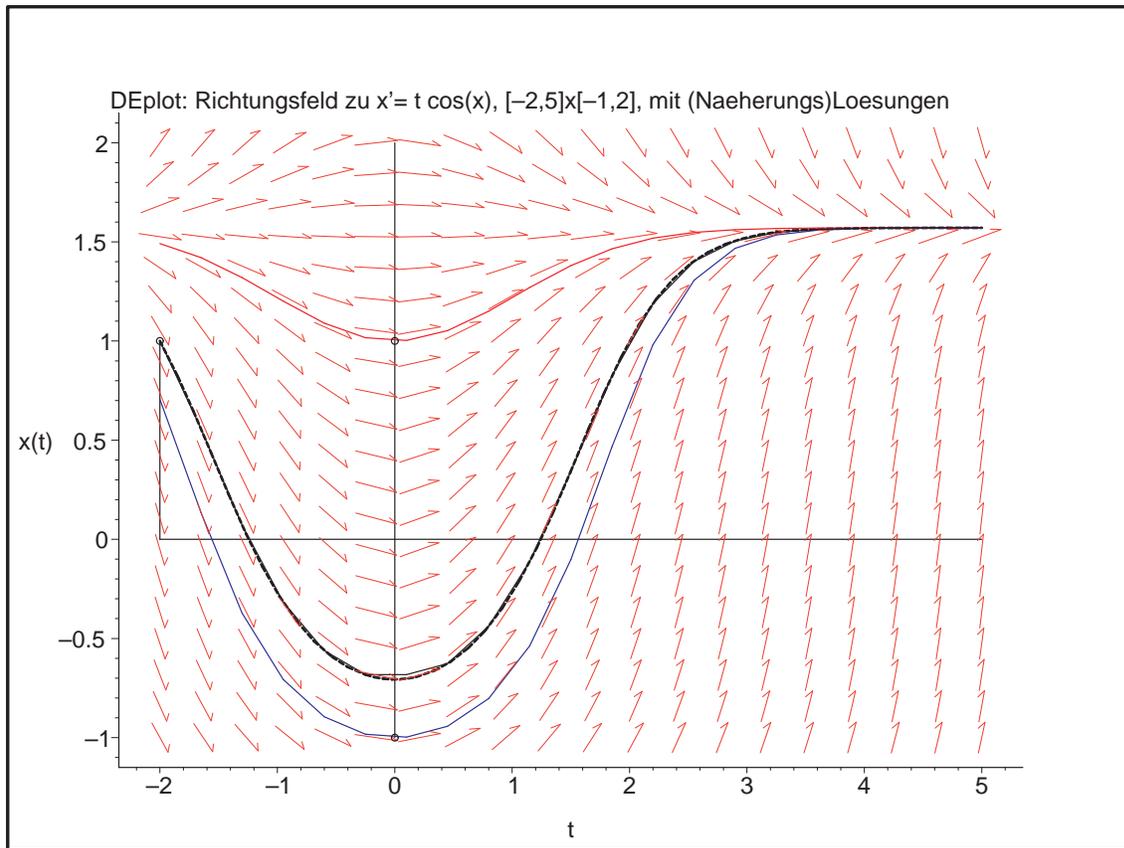


Abb. 2.5: Datei *richt8.ps*,
 Richtungsfeld zur gDGL $x' = f(t, x) = t \cos(x)$ sowie Näherungslösungen
 mit den AB $x(-2) = 1, x(0) = -1, x(0) = 1$, sowie $x(t)$ mit AB $x(-2) = 1$

Wir zeichnen das Richtungsfeld mit MATLAB und tragen die Trajektorie (2.59) ein.

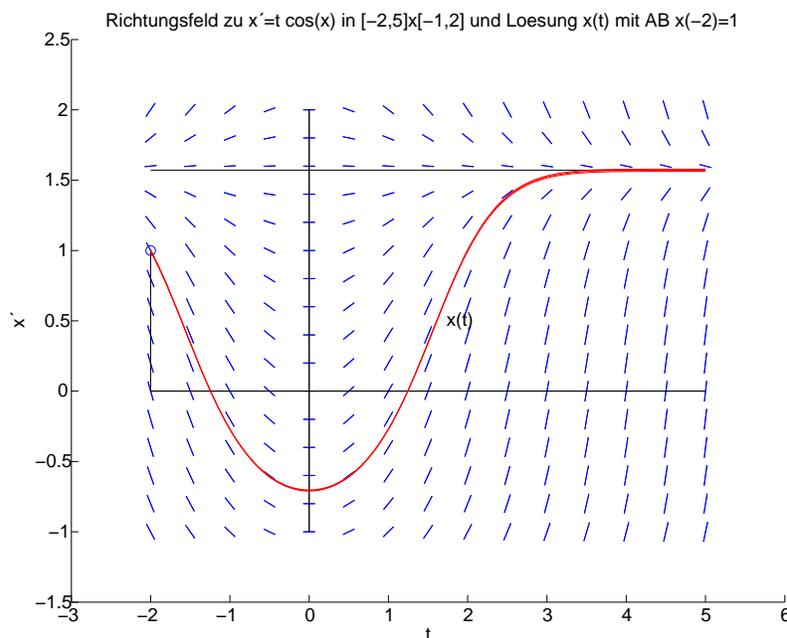


Abb. 2.6:
 Datei *r_f12b.ps*,
 Richtungsfeld zur gDGL
 $x' = f(t, x) = t \cos(x)$
 sowie Lösung $x(t)$
 mit der AB $x(-2) = 1$

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 (2.18)	1.570 768 044	0.000 012 123	0.000 007 717
(5,3)-RKV, b mit (2.56)	1.570 785 264	0.000 005 096	0.000 003 244
(5,3)-RKV, b mit (2.57)	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. 2.3: 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 (2.18)	0.010 693 805	1.362 920 749
(5,3)-RKV, b mit (2.56)	0.000 892 715	0.063 207 246
(5,3)-RKV, b mit (2.57)	0.000 316 554	0.041 707 089
KRKV	0.000 302 954	0.020 213 802

Tab. 2.4: Maximale Fehler im Intervall $[-2, 5]$

Wir bemerken, dass das (5,3)-RKV, **b** mit (2.57), einen Vergleich mit dem KRKV einigermaßen Stand hält.

```
% 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)
    h = (t1-t0)/N;
    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;
```

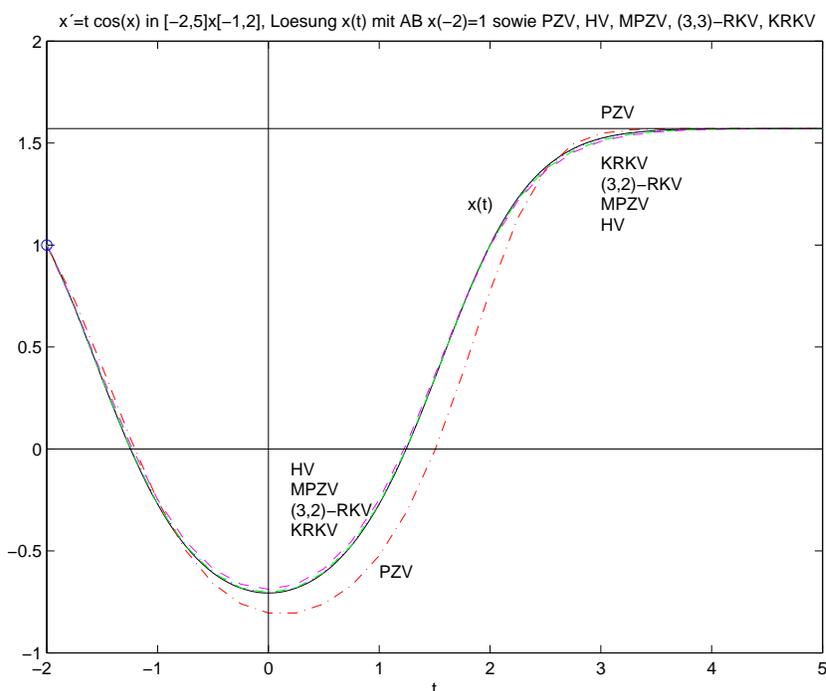


Abb. 2.7:
 Datei *r-f12c.ps*,
 $x' = f(t, x) = t \cos(x)$
 und Lösung $x(t)$
 mit der AB $x(-2) = 1$,
 dazu Näherungen
 mit PZV, HV, MPZV,
 (3,2)-RKV, KRKV

Die Schrittweite h soll nun kontrolliert, aber noch nicht gesteuert werden.

Wir rechnen hier an äquidistanten Knoten $t_i = t_0 + ih = -2 + \frac{i}{4}$, $i = 0, 1, \dots, N = 28$, und beobachten nur, wie sich der Fehler verhält. Damit wäre dann im Weiteren auch eine Empfehlung für Schrittweitenveränderungen machbar.

Für die Schrittweitenkontrolle benutzen wir, wie das in der Datei *rkv1_3b.m* erkennbar ist, die RKV der Ordnung $p = 3$ mit (2.56), (2.57) 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 der nachfolgenden Tabelle verwenden wird das (5,3)-RKV, **b** mit (2.56), und das KRKV.

In der Spalte 2 sind 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 Gegenüberstellung.

$t(i)$	η_{i+1}	$\hat{\eta}_{i+1}$	$ \eta_{i+1} - \hat{\eta}_{i+1} $	$ \eta_{i+1} - \hat{\eta}_{i+1} $	η_{i+1}	$\hat{\eta}_{i+1}$
-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 (2.57), 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 führen auf ähnliche Ergebnisse. Dazu gewinnt man noch folgende Erkenntnisse. Am Anfang der Rechnung muss sich eine Schrittweite meist erst "einpendeln". Im Verlauf der Iteration werden dann in Bereichen mit flachem Lösungverlauf die Kontrollgrößen kleiner, was zu einem Vorschlag für eine Schrittweitenvergrößerung führt. Dort wo starke Anstiege bzw. Krümmungen auftreten, wachsen die Kontrollgrößen, so dass man h entsprechend verkleinern sollte.

(b) Gegeben sei das AWP

$$x' = f(t, x) = \frac{1}{x + 10^{-p}}, \quad p = 1, \quad x(0) = 0, \quad t \in [t_0, T] = [0, 5]. \quad (2.60)$$

Die beiden allgemeinen Lösungen sind

$$x_1(t) = -10^{-p} + \sqrt{10^{-2p} + 2(t + C)}, \quad x_2(t) = -10^{-p} - \sqrt{10^{-2p} + 2(t + C)}. \quad (2.61)$$

Die exakte Lösung des AWP ist

$$x(t) = -10^{-p} + \sqrt{10^{-2p} + 2t}, \quad p = 1. \quad (2.62)$$

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`.

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.

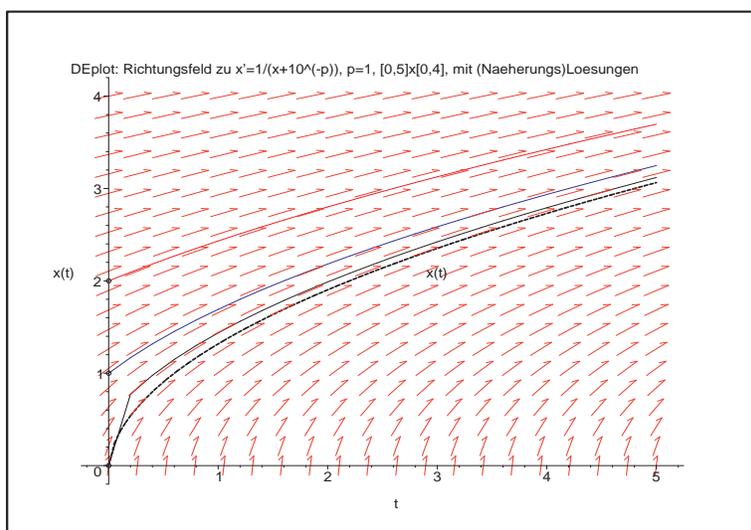


Abb. 2.8:
 Datei `richt11.ps`,
 Richtungsfeld zur gDGL
 $x' = 1/(x + 10^{-p})$, $p = 1$,
 sowie Näherungslösungen
 mit den AB $x(0) = 0$,
 $x(0) = 1$, $x(0) = 2$,
 sowie $x(t)$ mit AB $x(0) = 0$
 (untere gestrichelte Kurve)

Wir zeichnen das Richtungsfeld mit `MATLAB` und tragen die Trajektorie (2.62) ein.

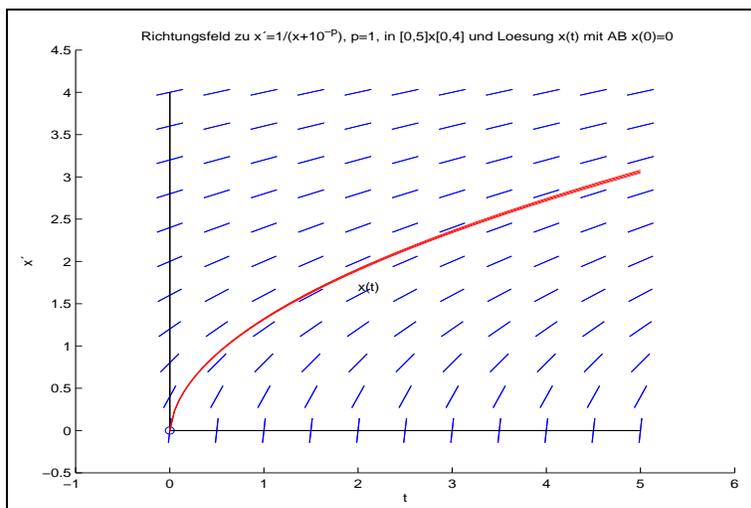


Abb. 2.9:
 Datei `r_f11b.ps`,
 Richtungsfeld zur gDGL
 $x' = 1/(x + 10^{-p})$, $p = 1$,
 sowie Lösung $x(t)$
 mit der AB $x(0) = 0$

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 (2.18)	3.127 357 066	0.063 498 662	0.020 725 064
(5,3)-RKV, b mit (2.56)	3.121 312 066	0.057 453 662	0.018 752 062
(5,3)-RKV, b mit (2.57)	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. 2.5: 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 (2.18)	0.262 045 627	0.484 989 084
(5,3)-RKV, b mit (2.56)	0.241 051 231	0.446 133 052
(5,3)-RKV, b mit (2.57)	0.236 465 096	0.437 645 122
KRKV	0.231 878 962	0.429 157 191

Tab. 2.6: Maximale Fehler im Intervall $[0, 5]$

Wir bemerken, dass die (5,3)-RKV durchaus mit dem KRKV vergleichbar sind.

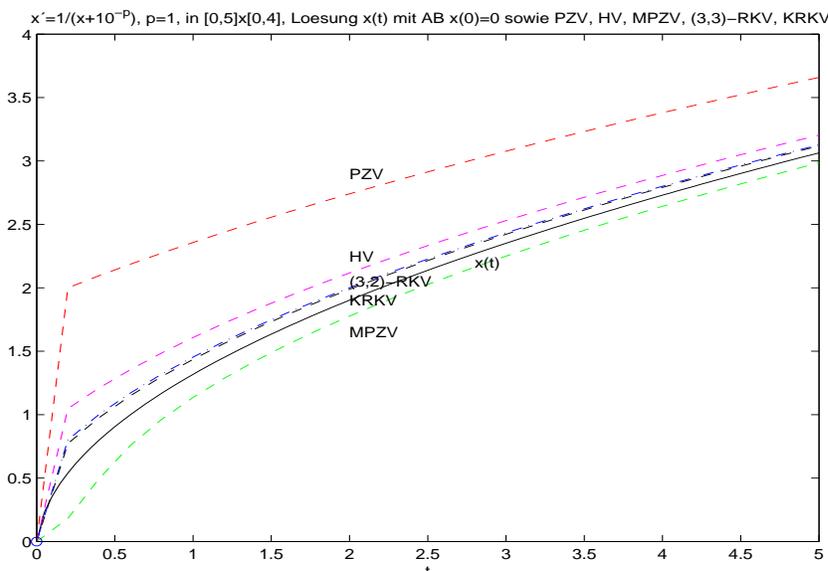


Abb. 2.10:
 Datei *r_f11c.ps*,
 $x' = 1/(x + 10^{-p})$, $p = 1$,
 und Lösung $x(t)$
 mit der AB $x(0) = 0$,
 dazu Näherungen
 mit PZV, HV, MPZV,
 (3,2)-RKV, KRKV
 bei $h = 0.2$

Die Schrittweite h soll nun 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 nur, wie sich die Fehler verhalten.

Für die Schrittweitenkontrolle benutzen wir wiederum (5,3)-RKV der Ordnung $p = 3$ mit (2.56), (2.57) und das KRKV als Kontrollverfahren.

Die nachfolgenden Tabellen sind wie im Teil (a) des Beispiels zu verstehen.

(5,3)-RKV, \mathbf{b} mit (2.56), 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 (2.57), und KRKV an ausgewählte 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

Die Ergebnisse sind wie im Teil (a). Nach einer notwendigen Anpassung der Schrittweite am Anfang des Intervalls durch den starken Anstieg der Lösung kann anschließend eine stetige und schnelle Schrittweitenvergrößerung vorgenommen werden. ◀

2.4.2 Schrittweitenkontrolle mit Schrittkenzahl

In numerischen Berechnungen hat man beobachtet, 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 (1.18) für f

$$\begin{aligned} \kappa &= 2 \left| \frac{k_3 - k_2}{k_2 - k_1} \right| \\ &= 2 \left| \frac{f(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_2) - f(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_1)}{k_2 - k_1} \right| \\ &\approx 2 \left| \frac{\frac{h}{2}(k_2 - k_1)}{k_2 - k_1} \right| \left| \frac{\partial f}{\partial x} \right| = h \left| \frac{\partial f}{\partial x} \right|. \end{aligned} \tag{2.63}$$

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 Iteration berechnen und kontrollieren. Mehr noch, mittels geeigneter Wahl von h gelingt es, 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 die nächste Iteration machen. 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 (2.63) umgekehrt die günstige Schrittweite

$$h = h(t) = \frac{\kappa}{|f_x(t, x(t))|}, \quad \kappa \in [0.1, 0.2], \tag{2.64}$$

an jeder Stelle t des Lösungsintervalls.

Für lineare gDGL der Form

$$x' = \alpha x + \beta t + \gamma \tag{2.65}$$

mit reellen Koeffizienten gilt sofort $f_x = \alpha = \text{const}$ und

$$\kappa = 2 \left| \frac{k_3 - k_2}{k_2 - k_1} \right| = h|\alpha| = h|f_x|, \tag{2.66}$$

so dass eine konstante Schrittweite h im ganzen Intervall genommen werden kann. Sie wird lediglich durch die gewünschte Genauigkeit mit der Ordnung $\mathcal{O}(h^4)$ bestimmt.

[2.18] Beispiel Wir überprüfen die Bedingung (2.63) bei AWP.

(a) Gegeben sei das AWP

$$x' = f(t, x) = -x + t, \quad x(0) = 1, \quad t \in [t_0, T] = [0, 2]. \tag{2.67}$$

Die exakte Lösung ist $x(t) = 2e^{-t} + t - 1$. Die Schrittkenzahl beträgt

$$\kappa = 2 \left| \frac{k_3 - k_2}{k_2 - k_1} \right| = h |f_x| = h \cdot 1, \tag{2.68}$$

so dass man bei $\kappa = 0.2$ mit durchweg $h = 0.2$ rechnen kann.

(b) Für das AWP

$$x' = f(t, x) = tx, \quad x(0) = 1, \quad t \geq 0, \tag{2.69}$$

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. 2.7: Kontrolle der Schrittkenzahl κ bei konstantem $h = 0.4$

Damit sollte man die Schrittweite h im Laufe der Iteration geeignet verkleinern. Für die Schrittweitenfolge $(h_0, h_1, \dots, h_{13}) = (0.4, 0.4, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)$ 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. 2.8: Ungefähres Einhalten der Schranken der Schrittkenzahl κ bei variablem h

(c) Für das AWP

$$x' = f(t, x) = \frac{1}{x + 10^{-p}}, \quad p = 1, \quad x(0) = 0, \quad t \in [t_0, T] = [0, 5], \quad (2.70)$$

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.

$$\begin{aligned} h &= \frac{\kappa}{|f_x|} = 0.1 (x(t) + 10^{-p})^2 \\ &= 0.1 (-10^{-p} + \sqrt{10^{-2p} + 2t} + 10^{-p})^2 \\ &= 0.1 (10^{-2p} + 2t) \\ &= 0.2t + 10^{-2p-1}, \end{aligned} \quad (2.71)$$

was der Gestalt

$$h(t) = \alpha t + \beta, \quad \alpha = 0.2, \quad \beta = 10^{-2p-1}, \quad (2.72)$$

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.

$$m = \frac{\lg(\frac{\alpha T}{\beta} + 1)}{\lg(\alpha + 1)} = \frac{\lg(\beta^{-1} + 1)}{\lg(\alpha + 1)} = \frac{\lg(10^{2p+1} + 1)}{\lg(1.2)} \geq 12.629 \cdot 10^{2p+1}. \quad (2.73)$$

Bei $p = 1$ ist $h_0 = \beta = 10^{-3}$ 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. 2.9: 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, so dass insgesamt eine hohe Schrittzahl m folgt. ◀

Die Schrittkenzahl ist nicht in jedem Fall 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 [2.20] (b)).

2.4.3 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.

Bei der Schrittweite $2h$ liefert sie 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 $I = I_F + F_F$ für eine bessere Approximation mit höherer Genauigkeitsordnung verwendet werden.

Wir wollen hier mit dem ESV vorab nicht zwei Näherungen η_i und $\hat{\eta}_i$ 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

$$e_i = e(t_i, h) = \eta_i - x(t_i; t_0, x_0) = \eta_i - \Psi^{t_i, t_0} x_0 = Ch^p + \mathcal{O}(h^{p+1}). \quad (2.74)$$

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 diesem Teilintervall 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

$$e_{i+2} = e(t_{i+2}, 2h) = \eta_{i+2} - x(t_{i+2}; t_0, x_0) = C(2h)^p + \mathcal{O}(h^{p+1}), \quad (2.75)$$

$$\hat{e}_{i+2} = \hat{e}(t_{i+2}, h) = \hat{\eta}_{i+2} - x(t_{i+2}; t_0, x_0) = Ch^p + \mathcal{O}(h^{p+1}). \quad (2.76)$$

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

$$\hat{e}_{i+2} = \frac{\eta_{i+2} - \hat{\eta}_{i+2}}{2^p - 1} + \mathcal{O}(h^{p+1}). \quad (2.77)$$

Daraus folgt auch eine zweite für den Lösungswert

$$\begin{aligned} x(t_{i+2}; t_0, x_0) &= \hat{\eta}_{i+2} - \hat{e}_{i+2} = \hat{\eta}_{i+2} + \frac{\hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1} + \mathcal{O}(h^{p+1}) \\ &= \frac{2^p \hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1} + \mathcal{O}(h^{p+1}). \end{aligned} \quad (2.78)$$

Die Formel (2.78) 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.

[2.19] Bemerkungen

(a) Beide Beziehungen (2.77) und (2.78) 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 Gleitpunktarithmetik hinreichend stark ist.

(b) Für den Fehler der Feinrechnung \hat{e}_{i+2} liefert die Größe

$$est = \frac{\hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1} \quad (est = estimate) \quad (2.79)$$

i. Allg. eine gute Näherung.

(c) Das Runge-Prinzip beinhaltet eine Extrapolation.

Den aus Fein- und Grobrechnung zusammengesetzten oder extrapolierten Wert findet man in verschiedenen Darstellungen als

$$\tilde{\eta}_{i+2} = \frac{2^p \hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1} = \hat{\eta}_{i+2} + \frac{\hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1}. \quad (2.80)$$

Er ist eine um Größenordnung bessere Näherung als der Feinwert $\hat{\eta}_{i+2}$.

Beim KRKV (2.30) 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

$$est = \frac{k' + k'' - K}{15}. \quad (2.81)$$

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 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 , so dass 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 gilt

$$|\eta_{i+1} - x(t_{i+1}; t_i, \eta_i)| \leq \varepsilon$$

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 2.4.1 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 (2.75), (2.76) 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

$$x(t_{i+2}; t_i, \eta_i) = \hat{\eta}_{i+2} - \hat{e}'_{i+2} = \hat{\eta}_{i+2} + \underbrace{\frac{\hat{\eta}_{i+2} - \eta_{i+2}}{2^p - 1}}_{\substack{\text{est} \\ \tilde{\eta}_{i+2}}} + \mathcal{O}(h^{p+2}).$$

Das entspricht dem Ergebnis der Formel (2.78). Damit treffen auch die bisherigen Bemerkungen zum Runge-Prinzip und 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 eine Prozedur $KRKV_S(\)$ mit entsprechenden Parametern für den RK-Schritt verwendet.

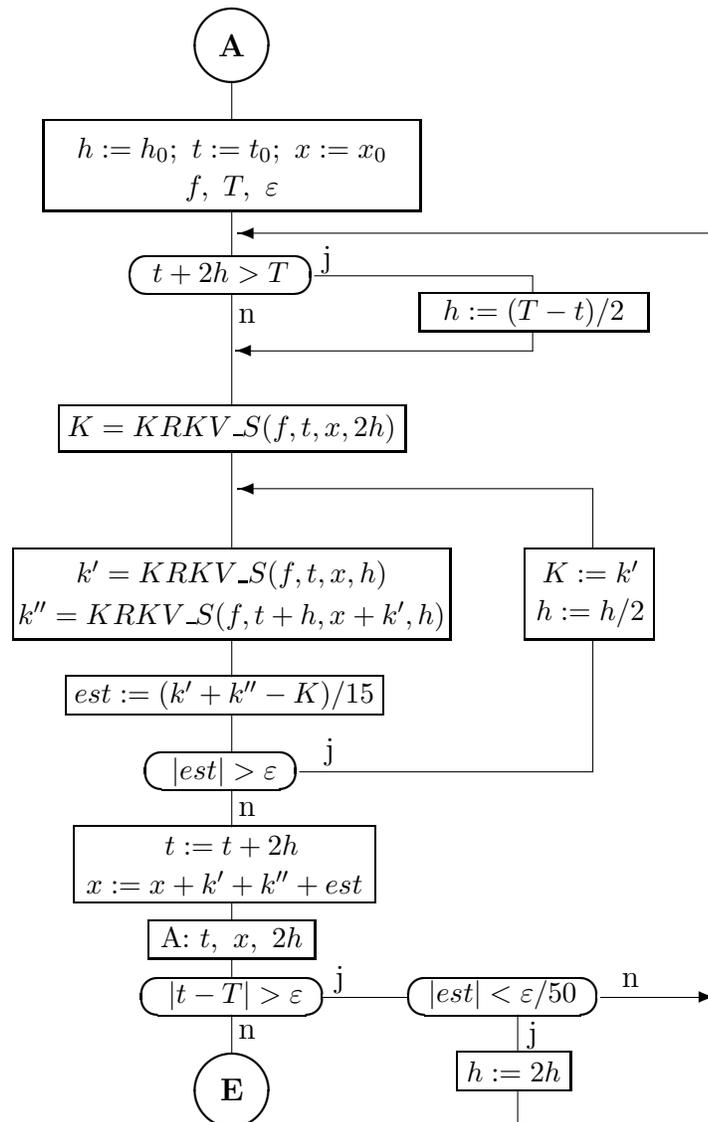


Abb. 2.11: Datei *dsws_rkv.pic*,
Flussdiagramm zum KRKV mit dualer SWS

Die duale SWS mit Verdoppeln und Halbieren der Schrittweite ist überschaubar, aber natürlich noch nicht optimal bezüglich einer möglichst kleinen Anzahl der Funktionswertberechnungen. Die Voraussage einer variablen und damit mehr angepassten Schrittweite erfordert jedoch einen aufwändigeren Steuermechanismus.

Folgende MATLAB-Prozeduren 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 (Abb. 2.11) 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 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;
    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.

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));

```

[2.20] Beispiele

(a) Wir überprüfen anhand des AWP der RICATTI-DGL

$$x' = t - x^2, \quad x(0) = 0, \quad t \in [t_0, T] = [0, 1], \quad (2.82)$$

dass der lokale sowie globale Diskretisierungsfehler des KRKV $\eta_{i+1} = \eta_i + h\Psi(t_i, \eta_i, h)$, $i = 0, 1, \dots, N-1$, $h = (T - t_0)/N = 1/N$, von der Ordnung $p = 4$ ist.

Bei einer Schrittweithalbierung sollte sich der Fehler rund 16 Mal verkleinern.

In MATLAB bei *double*-Arithmetik lösen wir das AWP für $N = 5, 10, 20, \dots$ Schritte.

Die Ergebnisse η_i , $i = 0, 1, \dots, N$, und entsprechende 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 23	0.2760
640	0.0015 625	0.4555 4452 6081 83	0.0000 0000 0000 83	-
	$x(1)$	0.4555 4452 6081 8735		16

Tab. 2.10: 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

$$x' = f(t, x) = t \cos(x), \quad x(-2) = 1, \quad t \in [t_0, T] = [-2, 5], \quad (2.83)$$

mit der exakten Lösung

$$x(t) = \arctan \left(\frac{(1 + \sin(1)) e^{t^2-4} - 1 + \sin(1)}{(1 + \sin(1)) e^{t^2-4} + 1 - \sin(1)}, \frac{2 \cos(1) e^{t^2/2-2}}{(1 + \sin(1)) e^{t^2-4} + 1 - \sin(1)} \right). \quad (2.84)$$

Richtungsfeld, die Trajektorie $x(t)$ und Näherungslösungen mit ESV haben wir schon im Beispiel [2.17] (a) demonstriert.

Nun vergleichen wir Fehler des KRKV für verschiedene 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. 2.11: Fehler bei $t = 5$ und maximale Fehler im Intervall $[-2, 5]$

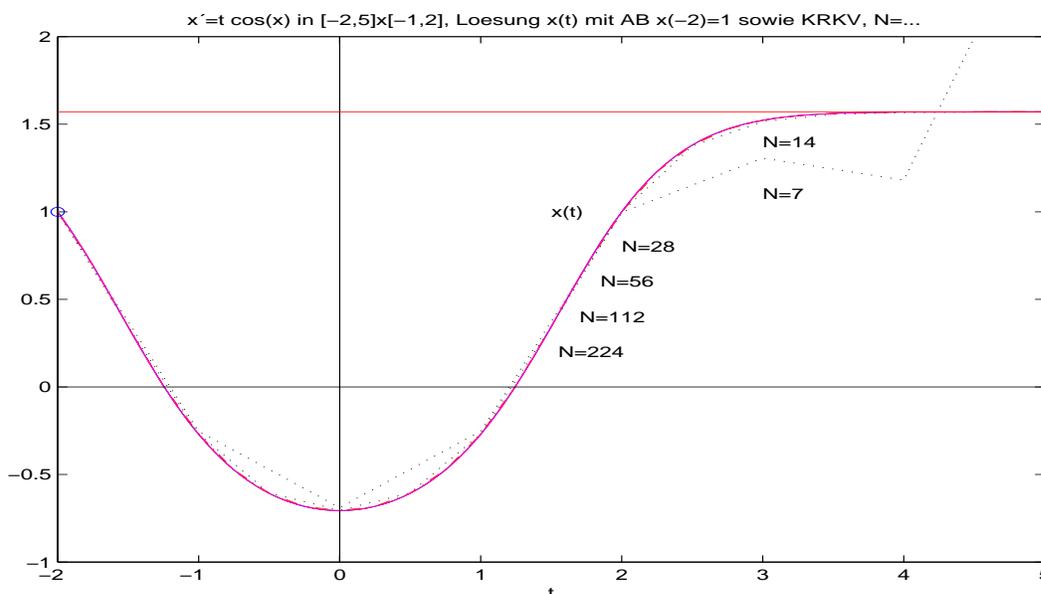


Abb. 2.12: Datei *r_f12f.ps*, Näherungslösungen zur gDGl $x' = t \cos(x)$, $x(-2) = 1$, mit KRKV bei verschiedenen N

Die Näherungslösungen zeigen ein moderates Verhalten.

Mit der Schrittweite $h = 1$ ist natürlich keine 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. 2.12: Ergebnisse zum KRKV mit SWS bei geforderter 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 Gleitpunktarithmetik *double* treten sehr große Schrittzahlen auf und mit den dabei in jedem akzeptierten Schritt gemachten 12 Funktionswertberechnungen machen sich Rundungsfehler bemerkbar. Diese lassen dann keine Verbesserung der Lösungsgenauigkeit mehr zu.

Nun noch zum Vergleich zwischen KRKV ohne und mit SWS.

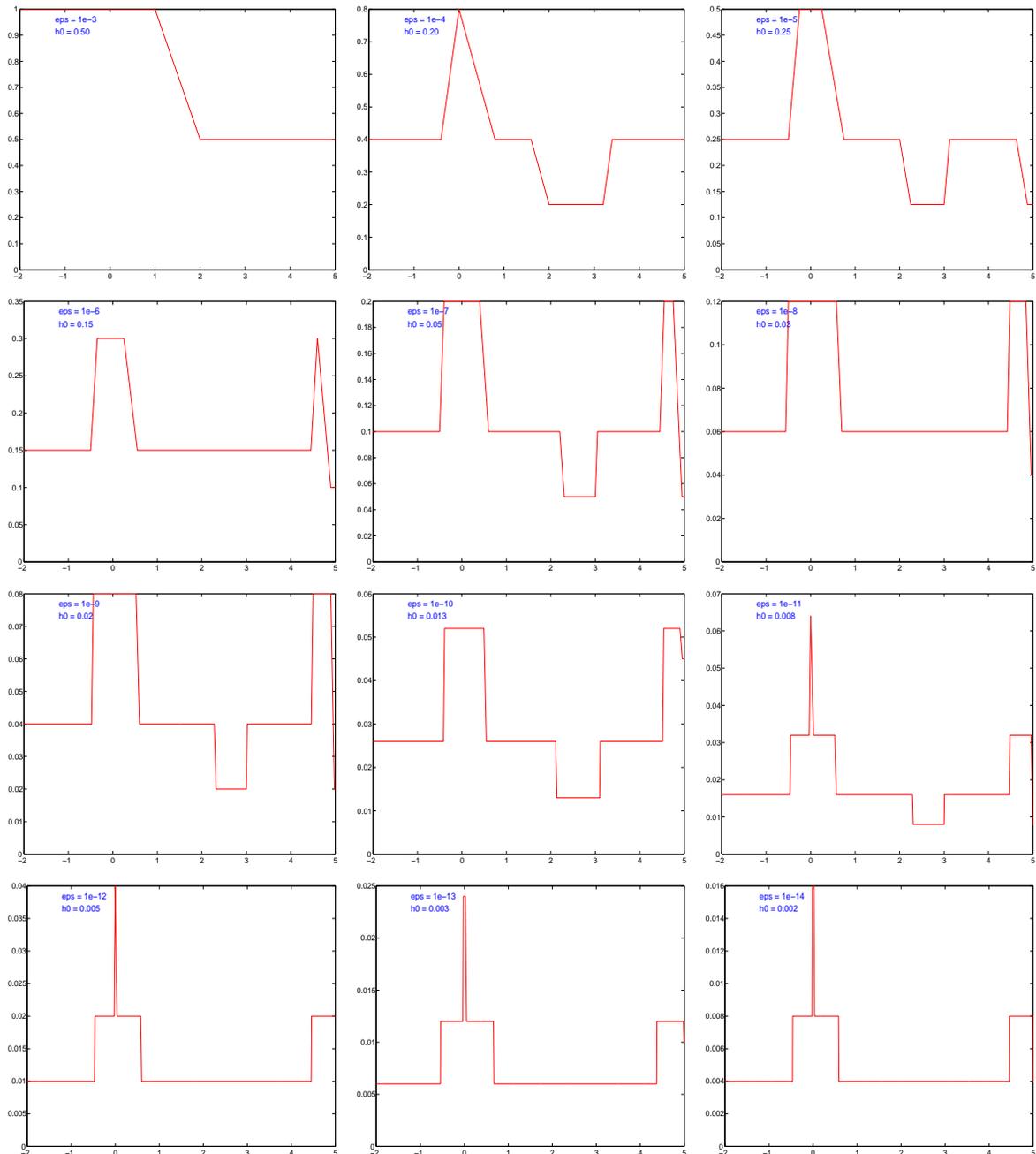
Wir wählen Varianten bei ähnlicher Lösungsgenauigkeit $\max_i |\eta_i - x(t_i)|$ aus und untersuchen dabei die einfachen Schrittweiten h und die Anzahl *anzf* der Funktionswertberechnungen.

$\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. 2.13: 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 Schrittweiten 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 nahe Null wieder schnell "zurückgenommen" werden muss auf das Niveau wie zu Beginn der Rechnung. Im Verlauf einer Phase eines starken Funktionsanstiegs bzw. Krümmung kann es, wie zwischen $t = 2$ und $t = 3$, zu einer weiteren Schrittweitenverkleinerung kommen.



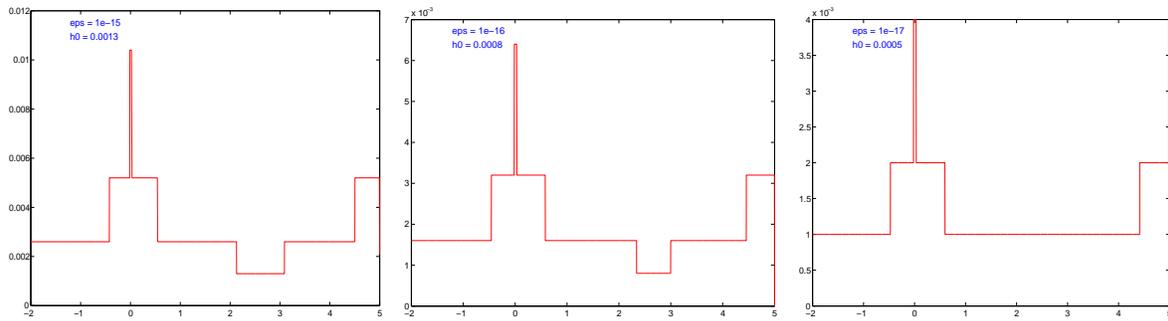


Abb. 2.13: Dateien *r_f12g3.ps*, *r_f12g4.ps*, ..., *r_f12g17.ps* (v.l.o. n.r.u.),
 Schrittweitenverhalten $2h_i(t_i)$ beim KRKV mit SWS
 bei $\varepsilon = 10^{-k}$, $k = 3, 4, \dots, 17$, und geeigneter Wahl von h_0

(c) Gegeben sei das AWP

$$x' = f(t, x) = 10\left(x - \frac{t^2}{1+t^2}\right) + \frac{2t}{(1+t^2)^2}, \quad t \in [t_0, T] = [0, 5], \quad x(t_0) = x_0. \quad (2.85)$$

Die allgemeine Lösung der gDGl ist

$$x(t) = Ce^{10t} + \frac{t^2}{1+t^2}, \quad (2.86)$$

die exakte Lösung des AWP mit $x(t_0) = x_0$

$$x(t) = x_0 e^{10t} + \frac{t^2}{1+t^2}. \quad (2.87)$$

Zunächst interessieren wir uns für das Richtungsfeld zur gDGl und AWP mit aus-
 gewählten AB. Wir nutzen MATLAB mit der Prozedur `dfield.m` aus Abschnitt 1.1.

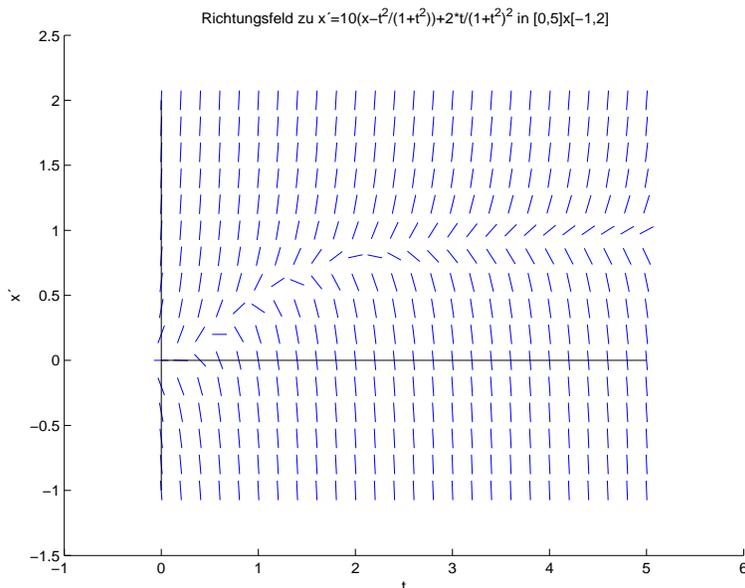


Abb. 2.14:
 Datei *r_f1a.ps*,
 Richtungsfeld zur gDGl
 $x' = f(t, x)$
 in $[0, 5] \times [-1, 2]$

Wir zeichnen das Richtungsfeld und tragen Trajektorien (2.87) mit gegebenen AB ein.

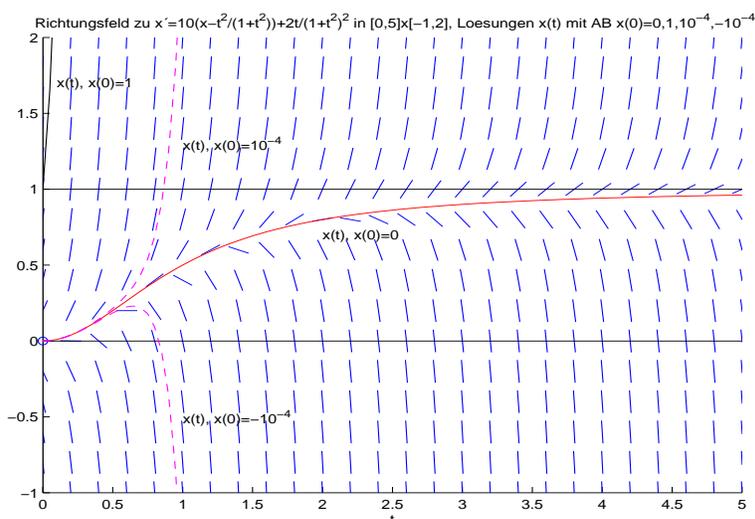


Abb. 2.15:
 Datei *r_f1b.ps*,
 Richtungsfeld zur gDGL
 $x' = f(t, x)$
 sowie Lösungen $x(t)$
 mit den AB $x(0) = 0$
 $x(0) = 1, 10^{-4}, -10^{-4}$

Dabei bemerkt man, dass die 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 interessant, wie sich die ESV verhalten, die ja auch nur in 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 $h = (T - t_0)/N = 5/N$, $N = 1000$.

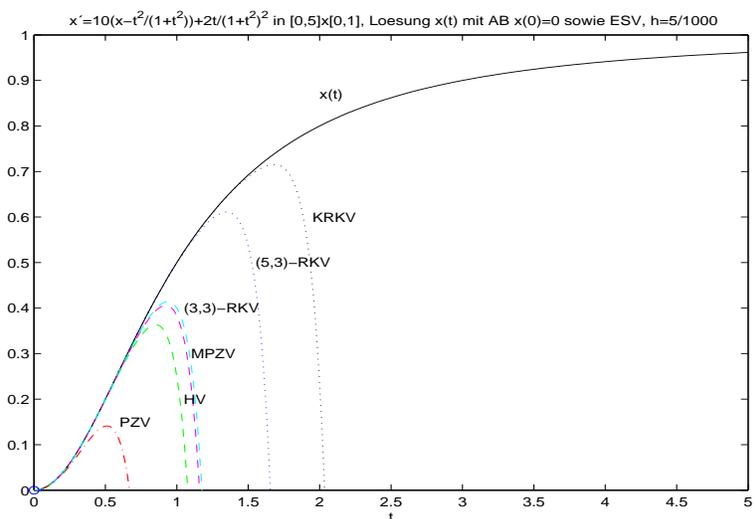


Abb. 2.16:
 Datei *r_f1c.ps*,
 gDGL $x' = f(t, x)$, $t \in [0, 5]$
 und Lösung $x(t)$
 mit den AB $x(0) = 0$,
 sowie Näherungslösungen
 mit ESV bei $N = 1000$

Keines des ESV kann über ein längeres Zeitintervall an der exakten Lösung "dran bleiben". Würden wir die Schrittkenzahl κ (2.63) zum KRKV auswerten, so kann man nur konstant bleibende Werte erkennen, so dass die in Abschnitt 2.4.2 beschriebene Methode nicht greift.

Wir vergleichen noch Rechnungen mit dem KRKV bei verschiedenen Schrittweiten $h = 5/N$.

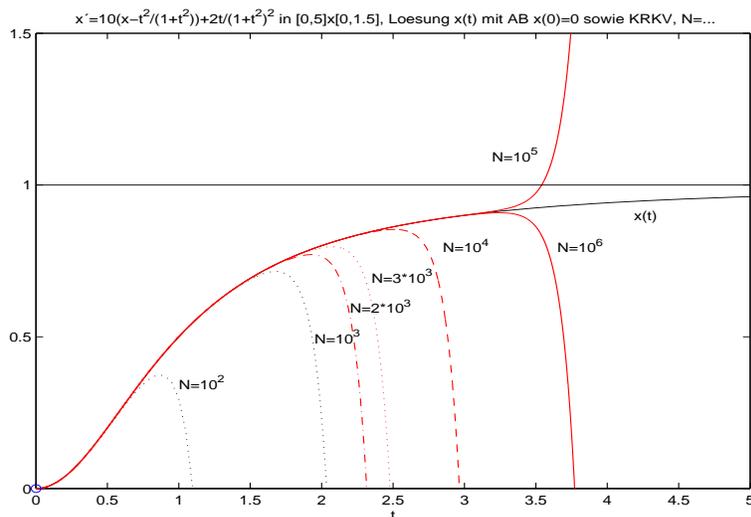


Abb. 2.17:

Datei *r_f1d.ps*,
 gDGL $x' = f(t, x)$, $t \in [0, 5]$
 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 Schrittweitensteuerung (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 *double*-Gleitpunktarithmetik 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.

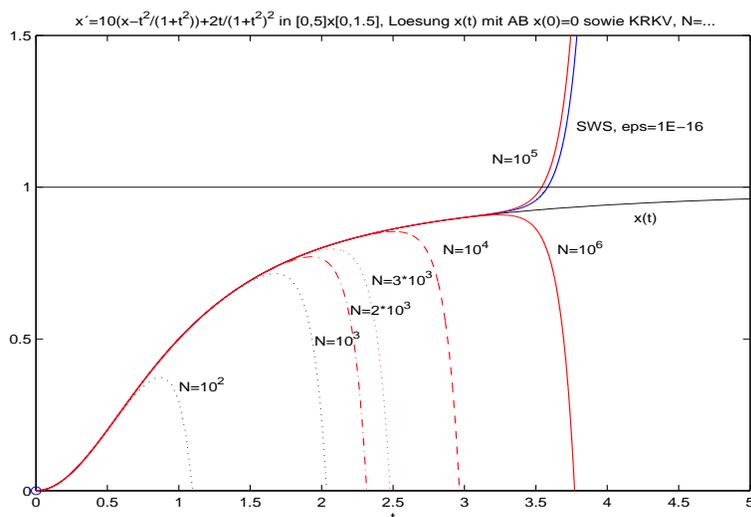


Abb. 2.18:

Datei *r_f1e.ps*,
 gDGL $x' = f(t, x)$, $t \in [0, 5]$
 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 $h_0 = 0.000\ 25$ erzielen wir das beste Ergebnis bezüglich des Aufwands an Funktionsauswertungen. Auch andere h_0 , wie z. B. $h_0 = 0.001$, die nach einigen anfänglichen Halbierungen/Verdopplungen auf die Teilintervalllänge $0.000\ 5$ 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 $0.000\ 5$ weiter zu rechnen.

Aber Rechnungen mit $h_0 \approx 0.000\ 25$, also knapp daneben, weichen schon deutlich ab.

In der ersten Hälfte des Bereiches $[t_0, T] = [0, 5]$ kann man bei $h_0 = 0.000\ 25$ fast mit konstanter Schrittweite rechnen. Die meisten Schrittweithalbierungen und damit verbunden die Nichtakzeptanz eines Schritts sind in der zweiten Hälfte.

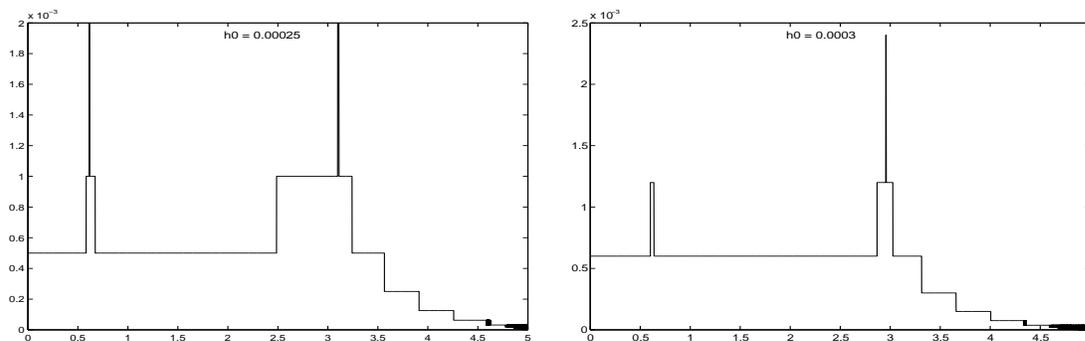
Trotz des strengen Abbruchkriteriums $|est| < 10^{-16}$ wird der Fehler der Näherungslösung dabei stetig schlechter.

Zum KRKV mit SWS erfolgen noch einige vergleichenden Rechnungen.

h_0	1. Schritt mit $2h_0$	Anzahl Schritte	Anzahl nicht akz.	$2h_{max}$	$2h_{min}$
0.000 4, nicht akz.	0.000 4	80 809	41 745	0.001 6	0.000 000 195
0.000 2, akzeptiert	0.000 4	80 809	41 744	0.001 6	0.000 000 195
0.000 225, akzeptiert	0.000 45	81 639	41 629	0.001 8	0.000 000 219
0.000 24, akzeptiert	0.000 48	53 010	21 907	0.001 92	0.000 000 937
0.000 245, akzeptiert	0.000 49	72 417	35 112	0.001 96	0.000 000 239
0.000 25, akzeptiert	0.000 5	34 522	9 457	0.002 0	0.000 001 953
0.000 255, akzeptiert	0.000 51	74 360	36 121	0.002 04	0.000 000 249
0.000 26, akzeptiert	0.000 52	58 304	25 821	0.002 08	0.000 000 507
0.000 275, akzeptiert	0.000 55	55 208	23 961	0.002 2	0.000 000 537
0.000 3, akzeptiert	0.000 6	56 113	24 968	0.002 4	0.000 000 585
0.01, nicht akz.	0.000 312 5	71 333	35 092	0.002 5	0.000 000 305
0.001, nicht akz.	0.000 5	34 522	9 459	0.002 0	0.000 001 953

Tab. 2.14: 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 $h_0 = 0.000\ 25$ ist. Die hohen Schrittzahlen der Rechnungen entstehen jedoch am Ende des rechten Teils des Intervalls $[t_0, t_1]$.



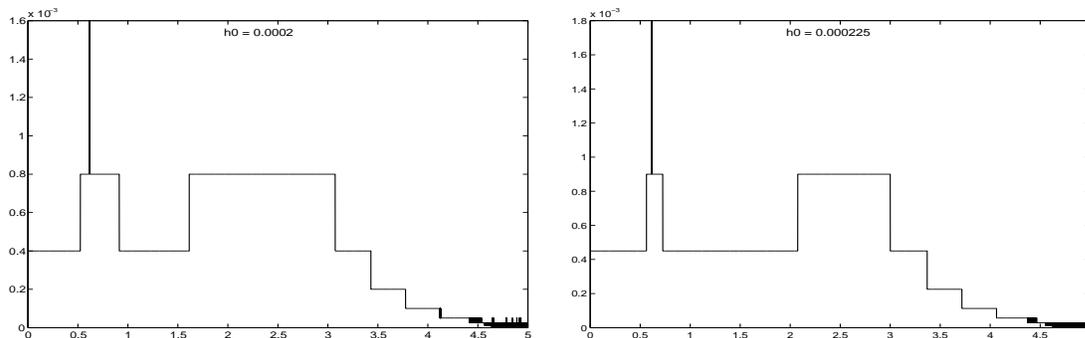


Abb. 2.19: Dateien *swh5.ps*, *swh3.ps*, *swh2.ps*, *swh6.ps* (v.l.o. n.r.u.),
 Schrittweitenverhalten $2h_i(t_i)$ beim KRKV mit SWS bei $\varepsilon = 10^{-16}$ und
 $h_0 = 0.000\ 25, 0.000\ 3, 0.000\ 2, 0.000\ 225$

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 $N = 10^6$, $h = 5 \cdot 10^{-6}$, 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	1.996800e-001 5.967e-015
2000	1.045500	5.222329e-001	5.222329e-001 6.928e-013	5.222329e-001 1.352e-012
3000	1.545500	7.048903e-001	7.048903e-001 1.038e-010	7.048903e-001 2.010e-010
4000	2.045500	8.071014e-001	8.071014e-001 1.540e-008	8.071013e-001 2.984e-008
5000	2.604000	8.714788e-001	8.714829e-001 4.104e-006	8.714708e-001 7.949e-006
6000	3.427000	9.215337e-001	9.369295e-001 1.540e-002	8.917112e-001 2.982e-002
7000	3.746500	9.334941e-001	1.309307e+000 3.758e-001	2.055207e-001 7.280e-001
8000	3.954250	9.398897e-001	3.940565e+000 3.001e+000	-4.872601e+000 5.812e+000
9000	4.079250	9.433116e-001	1.141670e+001 1.047e+001	-1.934428e+001 2.029e+001
10000	4.204250	9.464545e-001	3.750217e+001 3.656e+001	-6.986418e+001 7.081e+001
11000	4.293937	9.485541e-001	9.058056e+001 8.963e+001	-1.726697e+002 1.736e+002
12000	4.356437	9.499463e-001	1.684046e+002 1.675e+002	-3.234117e+002 3.244e+002
13000	4.418938	9.512837e-001	3.137977e+002 3.128e+002	-6.050664e+002 6.060e+002
14000	4.481438	9.525691e-001	5.854266e+002 5.845e+002	-1.131238e+003 1.132e+003
15000	4.543938	9.538050e-001	1.092895e+003 1.092e+003	-2.114256e+003 2.115e+003
16000	4.602125	9.549134e-001	1.954864e+003 1.954e+003	-3.783886e+003 3.785e+003
17000	4.635563	9.555327e-001	2.730698e+003 2.730e+003	-5.286853e+003 5.288e+003
18000	4.666813	9.561002e-001	3.732071e+003 3.731e+003	-7.226622e+003 7.228e+003
19000	4.698063	9.566570e-001	5.100786e+003 5.100e+003	-9.877971e+003 9.879e+003
20000	4.729313	9.572035e-001	6.971598e+003 6.971e+003	-1.350194e+004 1.350e+004
21000	4.760563	9.577399e-001	9.528694e+003 9.528e+003	-1.845531e+004 1.846e+004
22000	4.791625	9.582632e-001	1.299943e+004 1.300e+004	-2.517788e+004 2.518e+004
23000	4.821750	9.587617e-001	1.756901e+004 1.757e+004	-3.402943e+004 3.403e+004
24000	4.847531	9.591813e-001	2.273571e+004 2.273e+004	-4.403709e+004 4.404e+004
25000	4.869555	9.595348e-001	2.833684e+004 2.834e+004	-5.488754e+004 5.489e+004
26000	4.889078	9.598443e-001	3.444594e+004 3.444e+004	-6.672232e+004 6.672e+004
27000	4.906031	9.601103e-001	4.080963e+004 4.081e+004	-7.904693e+004 7.905e+004
28000	4.921703	9.603539e-001	4.773350e+004 4.773e+004	-9.246161e+004 9.246e+004
29000	4.937484	9.605970e-001	5.589322e+004 5.589e+004	-1.082664e+005 1.083e+005
30000	4.952406	9.608248e-001	6.488779e+004 6.489e+004	-1.256872e+005 1.257e+005
31000	4.964438	9.610070e-001	7.318352e+004 7.318e+004	-1.417617e+005 1.418e+005
32000	4.974961	9.611654e-001	8.130466e+004 8.130e+004	-1.574878e+005 1.575e+005
33000	4.984898	9.613141e-001	8.979930e+004 8.980e+004	-1.739466e+005 1.739e+005
34000	4.994336	9.614545e-001	9.868681e+004 9.869e+004	-1.911577e+005 1.912e+005
34522	5.000000	9.615385e-001	1.044378e+005 1.044e+005	-2.022995e+005 2.023e+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 bez. der Anzahl beim KRKV ohne SWS $4 \cdot 10^6$, bei KRKV mit SWS bei 35422 Schritten und 9457 nicht akzeptierten Schritten $12 \cdot 35422 + 8 \cdot 9457 \approx 4.9 \cdot 10^5$, 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 Schrittweiten 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 Prozeduraufruf

```
[ts, xs, hs, anz] = krkv1_sw('f1', 2.604, 5, 0.87147877549090, 0.0005, 10^(-16))
```

weiterrechnet, kommt man in der zweiten Intervallhälfte auf ein annehmbares Ergebnis mit nur einer Nichtakzeptanz der Schrittweite und damit Schrittweithalbierung 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 die Erkenntnis gewonnen, dass die näherungsweise Integration an Trajektorien, die instabile Gleichgewichtslagen sind, nur in sehr beschränkten Intervallen und bei hohen Genauigkeitsanforderungen möglich ist. ◀

3 Lineare Mehrschrittverfahren

3.1 Das Konzept der linearen Mehrschrittverfahren (MSV)

Das Konzept der ESV bestand 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 Zeitschritt nötig, die Funktion $f(t, x)$ an mehreren Zwischenstellen auszuwerten. 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.¹³

Dies illustrieren die folgenden ersten Beispiele zur Lösung des skalaren AWP

$$x' = f(t, x), \quad x(t_0) = x_0, \quad t \in [t_0, T]. \quad (3.1)$$

Die exakte Lösung werde wieder mit $x(t)$ bezeichnet. Wie früher seien zu einem vorgegebenen Zeitschritt $h = \Delta t$ die diskreten äquidistanten Zeiten $t_i = t_0 + i h$ definiert.

[3.1] Beispiele

Durch Integration der gDGL des AWP folgt

$$\int_{t_i}^{t_{i+1}} x'(t) dt = x(t_{i+1}) - x(t_i) = \int_{t_i}^{t_{i+1}} f(t, x(t)) dt. \quad (3.2)$$

Bezeichnen wir

$$F(t) = f(t, x(t)),$$

so erhalten wir die folgende exakte Formel

$$x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} F(t) dt. \quad (3.3)$$

Zu ihrer Integration erinnern wir uns an die Ideen zur Herleitung der Newton-Côtes-Formeln zur Integration.

(a) Wir ersetzen den Integranden $F(t)$ durch das lineare Interpolationspolynom zu den Punkten (t_{i-1}, F_{i-1}) und (t_i, F_i) , wobei $F_k = F(t_k)$, gemäß

$$\begin{aligned} p_1(t) &= \frac{t - t_{i-1}}{t_i - t_{i-1}} F_i + \frac{t - t_i}{t_{i-1} - t_i} F_{i-1} = \frac{t - t_{i-1}}{h} F_i - \frac{t - t_i}{h} F_{i-1} = \\ &= F_{i-1} + \frac{F_i - F_{i-1}}{t_i - t_{i-1}} (t - t_{i-1}) = F_i + \frac{F_{i-1} - F_i}{t_{i-1} - t_i} (t - t_i) = F_i + \frac{F_i - F_{i-1}}{h} (t - t_i). \end{aligned}$$

¹³Vergl. Vorlesung Num. Mathematik I

Setzen wir $p_1(t)$ anstelle von $F(t)$ in die Gleichung (3.3) ein und integrieren, so folgt

$$x(t_{i+1}) \approx x(t_i) + h \left[\frac{3}{2}f(t_i, x(t_i)) - \frac{1}{2}f(t_{i-1}, x(t_{i-1})) \right].$$

Dies führt zum **expliziten MSV**

$$\eta_{i+1} = \eta_i + h \left[\frac{3}{2}f(t_i, \eta_i) - \frac{1}{2}f(t_{i-1}, \eta_{i-1}) \right], \quad (3.4)$$

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

$$x(t_{i+1}) \approx x(t_i) + h \left[\frac{1}{2}f(t_{i+1}, x(t_{i+1})) + \frac{1}{2}f(t_i, x(t_i)) \right]$$

mit dem entsprechenden **impliziten MSV**

$$\eta_{i+1} = \eta_i + \frac{h}{2}[f(t_{i+1}, \eta_{i+1}) + f(t_i, \eta_i)], \quad (3.5)$$

implizit deshalb, weil die Gleichung nicht unmittelbar nach η_{i+1} aufgelöst werden kann.

(c) Ersetzen wir schließlich $F(t)$ durch das quadratische Interpolationspolynom $p_2(t)$ zu den Punkten (t_{i-2}, F_{i-2}) , (t_{i-1}, F_{i-1}) und (t_i, F_i) gemäß LAGRANGE¹⁴ oder NEWTON¹⁵, so folgen die verschiedenen Darstellungen

$$\begin{aligned} p_2(t) &= \frac{(t - t_{i-2})(t - t_{i-1})}{(t_i - t_{i-2})(t_i - t_{i-1})}F_i + \frac{(t - t_{i-2})(t - t_i)}{(t_{i-1} - t_{i-2})(t_{i-1} - t_i)}F_{i-1} + \frac{(t - t_{i-1})(t - t_i)}{(t_{i-2} - t_{i-1})(t_{i-2} - t_i)}F_{i-2} \\ &= \left[\frac{1}{2h^2}(t-t_i)^2 + \frac{3}{2h}(t-t_i) + 1 \right] F_i - \left[\frac{1}{h^2}(t-t_i)^2 + \frac{2}{h}(t-t_i) \right] F_{i-1} \\ &\quad + \left[\frac{1}{2h^2}(t-t_i)^2 + \frac{1}{2h}(t-t_i) \right] F_{i-2} \\ &= F_i + F[t_i, t_{i-1}](t - t_i) + F[t_i, t_{i-1}, t_{i-2}](t - t_i)(t - t_{i-1}) \\ &= F_i + \frac{1}{2h^2}[(t - t_i)^2(F_{i-2} - 2F_{i-1} + F_i) + h(t - t_i)(F_{i-2} - 4F_{i-1} + 3F_i)]. \end{aligned}$$

Daraus ergeben sich die Näherung

$$x(t_{i+1}) \approx x(t_i) + h \left[\frac{23}{12}f(t_i, x(t_i)) - \frac{16}{12}f(t_{i-1}, x(t_{i-1})) + \frac{5}{12}f(t_{i-2}, x(t_{i-2})) \right]$$

¹⁴Joseph Louis de Lagrange (1736-1813), französischer Mathematiker und Physiker, Arbeiten zur Variationsrechnung, Differentialgleichungen, Mechanik, Zahlentheorie, Analysis, Algebra

¹⁵Sir 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

sowie das explizite MSV

$$\eta_{i+1} = \eta_i + \frac{h}{12}[23f(t_i, \eta_i) - 16f(t_{i-1}, \eta_{i-1}) + 5f(t_{i-2}, \eta_{i-2})]. \quad (3.6)$$

(d) Nehmen wir in der Formel (3.2) als Intervall $[t_{i-1}, t_{i+1}]$ und ersetzen $F(t)$ durch das quadratische Interpolationspolynom $p_2(t)$ zu 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 folgt als Quadraturformel die SIMPSON¹⁶-Regel mit den Gewichten $\frac{1}{6}, \frac{4}{6}, \frac{1}{6}$ und 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

$$\eta_{i+1} = \eta_{i-1} + \frac{h}{3}[f(t_{i+1}, \eta_{i+1}) + 4f(t_i, \eta_i) + f(t_{i-1}, \eta_{i-1})]. \quad \blacktriangleleft \quad (3.7)$$

[3.2] Übung 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:

- Explizites oder implizites MSV,
- Anzahl der Stufen oder Schritte eines MSV,
- Wahl des Integrationsintervalls und der Knoten für die Interpolation.

Die oben erhaltenen Ansätze werden wie folgt verallgemeinert.

[3.3] Definition Ein Verfahren der allgemeinen einheitlichen Form

$$a_r \eta_{i+r} + a_{r-1} \eta_{i+r-1} + \dots + a_1 \eta_{i+1} + a_0 \eta_i = h (b_r f_{i+r} + \dots + b_1 f_{i+1} + b_0 f_i), \quad (3.8)$$

$$a_r = 1, \quad i \geq 0, \quad f_k = f(t_k, \eta_k),$$

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 maximalen und minimalen t -Index. Das ist die Anzahl der Teilintervalle, über die sich die Formel erstreckt.

In den Beispielen [3.1] haben die MSV in (a), (d) die Stufenzahl 2, in (b) 1 und in (c) 3. Somit beschreibt nach dieser Definition (a) ein explizites 2-stufiges MSV

$$\eta_{i+2} - \eta_{i+1} = h \left[\frac{3}{2} f(t_{i+1}, \eta_{i+1}) - \frac{1}{2} f(t_i, \eta_i) \right].$$

¹⁶Thomas Simpson (1710-1761), englischer Mathematiker

Die Formel (b) beschreibt ein implizites 1-Schritt-Verfahren und damit insbesondere also ein ESV

$$\eta_{i+1} - \eta_i = \frac{h}{2} [f(t_{i+1}, \eta_{i+1}) + f(t_i, \eta_i)].$$

Durch (c) ist ein explizites 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 [3.3] 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

$$\eta_{i+p} = \eta_{i-q} + h \sum_{j=0}^r \beta_j f_{i-j} = \eta_{i-q} + h (\beta_0 f_i + \beta_1 f_{i-1} + \dots + \beta_r f_{i-r}), \quad (3.9)$$

$$p \geq 0, \quad q \geq 0, \quad i \geq s = \max(r, q), \quad \eta_0, \eta_1, \dots, \eta_{s+p-1} \text{ gegeben.}$$

Eine dritte besondere implizite Form ist

$$\sum_{j=0}^r a_j \eta_{i+j} = h b_r f_{i+r}, \quad a_r, b_r \neq 0, \quad i \geq 0, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben.} \quad (3.10)$$

Eine vierte Möglichkeit der Notation von impliziten MSV ist

$$\eta_{m+1} = \sum_{j=1}^r \alpha_j \eta_{m+1-j} + h \sum_{j=0}^r \beta_j f_{m+1-j}, \quad m \geq r-1, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben,} \quad (3.11)$$

$$= h \beta_0 f(x_{m+1}, \eta_{m+1}) + \sum_{j=1}^r (\alpha_j \eta_{m+1-j} + h \beta_j f_{m+1-j}).$$

Die Bestimmung von η_{m+1} in (3.11) erfolgt meist iterativ bei vorgegebenem $\eta_{m+1}^{(0)}$ (dies z. B. aus einem expliziten MSV) mittels

$$\eta_{m+1}^{(k+1)} = h \beta_0 f(t_{m+1}, \eta_{m+1}^{(k)}) + C = g(\eta_{m+1}^{(k)}), \quad k = 0, 1, \dots \quad (3.12)$$

Bei der Herleitung der Formeln (3.8) - (3.11) 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

$$\nabla^j f_i = \nabla^{j-1} f_i - \nabla^{j-1} f_{i-1}, \quad \nabla^0 f_i = f_i$$

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 (3.8)- (3.11) kann man die grundlegenden Klassen von MSV bei entsprechender Wahl der Parameter p, q, r ableiten.

Formel (3.9)

- $p \geq 1$: 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 (3.10) : Rückwärtige Differentiationsformeln (engl. Backward Differentiation Formulas, BDF), steif-stabile GEAR-Methoden

Wir stellen im Folgenden daraus drei wichtige Klassen von MSV vor.

3.1.1 Adams-Bashforth-Verfahren

Adams¹⁷-Bashforth¹⁸-Verfahren sind explizite, bei denen in Verallgemeinerung zu den Beispielen [3.1] (a), (c) die Funktion $F(t)$ aus Gleichung (3.3) im Intervall $[t_i, t_{i+1}]$ durch das Interpolationspolynom zu den Knoten $t_{i-r}, \dots, t_{i-1}, t_i$ approximiert wird.

Wir suchen also für die Werte $r = 0, 1, 2, \dots$ die spezielle Darstellung

$$\eta_{i+1} = \eta_i + h \sum_{j=0}^r \beta_j f_{i-j} = \eta_i + h (\beta_0 f_i + \beta_1 f_{i-1} + \dots + \beta_r f_{i-r}) \quad (3.13)$$

als $(r + 1)$ -stufiges MSV. 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 $L_{i-j}(t)$. 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

$$L_k(t) = \prod_{\substack{l=0 \\ l \neq k}}^r \frac{t - t_l}{t_k - t_l}, \quad k = 0, 1, \dots, r.$$

¹⁷John Couch Adams (1819-1892), englischer Mathematiker und Astronom

¹⁸Francis Bashforth (1819-1912), englischer Physiker, Ballistik, Bashforth Chronograph

Die Integration von $p_r(t)$ anstatt von $F(t)$ liefert bei 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 damit die gesuchten Koeffizienten der MSV

$$\beta_k^{(r)} = \int_r^{r+1} \prod_{\substack{l=0 \\ l \neq k}}^r \frac{s - l}{k - l} ds, \quad k = 0, 1, \dots, r, \tag{3.14}$$

$$= \frac{(-1)^{r-k}}{k!(r-k)!} \int_r^{r+1} \prod_{\substack{l=0 \\ l \neq k}}^r (s - l) ds, \quad s = z + r, \quad z \in [0, 1]$$

$$= \frac{(-1)^{r-k}}{k!(r-k)!} \int_0^1 \prod_{\substack{l=0 \\ l \neq k}}^r (z + r - l) dz,$$

$$= \frac{(-1)^{r-k}}{k!(r-k)!} \int_0^1 \frac{1}{z + r - k} \prod_{l=0}^r (z + l) dz, \quad r - k = j,$$

$$\beta_{r-j}^{(r)} = \frac{(-1)^j}{j!(r-j)!} \int_0^1 \frac{1}{z + j} \prod_{l=0}^r (z + l) dz, \quad j = 0, 1, \dots, r. \tag{3.15}$$

Damit zeigen wir auch den Weg zur Vereinfachung und Auswertung der Formel.

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. 3.1: Koeffizienten der $(r + 1)$ -stufigen Adams-Bashforth-Verfahrenen, $r = 0, 1, \dots, 5$

Mit $f_k = f(t_k, \eta_k)$ ist durch

$$\eta_{i+1} - \eta_i = h(\beta_r^{(r)} f_i + \beta_{r-1}^{(r)} f_{i-1} + \dots + \beta_0^{(r)} f_{i-r}) \tag{3.16}$$

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 3.2).

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 (3.13) schreiben wir das erweiterte Koeffiziententableau mit $\beta_j = \beta_{r-j}^{(r)}$ und der Konstanten c im Fehlerglied wie folgt.

r	Ordnung = Stufen $n = r + 1$	Fehler- konstante $ch^n f^{(n+1)}$	K	j					
				0	1	2	3	4	5
				$K \beta_j$					
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. 3.2: Koeffizienten der n -stufigen Adams-Bashforth(n)-Verfahren n -ter Ordnung,

$$n = r + 1, \quad \eta_{i+1} - \eta_i = h \sum_{j=0}^r \beta_j f_{i-j}, \quad i = r, r + 1, \dots, \quad \eta_0, \eta_1, \dots, \eta_r \text{ gegeben}$$

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.

[3.4] Algorithmus $(r + 1)$ -stufiges Adams-Bashforth-Verfahren

S1 Initialisierung

Wähle Schrittzahl $N \in \mathbb{N}$ und Ordnung $r + 1$,
definiere $h = (T - t_0)/N$ und setze $\eta_0 = x_0$.

S2 Anlaufrechnung

Wähle 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 $\eta_{i+1} = \eta_i + h \sum_{j=0}^r \beta_j f(t_{i-j}, \eta_{i-j})$.

[3.5] Beispiel Wir betrachten das AWP aus Beispiel [2.20] (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 $x(t) = \frac{t^2}{1+t^2}$.

Nun wollen wir Adams-Bashforth-Verfahren mit RKV vergleichen. Dabei soll die Anlaufrechnung zum MSV mit den RKV erfolgen. Aus dem Abschnitt 2.2 nutzen wir das MATLAB-File `rkv.m` der vektoriellen Version des expliziten RKV.

Wir testen die 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, f1exv.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]';

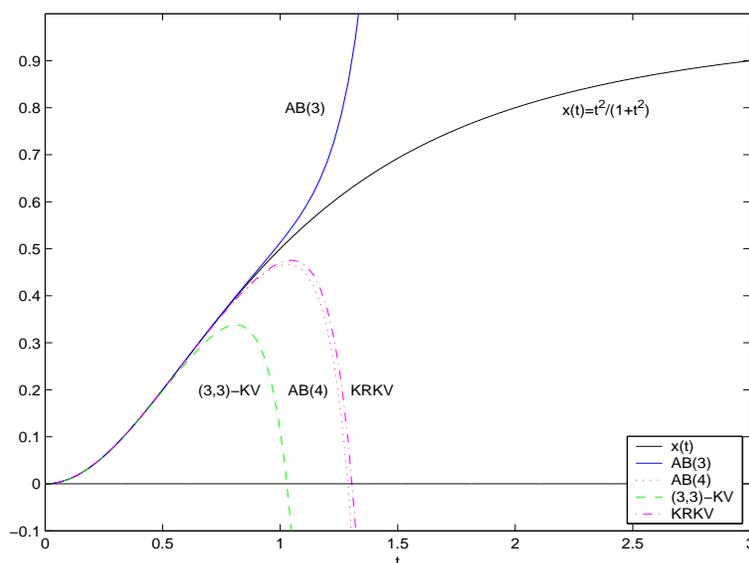
% 4-Schritt Adams-Bashforth -- Ordnung 4
beta2 = [ 55/24 -59/24 37/24 -9/24]';
```

```

% 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 = flexv(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. 3.1:**

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. ◀

3.1.2 Adams-Moulton-Verfahren

Die impliziten **Adams-Moulton**¹⁹-Verfahren erhält man, wenn man die Funktion $F(t)$ im Intervall $[t_{i-1}, t_i]$ durch das Interpolationspolynom zu 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

$$\eta_i = \eta_{i-1} + h \sum_{j=0}^r \beta_j f_{i-j} = \eta_{i-1} + h (\beta_0 f_i + \beta_1 f_{i-1} + \dots + \beta_r f_{i-r}). \quad (3.17)$$

Bei der Bestimmung der Koeffizienten verfährt man wie beim Adams-Bashforth-Verfahren. In dem Ansatz

$$\eta_i - \eta_{i-1} = h(\beta_r^{(r)} f_i + \beta_{r-1}^{(r)} f_{i-1} + \dots + \beta_0^{(r)} f_{i-r}), \quad f_k = f(t_k, \eta_k), \quad (3.18)$$

werden die Koeffizienten $\beta_k^{(r)}$ berechnet, einschließlich des impliziten ESV für $r = 0$ und $\beta_0^{(0)} = 1$.

$$\beta_k^{(r)} = \int_{r-1}^r \prod_{\substack{l=0 \\ l \neq k}}^r \frac{s-l}{k-l} ds, \quad k = 0, 1, \dots, r. \quad (3.19)$$

$\beta_k^{(r)}$	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. 3.3: Koeffizienten der s -stufigen Adams-Moulton-Verfahren, $s = 1, 1, 2, \dots, 5$

Als klassisches Adams-Moulton-Verfahren wird das Verfahren für $s = r = 4$ bezeichnet. Mit den Kriterien des Abschnitts 3.2 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 (3.17) und der Angabe der Stufenzahl s schreiben wir das erweiterte Koeffiziententableau mit $\beta_j = \beta_{r-j}^{(r)}$ und der Konstanten c im Fehlerglied wie folgt.

¹⁹Forest Ray Moulton (1872-1952), amerikanischer Physiker und Astronom, Differentialgleichungen und Ballistik

r	Stuf. s	Ordnung $n=r+1$	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	

Tab. 3.4: Koeffizienten der s -stufigen Adams-Moulton(n)-Verfahren,
 $s = \max(r, 1) = 1, 1, 2, \dots, 5$, 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

Adams-Moulton(1)-Verfahren ist das implizite Euler-Verfahren (IEV).

Als Schwierigkeit gegenüber den Adams-Bashforth-Verfahren ergibt sich hier, dass η_i nicht direkt aus Gleichung (3.17) 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 – etwa das Newton-Verfahren – auf die Gleichung $g(\xi) = 0$ anzuwenden, wobei

$$g(\xi) = \xi - \eta_{i-1} - h(\beta_0 f(t_i, \xi) + \beta_1 f_{i-1} + \dots + \beta_r f_{i-r}). \tag{3.20}$$

Zur numerischen Lösung wird das Adams-Bashforth-Verfahren [3.4] wie folgt modifiziert.

[3.6] Algorithmus s -stufiges Adams-Moulton-Verfahren

S1 Initialisierung

Wähle Schrittzahl $N \in \mathbb{N}$ und Ordnung $n = r + 1$, damit r, s ,
 definiere $h = (T - t_0)/N$ und setze $\eta_0 = x_0$.

S2 Anlaufrechnung

Wähle 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 $\eta_i = \eta_{i-1} + h \sum_{j=0}^r \beta_j f(t_{i-j}, \eta_{i-j})$

als Lösung von $g(\xi) = 0$ (z. B. mit dem Newton-Verfahren) mit $g(\cdot)$ gegeben durch (3.20). Dabei braucht man in der Regel einen Startwert $\xi_0 \approx \eta_i$ für die Iteration (z. B. $\xi_0 = \tilde{\eta}_i$ aus einem expliziten MSV oder als η_{i-1}).

[3.7] Beispiel Wir betrachten das Adams-Moulton(3)-Verfahren ($s = 2$) für die gDGL

$$x'(t) = t + x(t)^n. \quad (3.21)$$

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

$$\eta_i - \frac{5h}{12}\eta_i^n = C_i, \quad (3.22)$$

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 an zur Lösung der Gleichung (3.22). ◀

[3.8] Übung

Überlegen Sie sich, warum für $n = 2$ die andere Lösung der quadratischen Gleichung (3.22) nicht in Frage kommt. Entwickeln Sie hierzu η_i in eine Reihe bezüglich h .

3.1.3 Prädiktor-Korrektor-Verfahren

Die Lösung der Gleichung $g(\xi) = 0$ kann unter Umständen sehr aufwändig sein. Prädiktor-Korrektor-Verfahren²⁰ (PC-Verfahren) verkürzen diesen Weg, indem die Nullstelle von $g(\xi)$ nicht exakt, sondern nur näherungsweise berechnet wird.

Hierbei wird zunächst mit Hilfe eines expliziten MSV (3.8) ein Schätzwert $\bar{\eta}_{i+r}$ für η_{i+r} ermittelt (Prädiktor). Anschließend wird in der Funktion $g(\xi)$ des impliziten MSV (3.8) der Ausdruck $f(t_{i+r}, \xi)$ 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 PC-Verfahrens ist offensichtlich gleich der des Korrektors.

²⁰*praedicare* (lat.) = vorhersagen, *corrigere* (lat.) = verbessern

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 PC-Verfahrens nicht mehr verbessert werden kann.

Weiterhin sind nach jedem der beiden Schritte noch Nachbesserungen bzw. Modifikationen möglich. Offenbar hat man in den PC-Verfahren einen großen Spielraum.

Hier bietet sich als 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 des so veränderten Verfahrens gleich der des entsprechenden impliziten Adams-Moulton-Verfahrens ist (vergl. Abschnitt 3.2).

[3.9] Beispiel

(a) Wir benutzen zwei einfache ESV, die zugleich MSV sind.

Hier ist eine Anlaufrechnung nicht notwendig.

Prädiktor P : PZV, Adams-Bashforth(1)-Verfahren, Fehler $\mathcal{O}(h)$

$$\eta_{i+1}^{(0)} = \eta_i + hf(t_i, \eta_i)$$

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 $M = 1$ Korrektorschritt hat das PC-Verfahren die Fehlerordnung $\mathcal{O}(h^2)$.

(b) Für 3-stufige MSV der Fehlerordnung $\mathcal{O}(h^n)$ lautet ein Zeitschritt des PC-Verfahrens

P : 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 : 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 Lösungsgenauigkeit und Stabilitätsgebiete (vergl. Kapitel 4) betrifft, hat bez. der Adams-Verfahren 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 PC-Verfahren aus Teil (a) für das AWP

$$x'(t) = t - x(t), \quad x(0) = x_0 = 1, \quad t \in [0, 1] = [t_0, T]. \quad (3.23)$$

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 mit $M = 1, 2, \dots, 5$.

Wir wählen eine grobe Integrationsschrittweite h , um auch 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;

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\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 =

1.00000000000000
 0.84000000000000
 0.74480000000000
 0.70273600000000
 0.70424352000000
 0.74147968640000

xex =

1.00000000000000
 0.83746150615596
 0.74064009207128
 0.69762327218805
 0.69865792823444
 0.73575888234288

M=2

err1 = -3.2776e-003

eta =

1.00000000000000
 0.83600000000000
 0.73824800000000
 0.69468686400000
 0.69545385475200
 0.73248125318714

M=3

err1 = -2.3817e-003

eta =

1.00000000000000
 0.83640000000000
 0.73890248000000
 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.

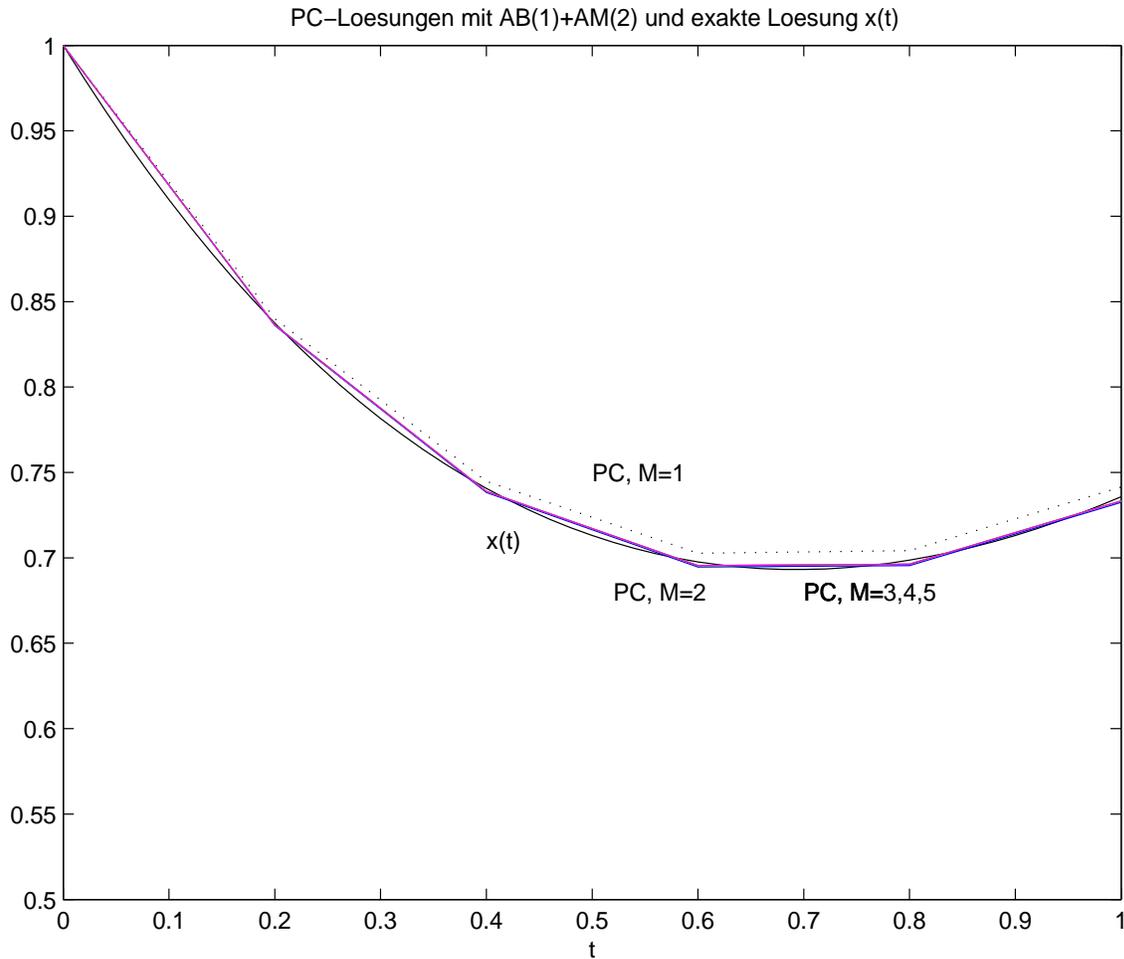


Abb. 3.2: Datei *pcv1.ps*,
 Lösung zum AWP $x' = 2e^{-t} + t - 1$, $x(0) = 1$, $t \in [0, 1]$, sowie
 Näherungslösungen aus dem PC-Verfahren mit M Korrektorschritten, $M = 1, 2, 3, 4, 5$

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	1.000000000	1.000000000	1.000000000	1.000000000
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 die Konsistenzbedingungen für PC-Verfahren 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:

$$\eta_{i+r} + \sum_{k=0}^{r-1} \alpha_k \eta_{i+k} = h \sum_{k=0}^{r-1} \beta_k f_{i+k}, \quad a_r = 1,$$

PC-Verfahren:

$$\begin{aligned} \eta_{i+r}^{(P)} &= - \sum_{k=0}^{r-1} a_k \eta_{i+k} + h \sum_{k=0}^{r-1} b_k f_{i+k}, \\ \eta_{i+r} &= - \sum_{k=0}^{r-1} \alpha_k \eta_{i+k} + h \beta_r f(t_{i+r}, \eta_{i+r}^{(P)}) + h \sum_{k=0}^{r-1} \beta_k f_{i+k}. \end{aligned}$$

Konsistenzordnung

Sei $z(t) = \Phi^{t,t_i} \eta_i$ sowie $z_j = z(t_j)$. Definiere

$$z_{i+r}^{(P)} = - \sum_{k=0}^{r-1} a_k z_{i+k} + h \sum_{k=0}^{r-1} b_k z'_{i+k}$$

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 [3.10])

$$\begin{aligned} \tau(t_i, \eta_i, h) &= \\ &= \frac{1}{h} (z_{i+r} + \alpha_{r-1} z_{i+r-1} + \dots + \alpha_0 z_i) - \left(\beta_r \left(z_{i+r}^{(P)} \right)' + \beta_{r-1} f_{i+r-1} + \dots + \beta_0 f_i \right) \\ &= \frac{1}{h} \left(z_{i+r} + \sum_{k=0}^{r-1} \alpha_k z_{i+k} \right) - \left[\beta_r \left(- \sum_{k=0}^{r-1} a_k z'_{i+k} + h \sum_{k=0}^{r-1} b_k z''_{i+k} \right) + \sum_{k=0}^{r-1} \beta_k z'_{i+k} \right] \\ &= \frac{1}{h} \left(z_{i+r} + \sum_{k=0}^{r-1} \alpha_k z_{i+k} \right) - \left(\sum_{k=0}^{r-1} (\beta_k - \beta_r a_k) z'_{i+k} + h \beta_r \sum_{k=0}^{r-1} b_k z''_{i+k} \right), \end{aligned}$$

der für entsprechende Parameter auszuwerten ist.

3.2 Die Ordnung linearer MSV

Ähnlich wie in der Definition [2.2] 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 gegebenes (t_i, η_i) mit

$$z(t) = \Phi^{t, t_i} \eta_i$$

die exakte Lösung der gDGL durch diesen Punkt bezeichnet.

[3.10] Definition (a) Der **lokale Diskretisierungsfehler** für das MSV (3.8) ist definiert durch

$$\begin{aligned} \tau(t_i, \eta_i, h) &= \frac{1}{h} [z(t_{i+r}) + a_{r-1}z(t_{i+r-1}) + \dots + a_0z(t_i)] \\ &\quad - [b_r f(t_{i+r}, z(t_{i+r})) + \dots + b_0 f(t_i, z(t_i))]. \end{aligned} \quad (3.24)$$

(b) Das MSV ist ein Verfahren **der Ordnung** $p > 0$, wenn

$$\tau(t_i, \eta_i, h) = \mathcal{O}(h^p) \quad \text{bzw.} \quad |\tau(t_i, \eta_i, h)| \leq Ch^p, \quad C \geq 0. \quad (3.25)$$

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

$$f(t_{i+k}, z(t_{i+k})) = z'(t_{i+k}) = z'(t_i) + kh z''(t_i) + \frac{(kh)^2}{2!} z'''(t_i) + \frac{(kh)^3}{3!} z''''(t_i) + \dots$$

folgt leicht die Entwicklung für den lokalen Diskretisierungsfehler

$$\tau(t_i, \eta_i, h) = \frac{1}{h} c_0 z(t_i) + c_1 z'(t_i) + c_2 h z''(t_i) + \dots = \sum_{k=0}^{\infty} c_k h^{k-1} z^{(k)}(t_i) \quad (3.26)$$

mit

$$c_0 = a_0 + a_1 + \dots + a_r, \quad (3.27)$$

$$c_1 = (a_1 + 2a_2 + \dots + ra_r) - (b_0 + b_1 + \dots + b_r) \quad (3.28)$$

sowie für $k \geq 2$

$$c_k = \frac{1}{k!} (a_1 + 2^k a_2 + \dots + r^k a_r) - \frac{1}{(k-1)!} (b_1 + 2^{k-1} b_2 + \dots + r^{k-1} b_r). \quad (3.29)$$

Hierbei ist $a_r = 1$.

Wir fassen zusammen.

[3.11] Satz Das MSV (3.8) hat genau dann (mindestens) die Konsistenzordnung p ($p \geq 1$), wenn mit den Definitionen (3.27), (3.28) und (3.29) für c_k gilt

$$c_0 = c_1 = \dots = c_p = 0. \tag{3.30}$$

Beim Entwerfen von MSV ist es nun ein Ziel, die Koeffizienten a_k, b_k so zu bestimmen, dass möglichst viele der Größen c_k verschwinden.

[3.12] Beispiele

(a) Ein explizites 3-Schritt-Verfahren der Form

$$\eta_{i+1} + a_1\eta_{i-1} = h[b_2f(t_i, \eta_i) + b_1f(t_{i-1}, \eta_{i-1}) + b_0f(t_{i-2}, \eta_{i-2})] \tag{3.31}$$

mit maximaler Ordnung soll ermittelt werden. Man beachte die Indexverschiebung bei der Notation der η_j und dass $a_3 = 1$ und $a_0 = a_2 = b_3 = 0$. Die maximale Ordnung 3 erhält man, wenn a_1, b_0, b_1 und b_2 so bestimmt werden, dass

$$\begin{aligned} 0 &= c_0 = a_1 + 1, \\ 0 &= c_1 = a_1 + 3 - b_0 - b_1 - b_2, \\ 0 &= 2c_2 = a_1 + 9 - 2b_1 - 4b_2, \\ 0 &= 6c_3 = a_1 + 27 - 3b_1 - 12b_2. \end{aligned}$$

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 3. Ordnung

$$\eta_{i+1} = \eta_{i-1} + \frac{h}{3}[7f(t_i, \eta_i) - 2f(t_{i-1}, \eta_{i-1}) + f(t_{i-2}, \eta_{i-2})]. \tag{3.32}$$

Das MSV gehört zu den expliziten Adams-Nyström-Formeln oder kurz Nyström-Formeln n -ter Ordnung. Dafür eignet sich besonders die spezielle Darstellung (3.9)

$$\eta_{i+1} = \eta_{i-1} + h \sum_{j=0}^r \beta_j f_{i-j}, \quad i = s = \max(r, 1), s + 1, \dots, r, \quad \eta_0, \eta_1, \dots, \eta_s \text{ gegeben.} \tag{3.33}$$

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. 3.5:
Koeffizienten der n -stufigen
Adams-Nyström(n)-Verfahren (3.33),
der Ordnung $n = s + 1$,
 $s = \max(r, 1)$, $r = 0, 2, 3$

(b) Zur Bestimmung einer impliziten 3-Schritt-Methode der Form

$$\eta_{i+1} + a_1\eta_{i-1} = h[b_3f(t_{i+1}, \eta_{i+1}) + b_2f(t_i, \eta_i) + b_1f(t_{i-1}, \eta_{i-1}) + b_0f(t_{i-2}, \eta_{i-2})] \quad (3.34)$$

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 mit dem Ergebnis

$$a_1 = -1, \quad b_3 = \frac{1}{3}, \quad b_2 = \frac{4}{3}, \quad b_1 = \frac{1}{3}, \quad b_0 = 0.$$

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

$$\frac{1}{2h}(\eta_{i+1} - \eta_{i-1}) = \frac{1}{6}f_{i+1} + \frac{4}{6}f_i + \frac{1}{6}f_{i-1}$$

erkennt man leicht die Ähnlichkeit zur Quadraturformel von Simpson.

(c) Das implizite 3-Schritt-Verfahren maximaler Ordnung mit dem Ansatz

$$\eta_{i+1} + a_2\eta_i + a_1\eta_{i-1} + a_0\eta_{i-2} = hb_3 f_{i+1}$$

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

$$11\eta_{i+1} - 18\eta_i + 9\eta_{i-1} - 2\eta_{i-2} = 6h f(t_{i+1}, \eta_{i+1}).$$

Dieses Verfahren gehört in die Klasse der BDF-Methoden (Gear-Methoden), da die linke Seite eine Rückwärts-Approximation der ersten Ableitung zur Zeit t_{i+1} darstellt.

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 als besondere implizite Form (3.10)

$$\sum_{j=0}^r a_j \eta_{i+j} = h b_r f_{i+r}, \quad a_r, b_r \neq 0, \quad i \geq 0, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ gegeben,}$$

schon kennen gelernt. Mit einer kleinen Indexverschiebung bei der Notation der η_j notieren wir die modifizierten Koeffizienten (ganze Zahlen) der impliziten n -stufigen BDF(n)-Verfahren n -ter Ordnung

$$\alpha_r \eta_{i+1} + \alpha_{r-1} \eta_i + \dots + \alpha_0 \eta_{i-r+1} = h \beta_r f_{i+1}, \quad i = r-1, r, \dots, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ geg.} \quad (3.35)$$

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. 3.6: Koeffizienten der n -stufigen BDF(n)-Verfahren ◀

[3.13] Übung (a) Zeigen Sie. Das 3-Schritt-Adams-Bashforth-Verfahren hat die Ordnung 3, das 3-Schritt-Adams-Moulton-Verfahren die Ordnung 4.

(b) Zeigen Sie durch Taylor-Reihenentwicklung, dass das PC-Verfahren aus Beispiel [3.9] (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 [3.9], [3.12] lassen sich verallgemeinern.

[3.14] Satz Das n -Schritt-Adams-Bashforth-Verfahren hat die Ordnung n . Das n -Schritt-Adams-Moulton-Verfahren hat die Ordnung $n+1$ (außer Fall $r = 0, s = n = 1$). Das zugehörige PC-Verfahren, zusammengesetzt aus dem n -Schritt-Adams-Bashforth-Prädiktor und dem n -Schritt-Adams-Moulton-Korrektor, hat dann die Ordnung $n+1$.

3.3 Homogene lineare Differenzgleichungen

Lineare Differenzgleichungen

$$\sum_{k=0}^r a_k \eta_{i+k} = c_i, \tag{3.36}$$

wie sie bei der Formulierung von MSV entstehen (vergl. Definition [3.3]), entwickeln eine "Eigendynamik". Dies ist bereits der Fall für $c_i = 0$ entsprechend der gDGL $x' = 0$.

Diese Differenzgleichungen sind der Inhalt des folgenden Abschnitts.

[3.15] Beispiele (a) Die Formel

$$\eta_{i+1} + 4\eta_i - 5\eta_{i-1} = h(4f_i + 2f_{i-1}) \tag{3.37}$$

beschreibt ein explizites 2-Schritt-Verfahren. Wir bestimmen mit (3.24) 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		

und damit offenbar ein oszillierendes, stark divergierendes Verhalten. Diese Instabilität wollen wir untersuchen. Zur Bestimmung der allgemeinen Lösung der Gleichung

$$\eta_{i+1} + 4\eta_i - 5\eta_{i-1} = 0 \quad (3.38)$$

mit Startwerten $\eta_0 = c_0$ und $\eta_1 = c_1$ bestimmen wir zunächst die Nullstellen des charakteristischen Polynoms

$$\rho(\xi) = \xi^2 + 4\xi - 5 = 0. \quad (3.39)$$

Diese sind $\xi_1 = 1$ und $\xi_2 = -5$. Man kann sich nun leicht überlegen, dass die allgemeine Lösung von (3.38) gegeben ist durch

$$\eta_i = \gamma_1 \xi_1^i + \gamma_2 \xi_2^i = \gamma_1 + \gamma_2 (-5)^i. \quad (3.40)$$

$\gamma_{1,2}$ 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

$$\eta_i = \left(c_0 - \frac{c_0 - c_1}{6} \right) 1^i + \frac{c_0 - c_1}{6} (-5)^i = 1 + \frac{\delta h}{6} - \frac{\delta h}{6} (-5)^i. \quad (3.41)$$

Offenbar wird das instabile Verhalten des MSV dadurch verursacht, dass $\rho(\xi)$ eine Nullstelle besitzt, deren Betrag größer als Eins ist.

Auf das charakteristische Polynom kommt man auch einfach durch Einsetzen des Ansatzes $\eta_i = \xi^i$, $\xi \neq 0$, in die Differenzgleichung (3.38).

(b) Betrachten wir die parameterabhängige homogene Differenzgleichung

$$\eta_{i+1} - 2c\eta_i + c^2\eta_{i-1} = 0, \quad c \in \mathbb{R}, \quad (3.42)$$

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

$$\rho(\xi) = \xi^2 - 2c\xi + c^2 = (\xi - c)^2 \quad (3.43)$$

und seiner doppelten Nullstelle $\xi_{1,2} = c$.

Die allgemeine Lösung von (3.43) ist gegeben durch

$$\eta_i = \gamma_1 \xi_1^i + \gamma_2 i \xi_2^i = \gamma_1 c^i + \gamma_2 i c^i. \quad (3.44)$$

Für Parameterwerte $|c| > 1$ liegt ein starkes Wachstum der η_i -Folge vor, für $|c| < 1$ tendieren $|\eta_i|$ 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 $\xi_1 \neq \xi_2$ ist, wie das für die Differenzgleichung $\eta_{i+1} - \eta_{i-1} = 0$ z. B. der Fall ist, wird die allgemeine Lösung $\eta_i = \gamma_1 \xi_1^i + \gamma_2 \xi_2^i$ stabil sein. ◀

Die Argumente in den Beispielen [3.15] für die Lösung der homogenen Differenzgleichung lassen sich als Satz verallgemeinern.

[3.16] Satz Hat das Polynom r -ten Grades

$$\rho(\xi) = a_r \xi^r + a_{r-1} \xi^{r-1} + \dots + a_1 \xi + a_0 \quad (3.45)$$

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**

$$\sum_{s=0}^r a_s \eta_{i+s} = 0 \quad (3.46)$$

darstellen in der Form

$$\eta_i = \sum_{s=1}^r \gamma_s \xi_s^i. \quad (3.47)$$

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. Wichtig für unsere Zwecke ist es, dass die Lösungen der homogenen Gleichung (3.46) langsamer als linear bezüglich des Index anwachsen, dass also die **Wachstumsbeschränkung**

$$\lim_{i \rightarrow \infty} \frac{\eta_i}{i} = 0 \quad (3.48)$$

gilt. Hierzu haben wir das folgende wichtige Resultat.

[3.17] Satz Die Wachstumsbeschränkung (3.48) 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 (3.45) 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.

[3.18] Beispiele (a) Das MSV des Beispiels [3.15] (a) erfüllt nicht das Stabilitätskriterium des Satzes [3.17], da mit $\xi_2 = -5$ eine Nullstelle gegeben ist mit Betrag größer als 1.

(b) Für das Beispiel [3.12] (a) ist $\rho(\lambda)$ 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 [3.12] (c) ist

$$\rho(\xi) = 11\xi^3 - 18\xi^2 + 9\xi - 2.$$

Die Nullstellen sind $\xi_1 = 1$ sowie $\xi_{2,3} = (7 \pm \sqrt{39}i)/22 = 0.318\,181\,818 \pm 0.283\,863\,545i$. Das Stabilitätskriterium ist erfüllt. Übrigens gilt dies für alle BDF-Verfahren.

3.4 Konsistenz und Konvergenz von MSV

Gegeben sei das r -stufige lineare MSV ($r \geq 1$)

$$\sum_{k=0}^r a_k \eta_{i+k} = h \sum_{k=0}^r b_k f_{i+k}, \quad f_{i+k} = f(t_{i+k}, \eta_{i+k}), \quad a_r = 1. \quad (3.49)$$

[3.19] **Definition** Das **erste** und das **zweite charakteristische (erzeugende) Polynom** des MSV (3.49) sind definiert durch

$$\rho(\xi) = \sum_{k=0}^r a_k \xi^k, \quad \sigma(\xi) = \sum_{k=0}^r b_k \xi^k. \quad (3.50)$$

Das Polynom $\rho(\xi)$, das auch in der Formel (3.45) und im Stabilitätskriterium (Satz [3.17]) erscheint, nimmt also eine zentrale Stellung ein. Es ist das wichtigere von beiden, da wegen $h \approx 0$ das Polynom $\sigma(\xi)$ sozusagen entfallen kann.

[3.20] **Definition** (a) Das MSV (3.49) heißt **konsistent**, falls es mindestens die Ordnung 1 hat, falls also nach Satz [3.11] gilt

$$\begin{aligned} 0 &= c_0 = a_0 + a_1 + \dots + a_r, \\ 0 &= c_1 = a_1 + 2a_2 + \dots + ra_r - (b_0 + b_1 + \dots + b_r). \end{aligned} \quad (3.51)$$

Man überlegt sich leicht, dass diese Konsistenzbedingung äquivalent ist zur

Konsistenzbedingung: $\rho(1) = 0, \quad \rho'(1) - \sigma(1) = 0.$

(3.52)

Ein weitere dazu äquivalente Aussage ist, dass das MSV (3.49) exakt ist für die AWP

$$\begin{aligned} x' &= 0, \quad x(0) = c, \\ x' &= 1, \quad x(0) = c, \end{aligned} \quad (3.53)$$

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 2.3) 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

$$\lim_{N \rightarrow \infty} \eta_N = x(t). \quad (3.54)$$

Das heißt, dass der **globale Diskretisierungsfehler** oder 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.

Es geht bei der Konvergenz um das Verhalten der Näherungslösung aus dem MSV 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.

[3.21] Satz P. Henrici²¹ 1962

Betrachtet wird zum AWP $x' = f(t, x)$, $x(t_0) = x_0$, das MSV (3.49).

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 [3.17].

Das Stabilitätskriterium ist auch hinreichend für die Konvergenz, falls zusätzlich das MSV mindestens die Konsistenzordnung 1 (vergl. Definition [3.20]) besitzt.

[3.22] Satz Ein MSV der Form (3.49) ist genau dann konvergent für beliebige AWP, wenn es konsistent ist und das Stabilitätskriterium des Satzes [3.17] erfüllt.

Beweis: Hier zeigen wir nur die eine Richtung des Beweises, nämlich dass aus der Konvergenz des MSV sowohl seine Konsistenz als auch das Stabilitätskriterium folgen. Zum Nachweis ist eine Fallunterscheidung bezüglich der Nullstellen des charakteristischen Polynoms $\rho(\xi)$ sinnvoll. Von all diesen Fällen betrachten wir ausführlich nur die ersten drei.

(1) Konvergenz des MSV \Rightarrow Konsistenz

Die Konvergenz haben wir in der Definition [3.20] (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 zur gDGL $x' = f(t, x)$ die AWP

$$\begin{aligned} x' &= 0, & x(0) &= 1, \\ x' &= 1, & x(0) &= 0. \end{aligned}$$

²¹Peter Henrici (1923-1987), schweizer Mathematiker, Theorie und Numerik gDGL, Approximation, komplexe Analysis

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 (3.51).

(2) Konvergenz des MSV \Rightarrow Stabilitätskriterium

Hier brauchen wir das AWP $x' = 0$, $x(0) = 0$, mit seiner exakten Lösung $x(t) \equiv 0$ sowie das charakteristische Polynom und seine Ableitung

$$\rho(\xi) = \sum_{k=0}^r a_k \xi^k, \quad \rho'(\xi) = \sum_{k=1}^r k a_k \xi^{k-1}.$$

Seien $\xi_1, \xi_2, \dots, \xi_r$ die r komplexen Nullstellen von $\rho(\xi)$, also

$$\begin{aligned} \rho(\xi_j) &= \sum_{k=0}^r a_k \xi_j^k = 0, \quad j = 1, 2, \dots, r, \quad a_r = 1, \\ \xi_j^r &= - \sum_{k=0}^{r-1} a_k \xi_j^k. \end{aligned}$$

1. Fall: 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

$$\lim_{h \rightarrow 0} v_i = \lim_{h \rightarrow 0} \sum_{j=1}^r h \xi_j^i = 0 = x(0)$$

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. $\xi_1 = re^{i\varphi}$ 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. Es gilt

$$\begin{aligned} \eta_0 &= \sum_{s=1}^r \gamma_s \xi_s^0 = \sum_{s=1}^r \gamma_s, \\ \eta_1 &= \sum_{s=1}^r \gamma_s \xi_s^1 = \gamma_1 \xi_1 + \gamma_2 \xi_2 + \sum_{s=3}^r \gamma_s \xi_s \\ &= \gamma_1 r e^{i\varphi} + \gamma_2 r e^{-i\varphi} + \sum_{s=3}^r \gamma_s \xi_s \\ &= (\gamma_1 + \gamma_2) r \cos(\varphi) + i(\gamma_1 - \gamma_2) r \sin(\varphi) + \sum_{s=3}^r \gamma_s \xi_s, \end{aligned}$$

so dass $\gamma_1 = \gamma_2$ folgt.

Es gilt natürlich $\xi_1 + \xi_2 = 2\Re \xi_1 \in \mathbb{R}$, wie auch allgemein die Beziehung $\xi_1^l + \xi_2^l = \xi_1^l + \overline{\xi_1^l} = \xi_1^l + \overline{\xi_1^l} \in \mathbb{R}$ 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 analog.

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

$$\lim_{h \rightarrow 0} v_i = \lim_{h \rightarrow 0} h \left(\xi_1^i + i\xi_1^i + \sum_{j=3}^r \xi_j^i \right) = 0 = x(0)$$

erfüllt. Die Lösung des MSV in der neuen Darstellung

$$\eta_i = h\xi_1^i + hi\xi_1^i + \sum_{j=3}^r h\xi_j^i, \quad i = r, r+1, \dots,$$

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. \square

Wir kommen nun zur Fehleranalyse für gewisse explizite lineare MSV.

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 [3.10] 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, wobei eine kleine Indexverschiebung vorgenommen wurde.

$$\begin{aligned} \tau(t_i, \eta_i, h) &= \frac{1}{h} [z(t_{i+r}) + a_{r-1}z(t_{i+r-1}) + \dots + a_0z(t_i)] \\ &\quad - [b_{r-1}f(t_{i+r-1}, z(t_{i+r-1})) + \dots + b_0f(t_i, z(t_i))] \\ &= \frac{1}{h} [z(t_{i+1}) + a_{r-1}z(t_i) + \dots + a_0z(t_{i-r+1})] \\ &\quad - [b_{r-1}f(t_i, z(t_i)) + \dots + b_0f(t_{i-r+1}, z(t_{i-r+1}))] \\ &= \frac{1}{h} \left(z_{i+1} + \sum_{k=0}^{r-1} a_k z_{i-r+1+k} \right) - \sum_{k=0}^{r-1} b_k f_{i-r+1+k} \\ &= \frac{1}{h} \left[z_{i+1} - \sum_{k=0}^{r-1} (-a_k z_{i-r+1+k} + h b_k f_{i-r+1+k}) \right], \\ \tilde{\tau}_{i+1} &= \frac{1}{h} \left[x(t_{i+1}) - \sum_{k=0}^{r-1} (-a_k x(t_{i-r+1+k}) + h b_k f(t_{i-r+1+k}, x_{i-r+1+k})) \right]. \end{aligned} \quad (3.55)$$

[3.23] Satz Für das explizite MSV gelte

$$a_k \leq 0, \quad k = 0, 1, \dots, r-1, \quad a_r = 1. \quad (3.56)$$

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 $L \geq 0$

$$|f(t, x_1) - f(t, x_2)| \leq L |x_1 - x_2|,$$

so gilt die Abschätzung

$$|e_i| \leq \left(G + \frac{D}{LB} \right) e^{LB(t_i - t_0)}. \quad (3.57)$$

Beweis: Zuerst wollen wir erläutern, dass hinter den Beziehungen (3.56) das Stabilitätskriterium sich 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, konvex und haben in $\xi = 1$ einen "echten" Schnittpunkt. Rechnerisch führt man den Nachweis mit $\rho'(1) \neq 0$.

$$\begin{aligned} \rho'(\xi) &= r \xi^{r-1} - \sum_{k=1}^{r-1} k \tilde{a}_k \xi^{k-1}, \\ \rho'(1) &= r - \sum_{k=1}^{r-1} k \tilde{a}_k = \left(1 - \sum_{k=1}^{r-1} \tilde{a}_k\right) + \left(1 - \sum_{k=2}^{r-1} \tilde{a}_k\right) + \dots + (1 - \tilde{a}_{r-1}) + 1 \\ &= \tilde{a}_0 + (\tilde{a}_0 + \tilde{a}_1) + \dots + \sum_{k=0}^{r-2} \tilde{a}_k + 1 \\ &\geq 1. \end{aligned}$$

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.

Bleibt noch nachzuweisen, dass Nullstellen auf dem Einheitskreis nicht doppelt auftreten.

Wenn man eine solche i. Allg. komplexe Nullstelle $\xi_s = e^{i\varphi}$, $|\xi_s| = 1$, hat, dann ist auch $\overline{\xi_s} = e^{-i\varphi}$ eine Nullstelle.

Für $\xi = 1$ haben wir das schon getan 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 (3.56) wenden wir uns nun dem Beweis zu. Aus den Beziehungen (3.49) ($j = i - r + 1$)

$$\begin{aligned}\sum_{k=0}^r a_k \eta_{i+k} &= h \sum_{k=0}^r b_k f_{i+k}, \quad a_r = 1, \quad b_r = 0, \\ \eta_{i+r} &= \sum_{k=0}^{r-1} (-a_k \eta_{i+k} + h b_k f_{i+k}), \\ \eta_{i+1} &= \sum_{k=0}^{r-1} (-a_k \eta_{j+k} + h b_k f_{j+k}) \quad (\text{Indexverschiebung})\end{aligned}$$

und (3.54)

$$x(t_{i+1}) = \sum_{k=0}^{r-1} [-a_k x(t_{j+k}) + h b_k f(t_{j+k}, x(t_{j+k}))] + h \tilde{\tau}_{i+1}$$

folgt durch Subtraktion, Anwenden der Dreiecksungleichung und Ausnutzung der Lipschitz-Stetigkeit von f für $i \geq r - 1$

$$|e_{i+1}| = |\eta_{i+1} - x(t_{i+1})| \leq \sum_{k=0}^{r-1} (|a_k| + hL |b_k|) |e_{j+k}| + h |\tilde{\tau}_{i+1}|. \quad (3.58)$$

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 (3.51) folgt $\sum_{k=0}^{r-1} a_k = -a_r = -1$ und damit

$$C = \sum_{k=0}^{r-1} |a_k| + hL \sum_{k=0}^{r-1} |b_k| = 1 + hLB \geq 1. \quad (3.59)$$

Eingesetzt in (3.58) 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

$$|e_{r+s}| \leq GC^{s+1} + hD \sum_{k=0}^s C^k, \quad (3.60)$$

bzw. mit $r \geq 1$ und $C \geq 1$ die gröbere Abschätzung

$$|e_{r+s}| \leq GC^{s+r} + hD \sum_{k=0}^{s+r-1} C^k. \quad (3.61)$$

Mit $i = r + s$ folgt wegen (3.59)

$$\begin{aligned}
|e_i| &\leq GC^i + hD \sum_{k=0}^{i-1} C^k \\
&= GC^i + hD \frac{C^i - 1}{C - 1} \\
&= G(1 + hLB)^i + \frac{D}{LB} [(1 + hLB)^i - 1], \quad L > 0 \\
&\leq Ge^{ihLB} + \frac{D}{LB} (e^{ihLB} - 1) \\
&\leq \left(G + \frac{D}{LB} \right) e^{ihLB} \\
&= \left(G + \frac{D}{LB} \right) e^{LB(t_i - t_0)}
\end{aligned}$$

die Behauptung des Satzes. □

[3.24] Bemerkungen

- (a) Die Adams-Bashforth-Formeln erfüllen die Voraussetzung (3.56) des Satzes [3.23].
 (b) Eine ähnliche Abschätzung wie die des Satzes [3.23] kann auch für allgemeinere und insbesondere für implizite MSV durchgeführt werden.
 (c) Einige Ungleichungen im Satz [3.23] stellen eine Vergrößerung der Abschätzung dar.
 (d) Um in jedem Fall die Konvergenz zu sichern, muss $D = \mathcal{O}(h^p)$, $p > 0$, und bezüglich der AB

$$\lim_{h \rightarrow 0} G = 0 \text{ bzw. } G \leq c_1 h^q, \quad q > 0, \quad c_1 \geq 0, \quad (3.62)$$

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 [2.12] für ESV

$$|e_i| \leq \frac{c h^p}{LB} e^{LB(t_i - t_0)} = \mathcal{O}(h^p). \quad (3.63)$$

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

$$\begin{aligned} |e_{r+s}| &\leq GC^{s+1} + hD \sum_{k=0}^s C^k = G + (s+1)hD \leq G + (s+r)hD, \\ |e_i| &\leq G + ihD = G + D(t_i - t_0). \end{aligned} \quad (3.64)$$

Das bedeutet für $G = \mathcal{O}(h^p)$ und $D = \mathcal{O}(h^p)$ die Konvergenzordnung p des MSV.

4 Stabilität von Ein- und Mehrschrittverfahren

4.1 Absolute Stabilität

Während der Abschnitt 3.3 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 später folgenden Untersuchungen beziehen sich 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 $x' = x$, $x(0) = 1$, mit seiner exponentiell wachsenden Lösung $x(t) = e^t$ die Näherung aus einem Schrittverfahren vergleicht, so kann 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 [1.11] (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 monotone exakte Lösung ist eine oszillierende Näherungslösung wenig geeignet. 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 [1.11] (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 bei Rechnungen in Gleitpunktarithmetik, die damit Rundungseffekten und verschiedenen Störungen ausgesetzt sind, können Lösungen erzeugen, die sich schnell von der exakten Trajektorie entfernen.

Für die gDGL $x' = t \cos(x)$ in den Beispielen [2.17] (a) bzw. [2.20] (b) erkennen wir in $x(t) \equiv \frac{\pi}{2}$ 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. 1.1 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 [2.20] (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 kommen hinzu,

- wenn das Richtungsfeld sehr steile Anstiege enthält wie bei $x' = 1/(x + 10^{-p})$, $p \gg 1$ (vergl. Beispiel [2.17] (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 [1.8] (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 [1.8] (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 [1.8] (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)

$$x' = \lambda x, \quad t \geq t_0, \quad x(t_0) = x_0 \in \mathbb{R}, \quad \lambda \in \mathbb{C}, \quad (4.1)$$

im vektoriellen Fall das SysgDGL

$$\mathbf{x}' = A \mathbf{x}, \quad t \geq t_0, \quad \mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbb{R}^n, \quad \text{Matrix } A \text{ reell, regulär, diagonalisierbar.} \quad (4.2)$$

Mit der Lösung $x(t) = x_0 e^{\lambda(t-t_0)}$ der Test-DGL (4.1) sieht man, dass die Größe $\Re \lambda$ darüber entscheidet, ob für wachsendes t der Betrag der Lösung $x(t)$ fällt oder wächst.

[4.1] Definition

- (a) Die Test-DGL (4.1) heißt stabil, falls $\Re \lambda \leq 0$.
- (b) Die Test-DGL (4.1) 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

$$\lim_{t \rightarrow \infty} (\tilde{x}(t) - x(t)) = 0 \quad (4.3)$$

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 der Parameter λ aus negativen komplexen Halbebene \mathbb{C}^- genommen.

Natürlich wird man von Schrittverfahren adäquates Wachstumsverhalten verlangen bzw. untersuchen, welche Bedingungen dafür zu erfüllen sind.

[4.2] Beispiel Betrachten wir das lineare Testproblem

$$x' = \lambda x, \quad x(0) = 1, \quad \lambda \in \mathbb{C}, \quad (4.4)$$

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

$$\eta_{i+1} = \eta_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) = F(h\lambda) \eta_i \quad (4.5)$$

mit dem Polynom (Wachstums- oder Übergangsfunktion von $i \rightarrow (i+1)$)

$$F(z) = 1 + z + \frac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4, \quad z = h\lambda \in \mathbb{C}. \quad (4.6)$$

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

$$|F(z)| < 1. \quad (4.7)$$

Wegen $|F(z)| \rightarrow \infty$ für $|z| \rightarrow \infty$ ist das aber nur möglich, wenn $|z|$ und damit die Schrittweite h hinreichend klein ist. ◀

4.1.1 Stabilitätsgebiet von ESV

Das Beispiel [4.2] motiviert die folgende Definition.

[4.3] Definition (a) Führt ein ESV für das Problem (4.4) auf eine Rekursion der Form

$$\eta_{i+1} = F(z) \eta_i, \quad z = h\lambda, \quad (4.8)$$

so heißt die Menge

$$\mathcal{E} = \{z \in \mathbb{C} : |F(z)| < 1\}$$

das zum ESV gehörende **Gebiet der absoluten Stabilität**.

Wegen $F(0) = 1$ liegt $z = 0$ immer auf der Grenze des Stabilitätsgebiets.

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

$$\mathbb{C}^- = \{z \in \mathbb{C} : \Re z < 0\} \quad (4.9)$$

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 etwas schwächere Bedingung auch als A-Stabilität verwendet.

Entsprechend dem Beispiel [4.2] 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 $\lambda < 0$.

[4.4] Beispiele

(a) Für das PZV ist

$$\eta_{i+1} = \eta_i + hf(\eta_i) = (1 + h\lambda)\eta_i, \quad i = 0, 1, \dots, \eta_0 \text{ gegeben}, \quad (4.10)$$

also

$$F(z) = 1 + z. \quad (4.11)$$

Das Gebiet der absoluten Stabilität ist in der Gaußschen Zahlenebene wegen $1 = |1+z| = |z - (-1)|$ der Kreis um den Mittelpunkt $z_0 = -1$ und mit dem Radius $r = 1$.

Zu reellem $\lambda < 0$ ist die Ableitung 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 [1.11]).

Zu komplexem $\lambda = s + i\omega$ mit $\Re \lambda = s < 0$ könnte man zwei Abschätzungen vornehmen.

$$\begin{aligned} |h\lambda| - 1 &\leq ||h\lambda| - 1| \leq |1 + h\lambda| < 1, \\ |h\lambda| &< 2, \\ h &< \frac{2}{|\lambda|}, \end{aligned}$$

$$1 - |h\lambda| \leq |1 - |h\lambda|| \leq |1 + h\lambda| < 1, \quad \text{falls } |h\lambda| < 1, \quad \text{d. h. } h < \frac{1}{|\lambda|}.$$

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, im Komplexen zu arbeiten. Dann erhält man die Beziehungen

$$\begin{aligned} |1 + h\lambda| &= |1 + h(s + i\omega)| = |1 + hs + ih\omega| < 1, \\ (1 + hs)^2 + (h\omega)^2 &= 1 + 2hs + h^2s^2 + h^2\omega^2 < 1, \\ 2s + h(s^2 + \omega^2) &< 0, \\ h|\lambda|^2 &< -2 \Re\lambda, \\ h &< \frac{2 |\Re\lambda|}{|\lambda|^2} = -\frac{2s}{s^2 + \omega^2}. \end{aligned} \tag{4.12}$$

Die reelle Version ist als Sonderfall enthalten.

Betrachten wir uns die Schrittweitenfunktion $h = h(\lambda) = h(s, \omega)$ etwas genauer, indem wir eine der unabhängigen Variablen konstant lassen.

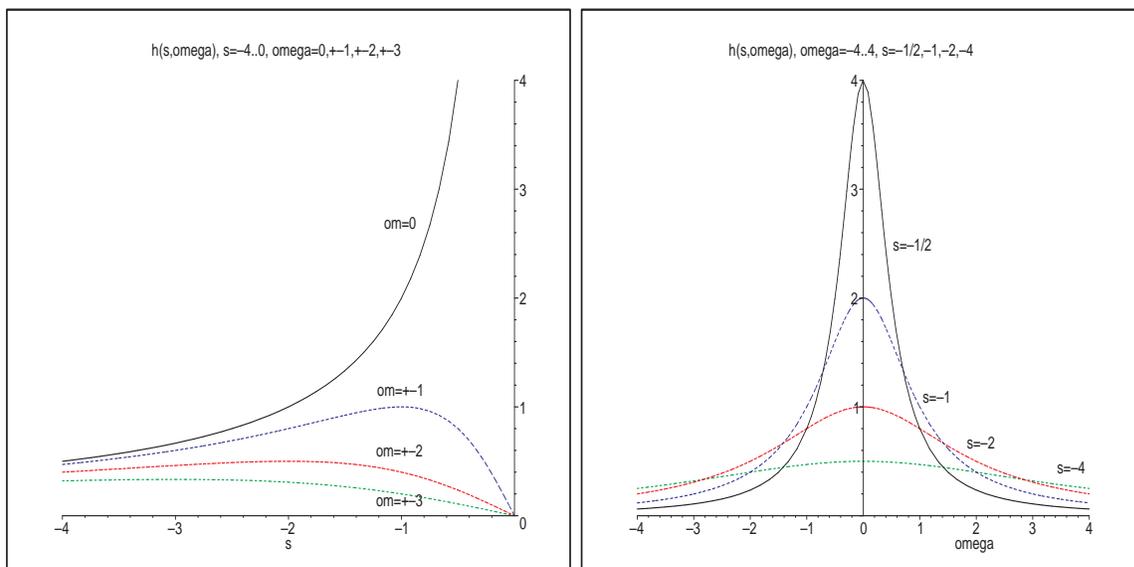


Abb. 4.1: Dateien *schr01.ps*, *schr02.ps*, Schrittweitenfunktion $h(s, \omega)$

Wir erkennen folgende Eigenschaften von $h(s, \omega)$.

- 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 $s \ll 0$, dann gibt es eine starke Einschränkung für h bei allen ω .
- Falls $\omega \approx 0$, dann ist mit $s \lesssim 0$ 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 ist

$$z = -1 + e^{i\varphi} = -1 + \cos(\varphi) + i \sin(\varphi), \quad \varphi \in [0, 2\pi),$$

was genau den Einheitskreis 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

$$\begin{aligned} x(t) &= e^{\lambda t} = e^{-t}(\cos(t) + i \sin(t)) = \Re x(t) + i \Im x(t) = x_1(t) + i x_2(t), \\ x_1(t) &= e^{-t} \cos(t), \quad x_1(0) = 1, \quad x_2(t) = e^{-t} \sin(t), \quad x_2(0) = 0. \end{aligned}$$

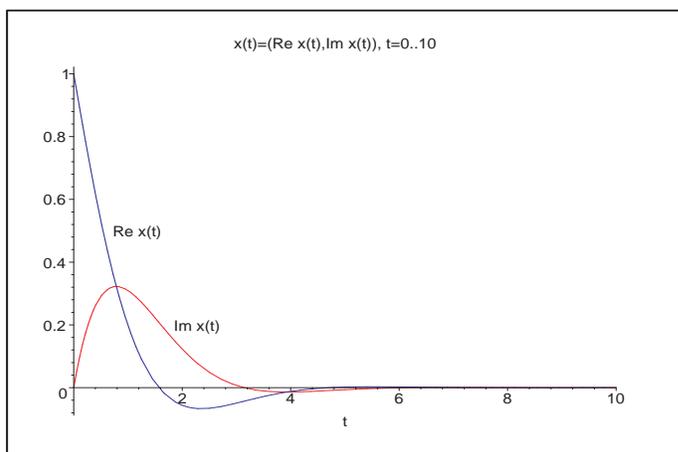


Abb. 4.2:

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);
```

$$\begin{aligned} \lambda &:= -1 + I \\ dgl &:= \left(\frac{d}{dt} x(t\sim) \right) + (1 - I)x(t\sim) \\ x(t\sim) &= _C1 e^{((-1+I)t\sim)} \\ loes &:= x(t\sim) = e^{((-1+I)t\sim)} \\ x1 &:= t\sim -> e^{(-t\sim)} \cos(t\sim) \\ x2 &:= t\sim -> e^{(-t\sim)} \sin(t\sim) \end{aligned}$$

Wollen wir Richtungsfelder zur gDGL wie in Abschnitt 1.1 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

$$(x_1(t) + \iota x_2(t))' = (-1 + \iota)(x_1(t) + \iota x_2(t)) \quad \text{mit } x_1(t) = \Re x(t), \quad x_2(t) = \Im x(t),$$

in ein System von zwei gDGL 1. Ordnung

$$\begin{aligned} x_1' + x_1 + x_2 &= 0, & x_1(0) &= 1, \\ x_2' - x_1 + x_2 &= 0, & x_2(0) &= 0. \end{aligned}$$

```
> 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});
```

$$\begin{aligned} dgl &:= \left(\frac{d}{dt} x_1(t) \right) + x_1(t) + x_2(t) = 0, \quad \left(\frac{d}{dt} x_2(t) \right) - x_1(t) + x_2(t) = 0 \\ \{x_1(t) &= e^{(-t)}(-C_1 \sin(t) + C_2 \cos(t)), \quad x_2(t) = e^{(-t)}(-C_1 \cos(t) + C_2 \sin(t))\} \end{aligned}$$

```
> inits:=x1(0)=1,x2(0)=0;
loes2:=dsolve({dgl,inits},{vars});
x1t:=rhs(op(1,loes2));
x2t:=rhs(op(2,loes2));
```

$$\begin{aligned} inits &:= x_1(0) = 1, \quad x_2(0) = 0 \\ loes2 &:= \{x_1(t) = e^{(-t)} \cos(t), \quad x_2(t) = e^{(-t)} \sin(t)\} \\ x1t &:= e^{(-t)} \cos(t) \\ x2t &:= e^{(-t)} \sin(t) \end{aligned}$$

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 Zeitintervall $t \in [0, T]$ aufgespannten rechteckigen Bereich $[\min x_1(t), \max x_1(t)] \times [\min x_2(t), \max x_2(t)]$.

Nur die Kurve erhält man, wenn das SysgDGL nicht autonom ist, 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({dgl},[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);

```

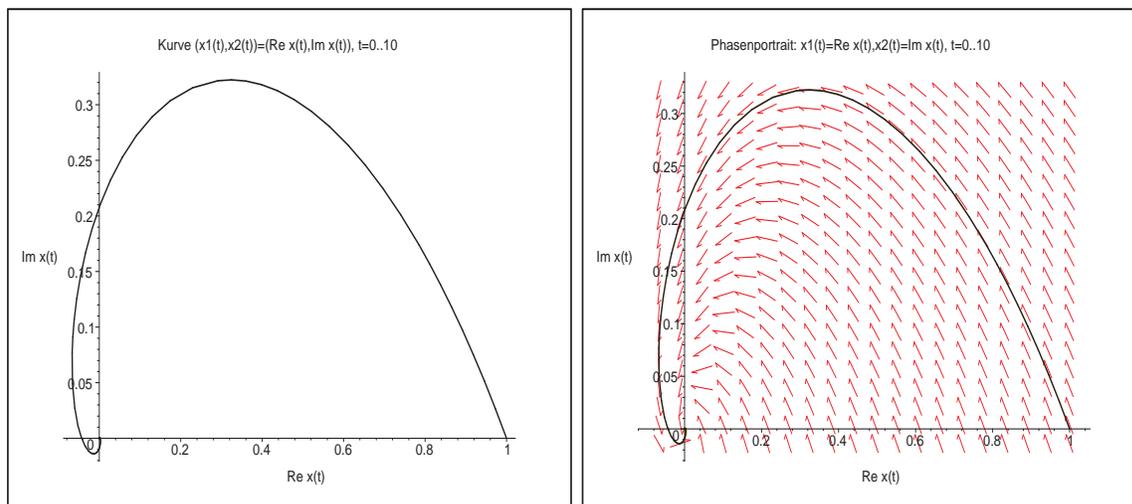


Abb. 4.3: Dateien *komp2.ps*, *komp3.ps*,

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 (4.10) 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 (4.12) 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:

```

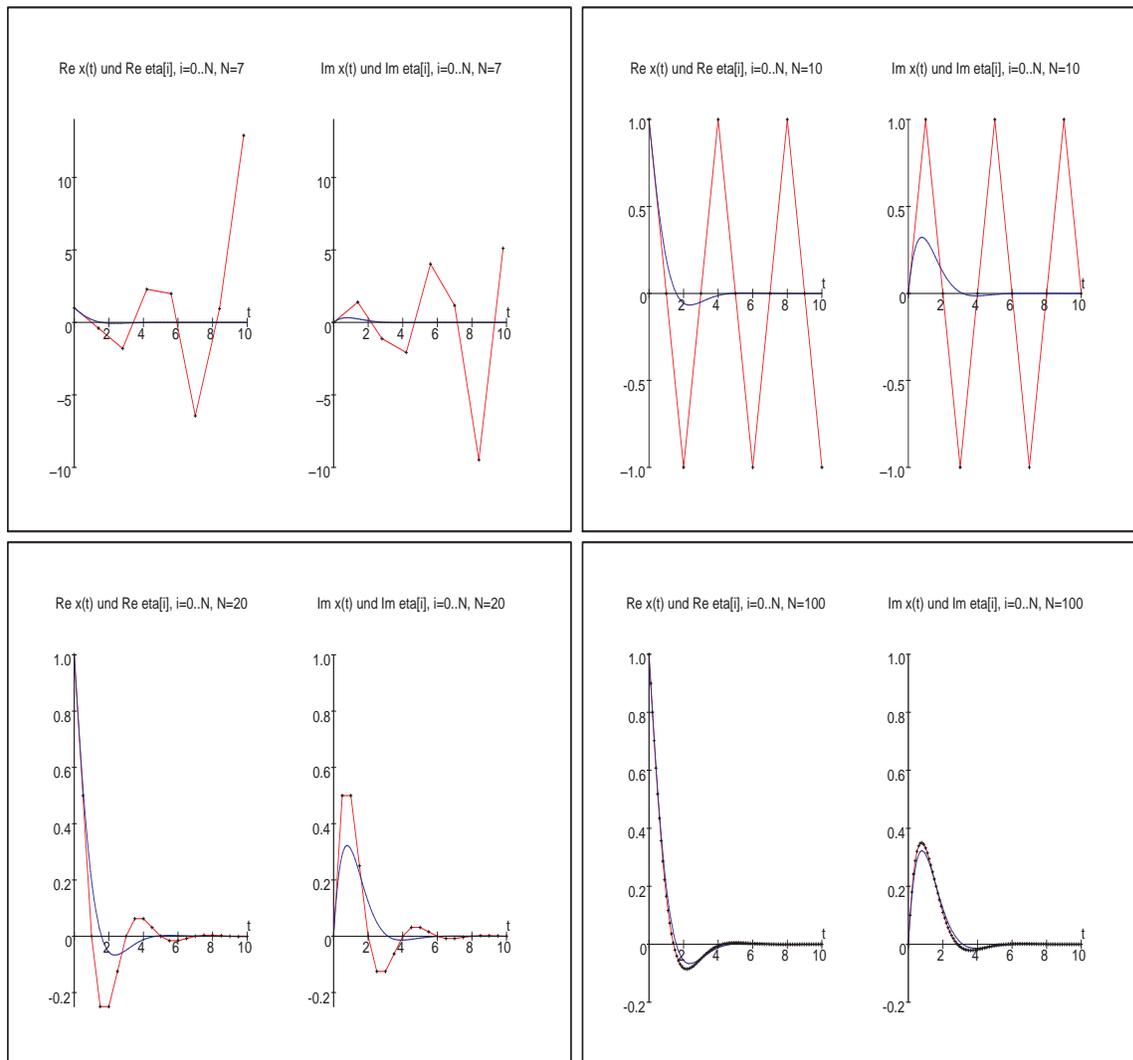


Abb. 4.4: Dateien *kpzv1.ps*, *kpzv2.ps*, *kpzv3.ps*, *kpzv4.ps*,
 Näherungswerte $\Re\eta_i$, $\Im\eta_i$, $i = 0, 1, \dots, N$, des PZV bei $h = 1.4, 1, \frac{1}{2}, \frac{1}{10}$,
 sowie exakte Lösungskurven $x_1(t)$, $x_2(t)$, $t \in [0, T]$, $T = 10$

```
> 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/kpvz4.ps':
  interface(plotdevice=ps,plotoutput=datei,
            plotoptions='color,portrait,width=640,height=480');
plots[display](pp);
interface(plotdevice=win);
```

(b) Für das IEV

$$\begin{aligned}\eta_{i+1} &= \eta_i + hf(t_{i+1}, \eta_{i+1}) = \eta_i + h\lambda\eta_{i+1}, \\ \eta_{i+1} &= \frac{1}{1 - h\lambda}\eta_i\end{aligned}\quad (4.13)$$

ist

$$F(z) = \frac{1}{1 - z}.\quad (4.14)$$

Für $\Re\lambda < 0$ ist $\Re z < 0$ 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 Einheitskreis um den Mittelpunkt $z_0 = 1$, 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 PC-Verfahren und definieren

$$\begin{aligned}\eta_{i+1}^{(P)} &= \eta_i + hf(t_i, \eta_i), \\ \eta_{i+1} &= \eta_i + hf(t_{i+1}, \eta_{i+1}^{(P)}).\end{aligned}$$

Einsetzen von $f(t, x) = \lambda x$ ergibt

$$\eta_{i+1} = (1 + h\lambda + (h\lambda)^2)\eta_i\quad (4.15)$$

und damit

$$F(z) = 1 + z + z^2.\quad (4.16)$$

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 PC-Verfahren hat ein Stabilitätsgebiet, das sich von denen seiner zwei Komponenten unterscheidet und wider Erwarten zu einer strengeren Forderung bezüglich der Schrittweite h führt.

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);
```

$$x1 := \phi \rightarrow \text{evalc}(\Re(s1))$$

$$x1 := \phi \rightarrow -\frac{1}{2} + \frac{1}{4} \sqrt{2 \sqrt{(-3 + 4 \cos(\phi))^2 + 16 \sin(\phi)^2} - 6 + 8 \cos(\phi)}$$

$$x2 := \phi \rightarrow \frac{1}{4} \text{csgn}(4 \sin(\phi) + 3I - 4I \cos(\phi)) \sqrt{2 \sqrt{(-3 + 4 \cos(\phi))^2 + 16 \sin(\phi)^2} + 6 - 8 \cos(\phi)}$$

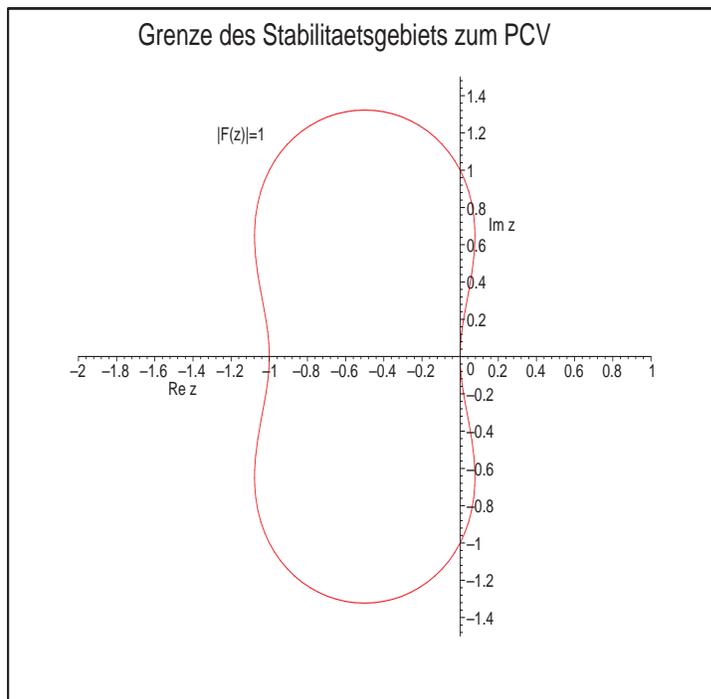
```
> y1:=unapply(evalc(Re(s2)),phi);
y2:=unapply(evalc(Im(s2)),phi);
```

$$y1 := \phi \rightarrow -\frac{1}{2} - \frac{1}{4} \sqrt{2 \sqrt{(-3 + 4 \cos(\phi))^2 + 16 \sin(\phi)^2} - 6 + 8 \cos(\phi)}$$

$$y2 := \phi \rightarrow -\frac{1}{4} \text{csgn}(4 \sin(\phi) + 3I - 4I \cos(\phi)) \sqrt{2 \sqrt{(-3 + 4 \cos(\phi))^2 + 16 \sin(\phi)^2} + 6 - 8 \cos(\phi)}$$

```
> 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. 4.5:**

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

$$F(z) = 1 + z + \frac{1}{2}z^2. \quad (4.17)$$

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 $F(z) = e^{z\varphi}$, $\varphi \in \mathbb{R}$, erfolgen. Damit ist

$$z = z(\varphi) = -1 \pm \sqrt{2e^{i\varphi} - 1}, \quad \varphi \in [-\pi, \pi]. \quad (4.18)$$

Die beiden Teilkonturen ergeben zusammen die Gebietsgrenze $\partial\mathcal{E}$, die einer Ellipse ähnlich ist (siehe Abb. 4.8). ◀

[4.5] Bemerkung

(a) Für explizite ESV 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 dann 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 z -Intervall $(z_{\min}, 0)$, aus dem die Schrittweitenbedingung $h < z_{\min}/\lambda$ folgt.

Darstellung mit Maple

```
> # Stabilitaetsgebiete (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='Stabilitaetsintervalle (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],[1,0.007]], [[-2,-0.01],[2,-0.01]],
           [[-2.785,0],[2,0]]},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);
```

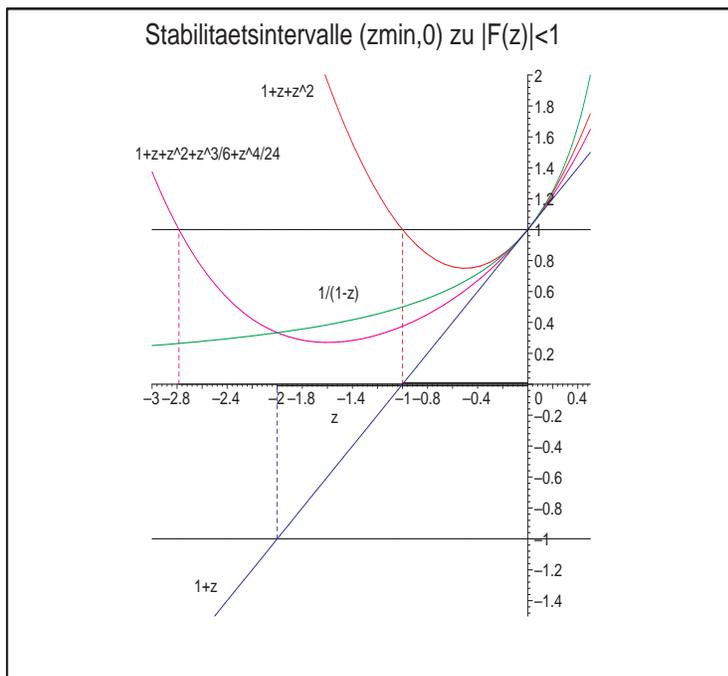


Abb. 4.6:
 Datei *stab_02.ps*,
 Polynome $F(z)$ zu
 PZV, IEV, PCV, KRKV und
 Stabilitätsintervalle $(z_{min}, 0)$,
 PZV: $z \in (-2, 0)$,
 IEV: $z \in (-\infty, 0)$,
 PCV: $z \in (-1, 0)$,
 KRKV: $z \in (-2.785, 0)$

(d) Analog kann man zu reellem $\lambda < 0$ die Polynome (Wachstumsfunktionen) $F(z)$ einiger RKV vergleichen.

Wir nehmen dazu die folgenden RKV: PZV, MPZV, HV, (3,2)-RKV, (3,3)-RKV (g), (3,3)-KV (h) und KRKV aus Abschnitt 2.2.1.

Die Polynome sind entsprechend dieser Reihenfolge

$$\begin{aligned}
 F(z) &= 1 + z, \\
 F(z) &= 1 + z + \frac{1}{2}z^2, \quad (2\times) \\
 F(z) &= 1 + z + \frac{1}{2}z^2 + \frac{1}{9}z^3, \\
 F(z) &= 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3, \quad (2\times) \\
 F(z) &= 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4, \quad z = h\lambda < 0.
 \end{aligned}$$

MPZV und HV haben das gleiche Polynom sowie (3,3)-RKV (g) und (3,3)-KV (h). Wir kontrollieren das zulässige z -Intervall $(z_{min}, 0)$, aus dem die Schrittweitenbedingung $h < z_{min}/\lambda$ folgt.

Die Vorgehensweise mit Maple ist wie in Teil (b).

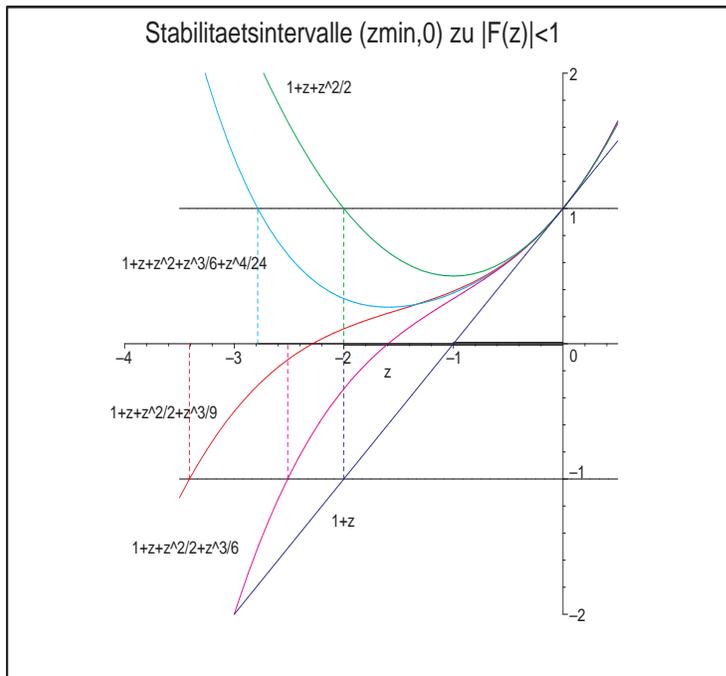


Abb. 4.7:

Datei *stab_03.ps*,

Polynome $F(z)$ zu mehreren RKV und

Stabilitätsintervalle $(z_{min}, 0)$,

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.

[4.6] Beispiel Ein **zweistufiges implizites RKV** sei definiert durch

$$k_1 = f(t_i + \frac{3-\sqrt{3}}{6}h, \eta_i + \frac{1}{4}hk_1 + \frac{3-2\sqrt{3}}{12}hk_2),$$

$$k_2 = f(t_i + \frac{3+\sqrt{3}}{6}h, \eta_i + \frac{3+2\sqrt{3}}{12}hk_1 + \frac{1}{4}hk_2)$$

sowie

$$\eta_{i+1} = \eta_i + \frac{h}{2}(k_1 + k_2). \tag{4.19}$$

Warum ist dieses Verfahren implizit? Worin besteht die Verallgemeinerung gegenüber den in Abschnitt 2.2 definierten RKV?

(1) Berechnung von $F(z)$

Mit $f(t, x) = \lambda x$ folgt das lineare Gleichungssystem für k_1 und k_2

$$k_1 = \lambda(\eta_i + \frac{1}{4}hk_1 + \frac{3-2\sqrt{3}}{12}hk_2),$$

$$k_2 = \lambda(\eta_i + \frac{3+2\sqrt{3}}{12}hk_1 + \frac{1}{4}hk_2)$$

mit der Lösung

$$k_1 = \frac{1 - \frac{\sqrt{3}}{6}h\lambda}{1 - \frac{1}{2}h\lambda + \frac{1}{12}(h\lambda)^2} \lambda\eta_i, \quad k_2 = \frac{1 + \frac{\sqrt{3}}{6}h\lambda}{1 - \frac{1}{2}h\lambda + \frac{1}{12}(h\lambda)^2} \lambda\eta_i.$$

Hieraus folgt

$$F(z) = \frac{1 + \frac{z}{2} + \frac{z^2}{12}}{1 - \frac{z}{2} + \frac{z^2}{12}}. \quad (4.20)$$

(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

$$\frac{1 + e^{i\varphi}}{1 - e^{i\varphi}} = \frac{1 + \cos(\varphi) + i \sin(\varphi)}{1 - \cos(\varphi) - i \sin(\varphi)} = \frac{(1 + \cos(\varphi) + i \sin(\varphi))(1 - \cos(\varphi) + i \sin(\varphi))}{(1 - \cos(\varphi))^2 - (i \sin(\varphi))^2} = \frac{i \sin(\varphi)}{1 - \cos(\varphi)}$$

auf die Gleichung

$$z^2 + i 6az + 12 = 0 \quad \text{mit} \quad a = \frac{\sin(\varphi)}{1 - \cos(\varphi)} \in \mathbb{R}.$$

Die Lösungen

$$z = i(-3a \pm \sqrt{9a^2 + 12})$$

liegen alle 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.

```
> # Stabilitaetsgebiete |F(z)|<1 im komplexen Fall
# bei z=h*lambda, Re(lambda)<0
# Grenzen der Stabilitaetsgebiete |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+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); # KRKV
```

```

> 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,
    ploptions='color,portrait,width=640,height=480');
plots[display](p0,p1,p2,p3,p4,p5,p6,p7);
interface(plotdevice=win);

```

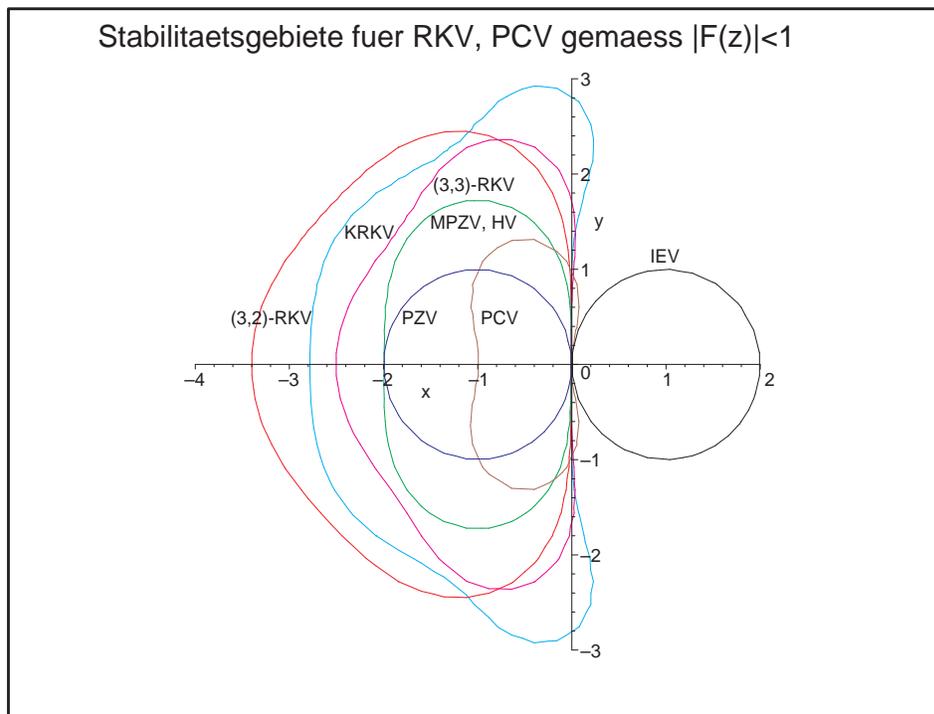


Abb. 4.8: Datei *stab_04.ps*, Stabilitätsgebiete \mathcal{E} gemäß $|F(z)| < 1$ zu mehreren RKV: IEV, PZV, PCV, MPZV=HV, (3,3)-RKV, KRKV, (3,2)-RKV

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.

$$I_A = \mathcal{E} \cap \mathbb{R}^- = \{\xi \in \mathbb{R} : \xi < 0 \text{ und } |F(\xi)| < 1\}. \quad (4.21)$$

[4.7] Beispiele

(a) Für das explizite und das implizite Euler-Verfahren haben wir dies in den Beispielen [4.4] (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 [4.4] (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 $3/4$. 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 $F(\xi) = 1$. Es folgt aus Maple-Berechnungen bzw. numerischen Rechnungen, z. B. mit dem Newton-Verfahren,

$$\begin{aligned} I_A &= \left(-\frac{4}{3} - \frac{1}{3}(172 + 36\sqrt{29})^{1/3} + \frac{20}{3}(172 + 36\sqrt{29})^{-1/3}, 0\right) \\ &= \left(-\frac{4}{3} - \sqrt[3]{\frac{4\sqrt{29}}{3} + \frac{172}{27}} + \sqrt[3]{\frac{4\sqrt{29}}{3} - \frac{172}{27}}, 0\right) \\ &= (-2.785\,293, 0). \end{aligned}$$

(d) Für das (3,3)-RKV ist

$$F(\xi) = 1 + \xi + \frac{1}{2}\xi^2 + \frac{1}{6}\xi^3$$

und

$$I_A = \left(-1 - \sqrt[3]{\sqrt{17} + 4} + \sqrt[3]{\sqrt{17} - 4}, 0\right) = (-2.512\,745, 0). \quad \blacktriangleleft$$

4.2 Stabilität von MSV

Übertragen wir nun diese Überlegungen auf MSV (3.49).

Setzen wir das Testproblem $x' = \lambda x$ in das MSV ein, so erhalten wir die homogene lineare Differenzgleichung

$$(1 - h\lambda b_r)\eta_{j+r} + \dots + (a_1 - h\lambda b_1)\eta_{j+1} + (a_0 - h\lambda b_0)\eta_j = 0. \quad (4.22)$$

Zur Lösung bedienen wir uns, entsprechend den Überlegungen des Abschnitts 3.3 des mit (4.22) verknüpften Polynoms

$$\phi(\xi) = (1 - h\lambda b_r)\xi^r + \dots + (a_1 - h\lambda b_1)\xi + (a_0 - h\lambda b_0) = \rho(\xi) - h\lambda\sigma(\xi), \quad (4.23)$$

wobei $\xi \in \mathbb{C}$ sowie $\rho(\xi)$ und $\sigma(\xi)$ die in der Definition [3.19] notierten charakteristischen Polynome des MSV sind.

Für $h = 0$ haben wir das Polynom $\phi(\xi) = \rho(\xi)$ (3.45) sowie dazu die Gestalt der Lösung η_i (3.47) schon untersucht.

Hat $\phi(\xi)$ die r paarweise verschiedenen Nullstellen $\xi_1, \xi_2, \dots, \xi_r$, so lässt sich nach Satz [3.16] die allgemeine Lösung der Differenzgleichung (3.45) darstellen in der Form

$$\eta_i = \gamma_1 \xi_1^i + \gamma_2 \xi_2^i + \dots + \gamma_r \xi_r^i. \quad (4.24)$$

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 [3.17]. Dies führt zur folgenden Definition.

[4.8] Definition Zu einem MSV heißt die Menge der Werte $z = h\lambda$, für welche das charakteristische Polynom

$$\phi(\xi) = \rho(\xi) - z\sigma(\xi) \quad (4.25)$$

nur Nullstellen ξ_k mit Beträgen $|\xi_k| < 1$ besitzt, das **Gebiet der absoluten Stabilität**.

Im Allgemeinen ist es schwierig, diese Gebiete 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 $\phi(\xi)$ 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

$$\rho(e^{i\varphi}) - z\sigma(e^{i\varphi}) = 0. \quad (4.26)$$

Damit lässt sich das Gebiet der absoluten Stabilität eingrenzen durch die numerische Auswertung des Ausdrucks

$$z = z(\varphi) = z_1(\varphi) + \iota z_2(\varphi) = \frac{\rho(e^{\iota\varphi})}{\sigma(e^{\iota\varphi})}, \quad \varphi = [0, 2\pi). \quad (4.27)$$

Der Wert $z(0)$ ist hierbei in der Regel bekannt.

Nach einer der beiden Konsistenzbedingungen (3.52) 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 $\eta_{i+1} = F(z)\eta_i$ (4.8) 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 Eins sein muss. Die Bedingung $F(z) = e^{\iota\varphi}$ wird für die Bestimmung des Randes des Stabilitätsgebietes als Konturkurve (Niveaulinie) $|F(z)| = |F(x + \iota y)| = 1$ verwendet. Diese haben wir in der Abbildung 4.8 bereits dargestellt. Die Vorgehensweise entspricht genau der Aussage in Definition [4.3].

4.2.1 Stabilitätsgebiet von MSV

Wir haben die Parameterschemata von vier Klassen von MSV in den Abschnitten 3.1, 3.2 gezeigt. Dazu wollen wir die Stabilitätsgebiete berechnen. Wie in Bemerkung [4.5] (b) würde die Illustration von \mathcal{E} im Bereich \mathbb{C}^- bei nicht negativen Imaginärteil genügen. Auch hier haben wir die Symmetrie des Gebietes zur reellen Achse, denn es gilt im Komplexen bei $\varphi = [0, 2\pi)$

$$z = \frac{\rho(e^{\iota\varphi})}{\sigma(e^{\iota\varphi})}, \quad \bar{z} = \frac{\rho(\overline{e^{\iota\varphi}})}{\sigma(\overline{e^{\iota\varphi}})} = \frac{\rho(e^{-\iota(\varphi-2\pi)})}{\sigma(e^{-\iota(\varphi-2\pi)})} = \frac{\rho(e^{\iota(2\pi-\varphi)})}{\sigma(e^{\iota(2\pi-\varphi)})} = \frac{\rho(e^{\iota\zeta})}{\sigma(e^{\iota\zeta})}, \quad \zeta \in (0, 2\pi].$$

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 1. Beträge von weiteren Nullstellen können dort durchaus sowohl kleiner als auch größer als 1 sein. Wir werden dazu auch Abbildungen mit Verlauf mehrerer Konturen

$$z = \frac{\rho(c e^{\iota\varphi})}{\sigma(c e^{\iota\varphi})}, \quad |c e^{\iota\varphi}| = c, \quad c \in (0, +\infty),$$

erstellen.

Die Berechnungen kann man sowohl in Maple also in MATLAB machen, was wir auch im ersten Fall tun wollen. Dabei muss man auf die in den Computeralgebrasystemen 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 AB(n), $n = 1, 2, \dots, 6$, entnehmen wir der Tabelle 3.2.

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:
```

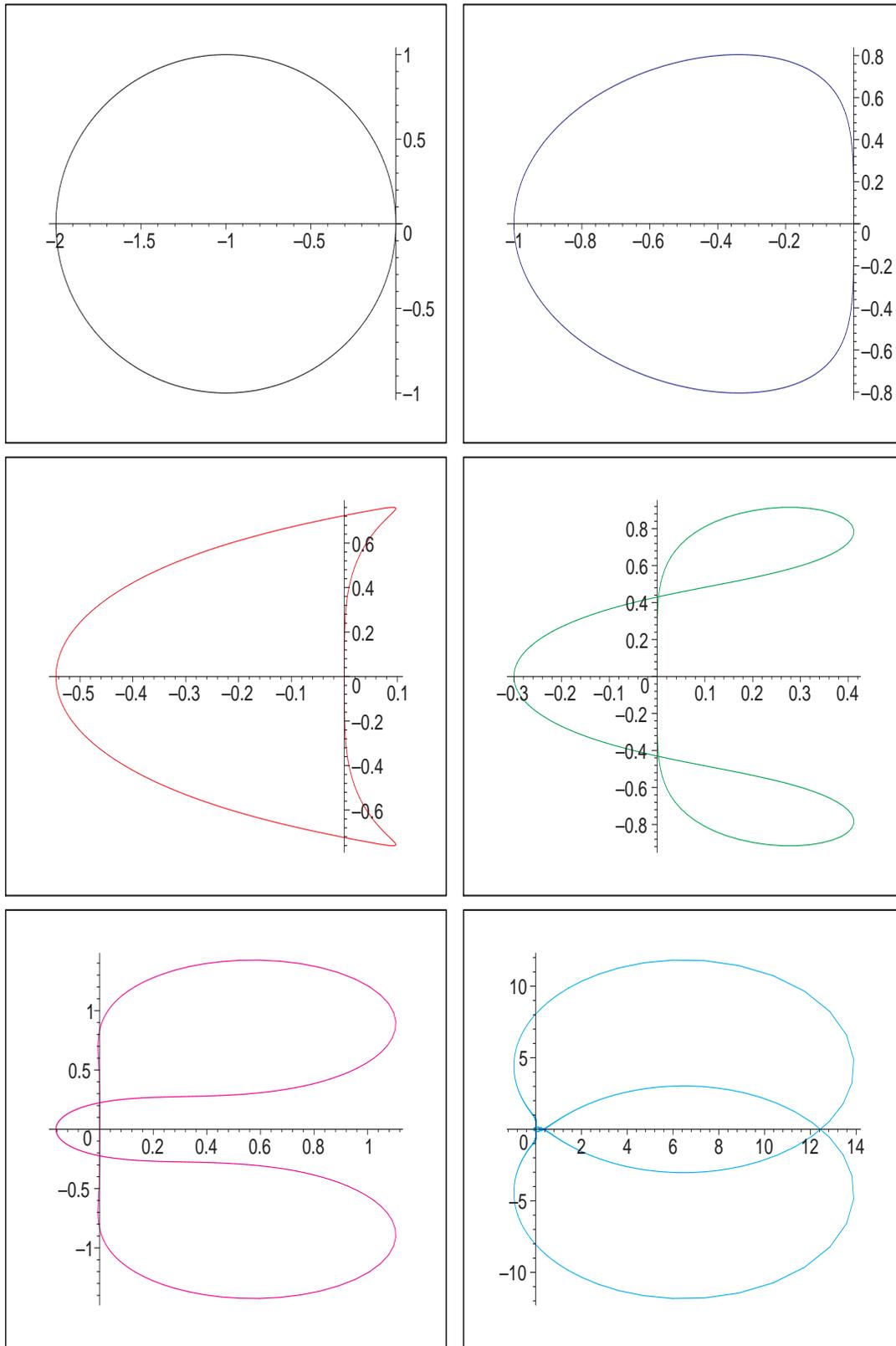


Abb. 4.9: Dateien *ab_stab01..06.eps*, Stabilitätsgebiete \mathcal{E} gemäß Nullstellen $|\xi_k| < 1$ von $\phi(\xi)$ zum MSV $AB(n)$, $n = 1, 2, \dots, 6$

Wir fassen alle 6 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');
```

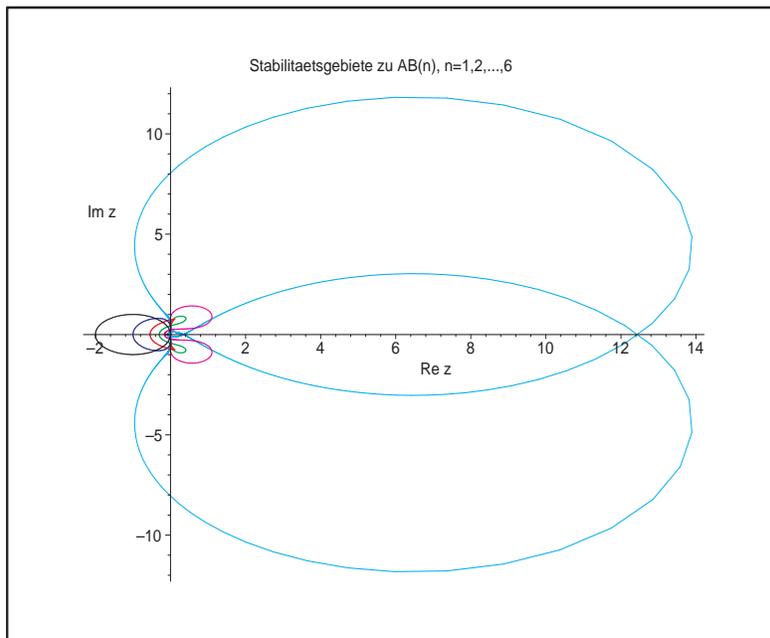


Abb. 4.10:

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');
```

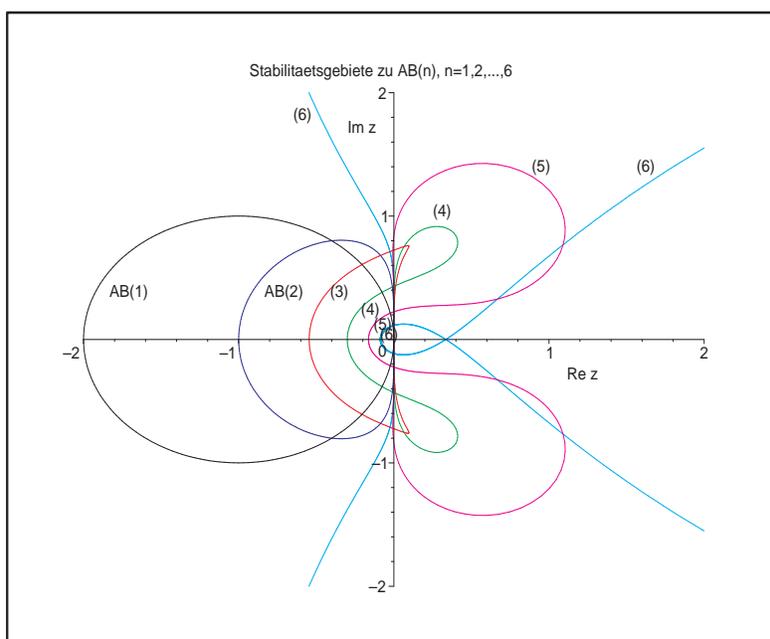


Abb. 4.11:

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');
```

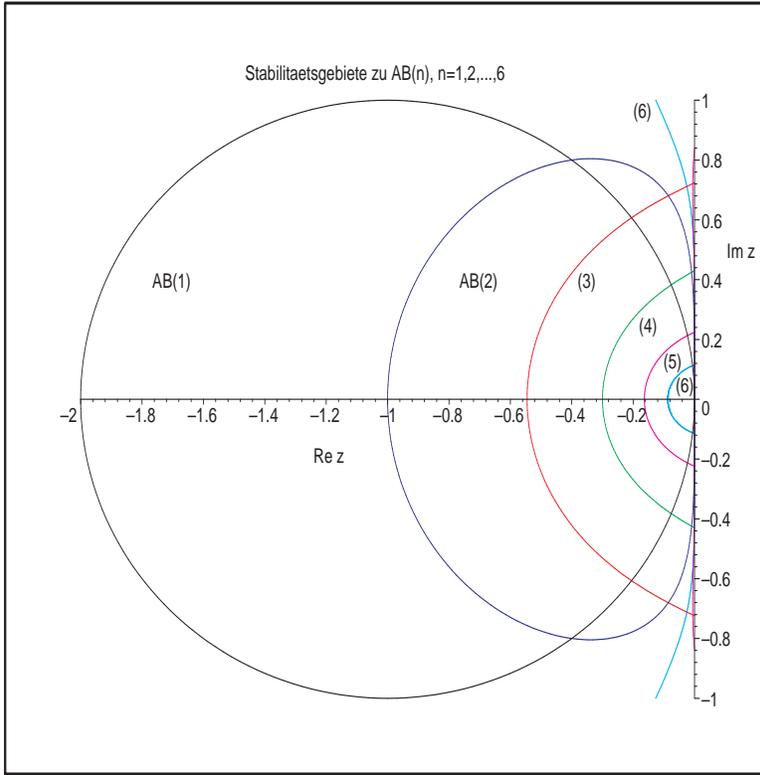


Abb. 4.12:
 Datei *ab_stab09.ps*,
 Stabilitätsgebiete \mathcal{E}
 zu $AB(n)$, $n = 1, 2, \dots, 6$

Man erkennt, dass die Stabilitätsgebiete zu $AB(n)$ mit wachsendem n systematisch "kleiner" werden. Damit wird die Schrittweitenbedingung für 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", so dass dort ein kleines symmetrisches Intervall auf der imaginären Achse ein Randstück des Stabilitätsgebiets bildet. Bei Abbildungen mit Stabilitätsgebieten in der Literatur wird meistens die komplexe Halbebene \mathbb{C}^+ nicht gezeichnet.

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 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('Stabilitaetsgebiete 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

```

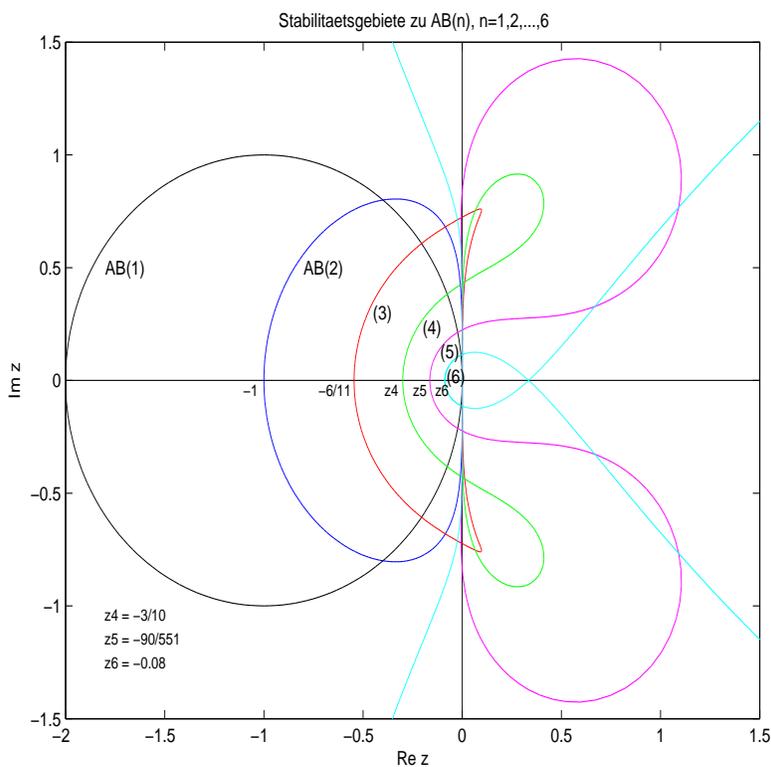


Abb. 4.13:
 Datei *ab_stab1.ps*,
 Stabilitätsgebiete \mathcal{E}
 zu $AB(n)$, $n = 1, 2, \dots, 6$

Wir betrachten nun die Situation zum Stabilitätsgebiet \mathcal{E} von AB(5).

Dabei zeichnen wir den Verlauf weiterer Konturen. Für Werte $c = |ce^{2\varphi}| < 1$ 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.

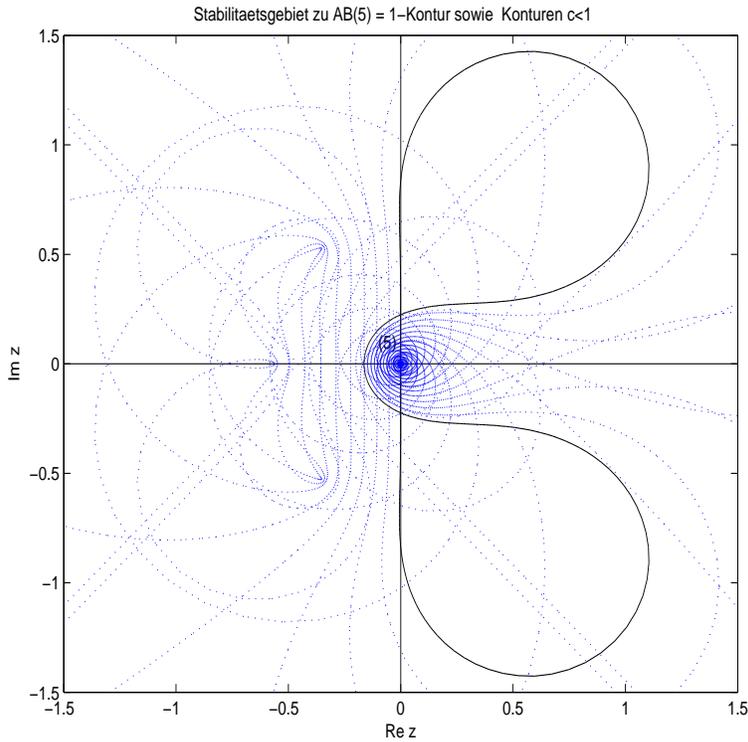


Abb. 4.14:
Datei *ab_stab2a.ps*,
Stabilitätsgebiet
zu AB(5) sowie
weitere Konturen
mit $c = 0.05(0.05)1$

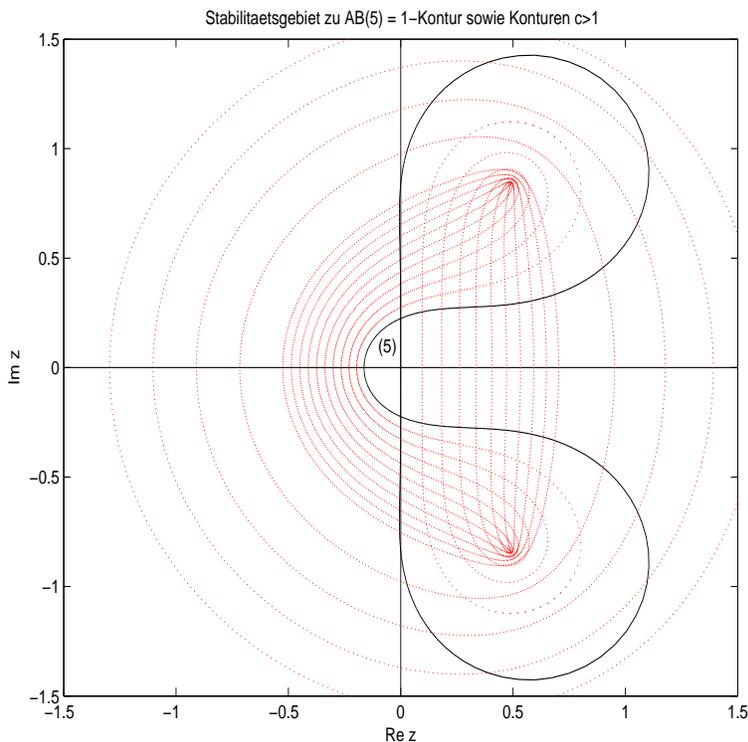


Abb. 4.15:
Datei *ab_stab2b.ps*,
Stabilitätsgebiet
zu AB(5) sowie
weitere Konturen
mit $c = 1(0.1)2(0.5)4$

Die nächsten Fälle machen wir nicht so ausführlich bzw. nur auszugsweise.

Zum AB(2)

$$\eta_{i+1} = \eta_i + \frac{h}{2}[3f(t_i, \eta_i) - f(t_{i-1}, \eta_{i-1})], \quad i = 1, 2, \dots,$$

machen wir einige zusätzliche Betrachtungen und Rechnungen, die für andere einfache MSV in ähnlicher Weise möglich sind.

Dazu haben wir

$$\begin{aligned} \eta_{i+1} - (1 + \frac{3}{2}h\lambda)\eta_i + \frac{1}{2}h\lambda\eta_{i-1} &= 0, \\ \rho(\xi) &= \xi^2 - \xi = \xi(\xi - 1), \quad \text{Stabilitätskriterium erfüllt,} \\ \sigma(\xi) &= \frac{3}{2}\xi - \frac{1}{2}, \\ \phi(\xi) &= \xi^2 - (1 + \frac{3}{2}z)\xi + \frac{1}{2}z, \quad z = h\lambda, \\ \xi_{1,2} &= \frac{1}{2}\left(1 + \frac{3}{2}z \pm \sqrt{(1 + \frac{3}{2}z)^2 - 2z}\right), \quad \text{Nullstellen von } \phi(\xi) \\ &= \frac{1}{4}\left(2 + 3z \pm \sqrt{9z^2 + 4z + 4}\right) \\ &= \frac{1}{4}\left(2 + 3z \pm 3\sqrt{(z + \frac{2}{9})^2 + \frac{32}{81}}\right). \end{aligned}$$

Der Rand des Stabilitätsgebiets ist in Abb. 4.12 gezeigt.

Das daraus erkennbare reelle Stabilitätsintervall $(-1, 0)$, wo $|\xi_{1,2}(z)| < 1$, kann am durch den grafischen Verlauf der beiden reellwertigen Nullstellenkurven $\xi_{1,2}(z)$ deutlich machen.

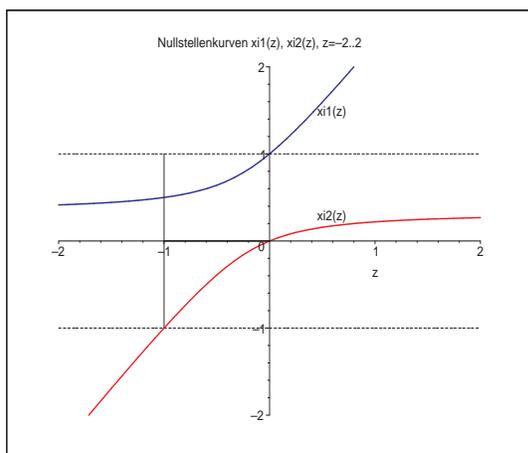


Abb. 4.16:

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 $\lambda = -1$, so dass $z = -h$ ist.

AB(2) ist absolut stabil für $h < 1$ und wächst unbeschränkt und oszillierend für $h > 1$.

Die allgemeine Lösung des MSV als Differenzengleichung 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 zu η_0, η_1 ergeben.

Wir nehmen $h = 1$ und $\eta_0 = 1, \eta_1 = 1 - h$ und erhalten

$$\eta_0 = c_1 + c_2 = 1, \eta_1 = c_1\xi_1 + c_2\xi_2 = 1 - h,$$

$$c_1 = \frac{1}{3}, \quad c_2 = \frac{2}{3},$$

$$\eta_i = \frac{1}{2}(\eta_{i-2} - \eta_{i-1}) = \frac{1}{3}(-1)^i + \frac{2}{3}\left(\frac{1}{2}\right)^i.$$

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

$$\begin{aligned} 1 &= \eta_0 > \eta_2 > \eta_4 > \dots > \eta_{2i} > \eta_{2i+2} > \dots > \frac{1}{3}, \\ 0 &= \eta_1 > \eta_3 > \eta_5 > \dots > \eta_{2i-1} > \eta_{2i+1} > \dots > -\frac{1}{3}. \end{aligned}$$

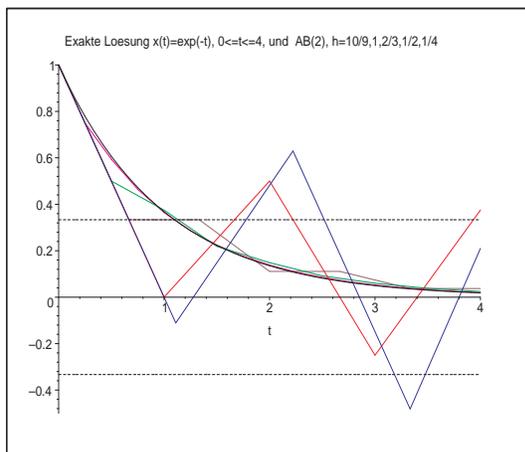


Abb. 4.17:

Datei *ab_stab20.ps*,

Exakte Lösung $x(t) = e^{-t}, 0 \leq t \leq 4$,
und AB(2) mit $h = \frac{10}{9}, 1, \frac{2}{3}, \frac{1}{2}, \frac{1}{4}$

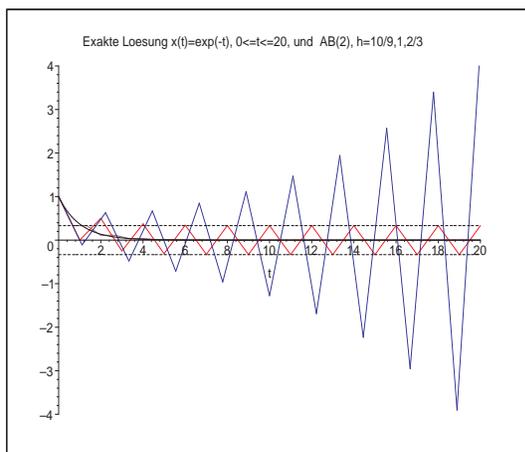


Abb. 4.18:

Datei *ab_stab21.ps*,

Exakte Lösung $x(t) = e^{-t}, 0 \leq t \leq 20$,
und 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 der AM(n), $n = 1, 2, \dots, 6$, $s = \max(n - 1, 1)$, entnehmen wir der Tabelle 3.4.

```
% 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;
    ...
```

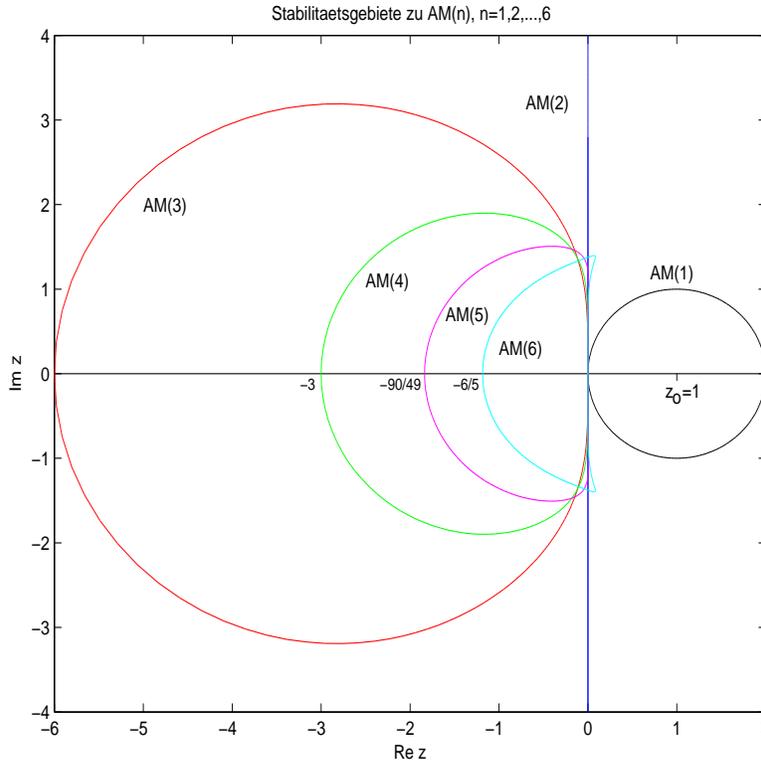


Abb. 4.19:
Datei *am_stab1.ps*,
Stabilitaetsgebiete \mathcal{E}
zu AM(n), $n = 1, 2, \dots, 6$

Nur AM(1) und AM(2) sind A-stabile MSV, weil die linke komplexe Halbebene \mathbb{C}^- zum Stabilitaetsgebiet \mathcal{E} gehoert. Die anderen vier AM-Formeln fuehren auf entsprechende Beschaerankungen der Integrationsschrittweite h .

Da AM(1) und AM(2) einstufige MSV sind, ist die Funktion $\phi(\xi)$ (4.25) problemlos zu berechnen.

AM(1): Es gilt

$$\phi(\xi) = \rho(\xi) - z\sigma(\xi) = \xi - 1 - z\xi = (1 - z)\xi - 1,$$

so dass ihre einzige Nullstelle $\xi_1 = \frac{1}{1-z}$ ist. Die Stabilitätsbedingung $|\xi_1| < 1$ ist einfach auszuwerten, genauso die Berechnung der Konturen $\left|\frac{1}{1-z}\right| = c$.

Damit ist $|z - 1| = 1$ die Kreisgleichung für den Rand $\partial\mathcal{E}$ des Stabilitätsgebiets \mathcal{E} .

Die Konturen $\left|\frac{1}{1-z}\right| = c$, $c > 0$, sind Kreisumfänge zum Mittelpunkt $z_0 = 1$ und mit dem Radius $\frac{1}{c}$, so dass 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

$$\phi(\xi) = \rho(\xi) - z\sigma(\xi) = (1 - \frac{z}{2})\xi - (1 + \frac{z}{2}),$$

so dass 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 $\left|\frac{1+z/2}{1-z/2}\right| = c$.

Die Bedingung $\left|\frac{1+z/2}{1-z/2}\right| = 1$ für den Rand des Stabilitätsgebiets kann nur für imaginäre Werte z erfüllt werden, denn dann sind Zähler und Nenner zueinander konjugierte 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 $z = 2(e^{i\varphi} - 1)/(e^{i\varphi} + 1) = 2(e^{i\varphi/2} - e^{-i\varphi/2})/(e^{i\varphi/2} + e^{-i\varphi/2}) = i 2 \tan(\varphi/2)$.

Die Konturen $\left|\frac{1+z/2}{1-z/2}\right| = c$ 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

$$abst(c) = -m(c) - r(c) = 2 \left[\frac{1 + c^2}{1 - c^2} - \sqrt{\left(\frac{1 + c^2}{1 - c^2}\right)^2 - 1} \right] \geq 0.$$

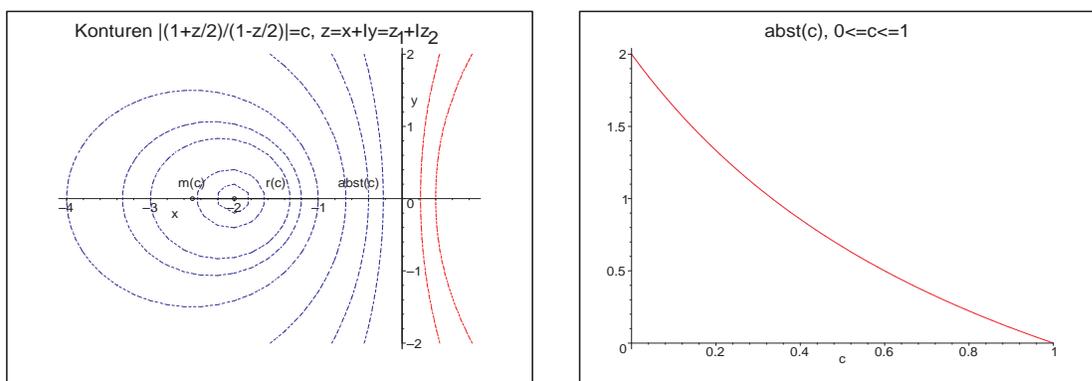


Abb. 4.20: Dateien *am_stab01.ps*, *am_stab02.ps*, Konturen $\left| \frac{1+z/2}{1-z/2} \right| = c$, $c = 0, \frac{1}{20}, \frac{1}{10}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{4}{5}, 1, \frac{5}{4}, \frac{3}{2}$, sowie Abstandsfunktion $abst(c)$, $c \in [0, 1]$, zu AM(2)

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.

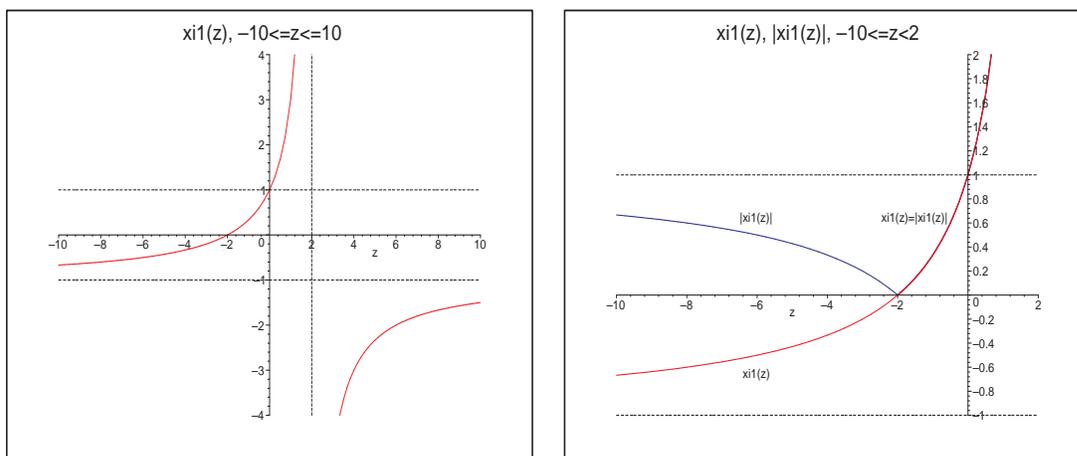


Abb. 4.21: Dateien *am_stab03.ps*, *am_stab04.ps*, Verlauf der Nullstellenfunktionen $\xi_1(z)$ bzw. $|\xi_1(z)|$ zu AM(2)

Nun vergleichen wir den Verlauf der 1-Kontur mit dem weiterer Konturen sowohl bei $c = |ce^{2\varphi}| < 1$ als auch $c > 1$. Dabei treffen ähnliche Bemerkungen wie beim MSV AB(5) zu.

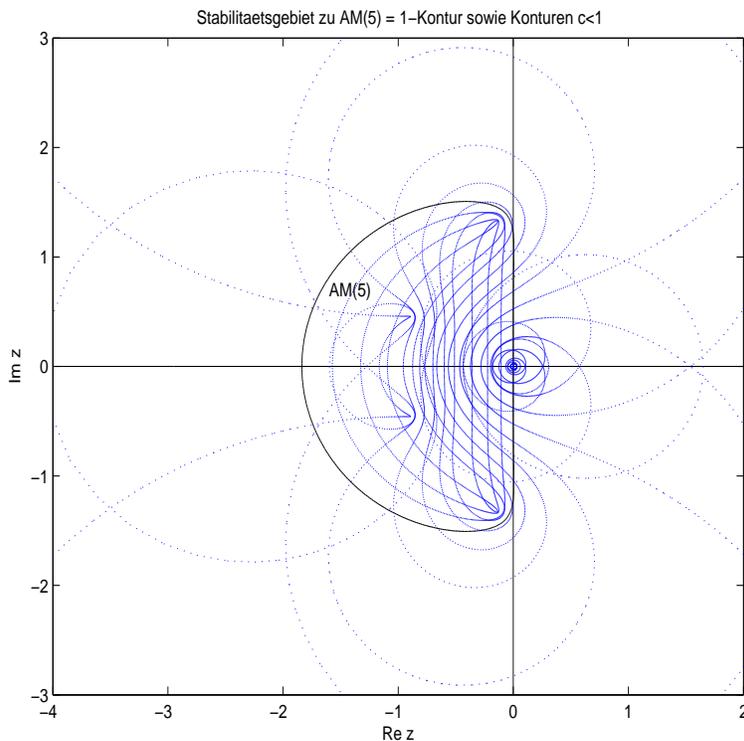


Abb. 4.22:

Datei *am_stab2a.ps*,
 Stabilitätsgebiet
 zu AM(5) sowie
 weitere Konturen
 mit $c = 0.05(0.05)0.6(0.1)1$

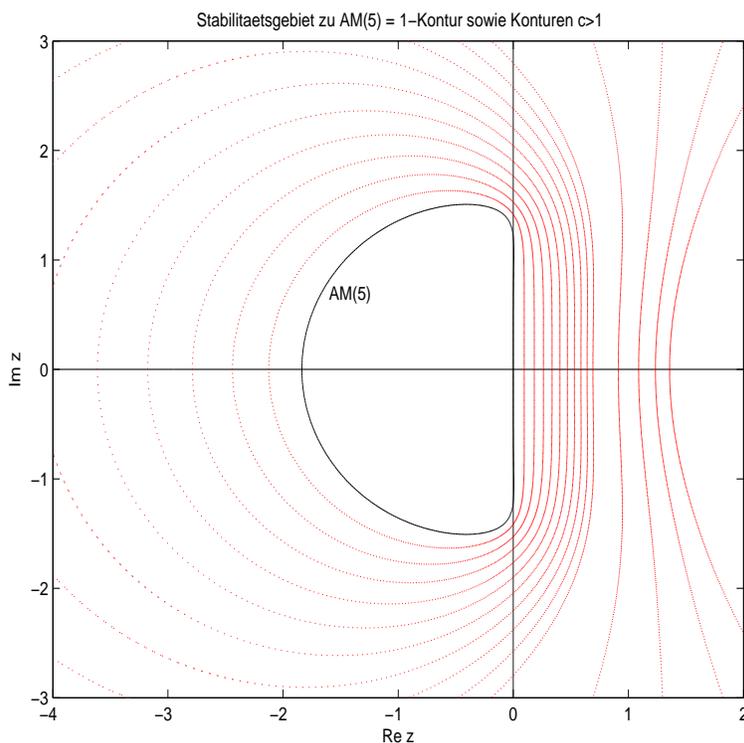


Abb. 4.23:

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 AN(n), $n = 2, 3, 4$, entnehmen wir der Tabelle 3.5.

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 Polynome findet man in

$$\phi(\xi) = \rho(\xi) - z\sigma(\xi) = (\xi^2 - 1) - z(2\xi) = \xi^2 - 2z\xi - 1, \quad z = h\lambda.$$

Das Stabilitätskriterium des Satzes [3.17] zu $\rho(\xi) = \xi^2 - 1$ ist erfüllt, denn seine beiden Nullstellen sind ± 1 . Dieser Umstand bzw. auch die Wachstumsbeschränkung (3.48) 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 $z = i \sin(\varphi)$, $\varphi \in [0, 2\pi)$, 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

$$\xi_{1,2} = z \pm \sqrt{z^2 + 1}$$

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 $\xi_1 = -1/\xi_2$ 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.

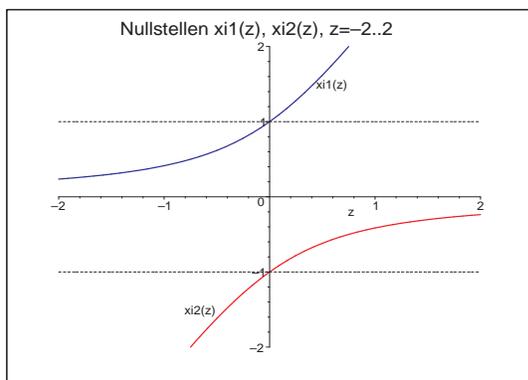


Abb. 4.24:

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 Differenzgleichung

$$\eta_{i+1} = \eta_{i-1} + 2h f(t_i, \eta_i), \quad i = 1, 2, \dots, \eta_0, \eta_1 \text{ gegeben,}$$

betrachten und für das Test-AWP $x' = \lambda x$, $x(0) = 1$, einfach als rekursive Formel berechnen 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 $x(t) = e^{-t}$.

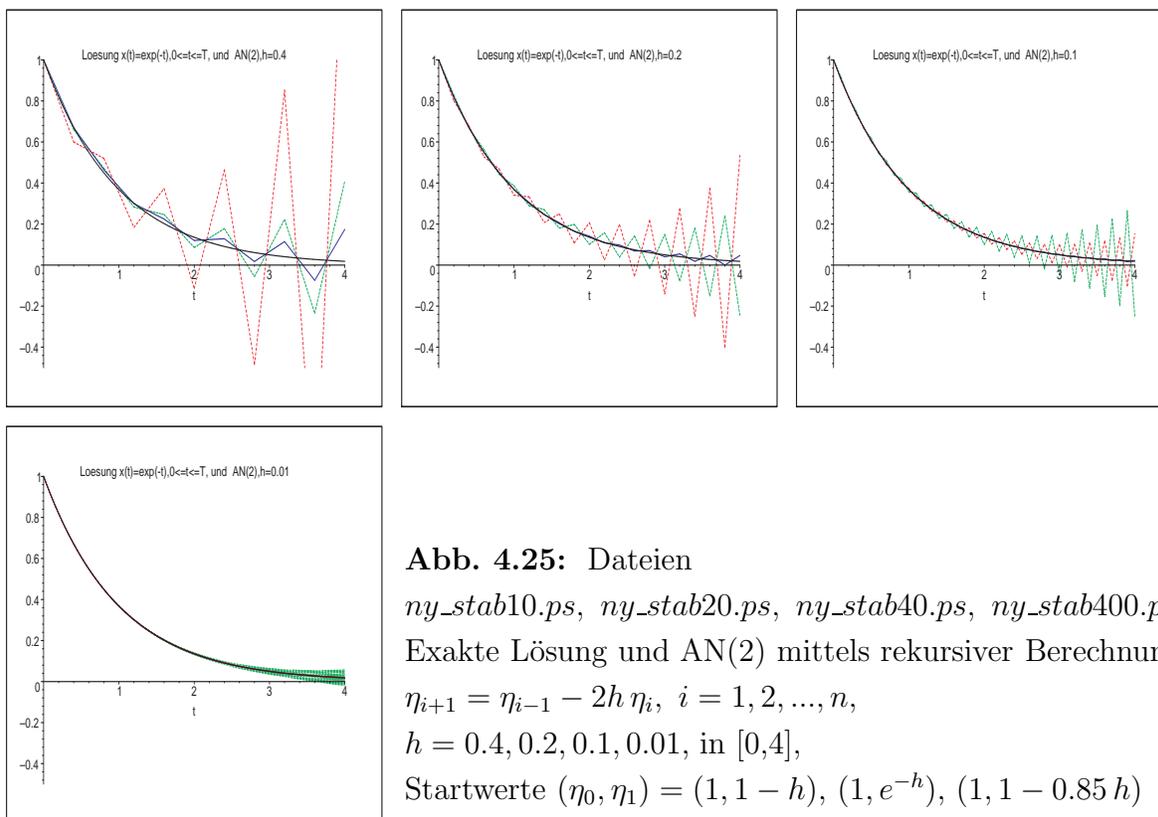


Abb. 4.25: Dateien

ny_stab10.ps, *ny_stab20.ps*, *ny_stab40.ps*, *ny_stab400.ps*,

Exakte Lösung und AN(2) mittels rekursiver Berechnung

$$\eta_{i+1} = \eta_{i-1} - 2h \eta_i, \quad i = 1, 2, \dots, n,$$

$$h = 0.4, 0.2, 0.1, 0.01, \text{ in } [0, 4],$$

$$\text{Startwerte } (\eta_0, \eta_1) = (1, 1 - h), (1, e^{-h}), (1, 1 - 0.85 h)$$

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 der Iterierten werden diese "früher oder später" divergieren und oszillierendes Verhalten zeigen.

Wir testen ebenfalls die drei Varianten mit den Startwerten $\eta_0 = 1$ und $\eta_1 = 1 - h, e^{-h}, 1 - 0.85 h$.

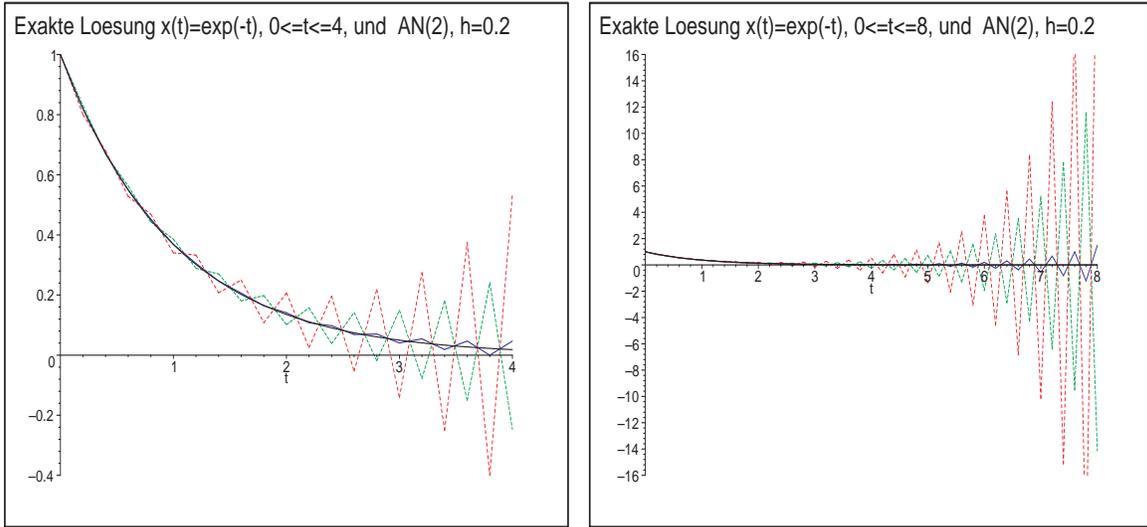


Abb. 4.26: Dateien *ny_stab01.ps*, *ny_stab02.ps*, AN(2) mittels rekursiver Berechnung $\eta_{i+1} = \eta_{i-1} - 2h\eta_i, i = 1, 2, \dots, n, h = 0.2$, in $[0, 4]$ bzw. $[0, 8]$, Startwerte $(\eta_0, \eta_1) = (1, 1 - h), (1, e^{-h}), (1, 1 - 0.85 h)$

Man sollte sich also nicht von dem guten anfänglichen Verlauf der Näherungswerte irritieren lassen. Die Divergenz tritt ein. Bei den exakten Startwerten $(\eta_0, \eta_1) = (1, e^{-h})$ beobachtet man das "Aufschaukeln" der Iterierten 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

$$\begin{aligned} \eta_0 &= c_1 + c_2 = 1, & \eta_1 &= c_1 \xi_1 + c_2 \xi_2 = 1 - h, \\ c_1 &= \frac{h^2 + 1 + \sqrt{h^2 + 1}}{2(h^2 + 1)}, & c_2 &= \frac{h^2 + 1 - \sqrt{h^2 + 1}}{2(h^2 + 1)}, \end{aligned}$$

$$\eta_i = \frac{1}{2} \left(1 + \frac{1}{\sqrt{h_1}} \right) \left(-h + \sqrt{h_1} \right)^i + \frac{1}{2} \left(1 - \frac{1}{\sqrt{h_1}} \right) \left(-h - \sqrt{h_1} \right)^i, \quad h_1 = h^2 + 1.$$

Die Grafik des Polygonzugs $(t_i, \eta_i), i = 1, 2, \dots, n$, ist die gestrichelte "größere" Zick-Zack-Kurve in der Abbildung 4.26.

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 [3.17] zu $\rho(\xi) = \xi^3 - \xi$ ist erfüllt, denn seine drei Nullstellen sind $0, \pm 1$ und vom Betrag ≤ 1 . Weitere Untersuchungen ergeben eine Randkontur $\partial\tilde{\mathcal{E}}$ in Form einer Doppelschleife zu einem Gebiet $\tilde{\mathcal{E}}$ in \mathbb{C}^+ . $\partial\tilde{\mathcal{E}}$ enthält als einzigen reellen Punkt $z = 0$. Im Gebiet $\tilde{\mathcal{E}}$ sowohl außerhalb verlaufen Konturen zum Wert $c = |e^{i\varphi}| > 1$ und $c < 1$. AN(4) zeigt analoges Verhalten. Daher kann nicht die Rede sein von einem Stabilitätsgebiet.

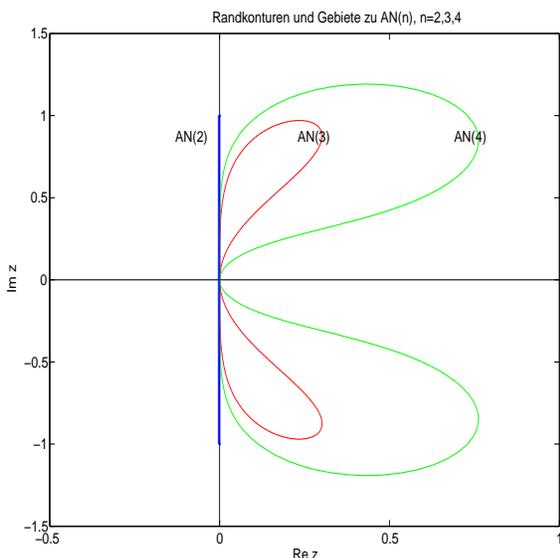


Abb. 4.27:
 Datei *ny_stab1.ps*,
 AN(2,3,4) mit Randkonturen $\partial\tilde{\mathcal{E}}$
 gemäß $z = \frac{\rho(\xi)}{\sigma(\xi)}$, $\xi = e^{i\varphi}$, $\varphi \in [0, 2\pi)$

Damit können z. B. für reelles $z < 0$ die Beträge nicht aller Nullstellenkurven $\xi_{1,2,3}$ kleiner als Eins werden.

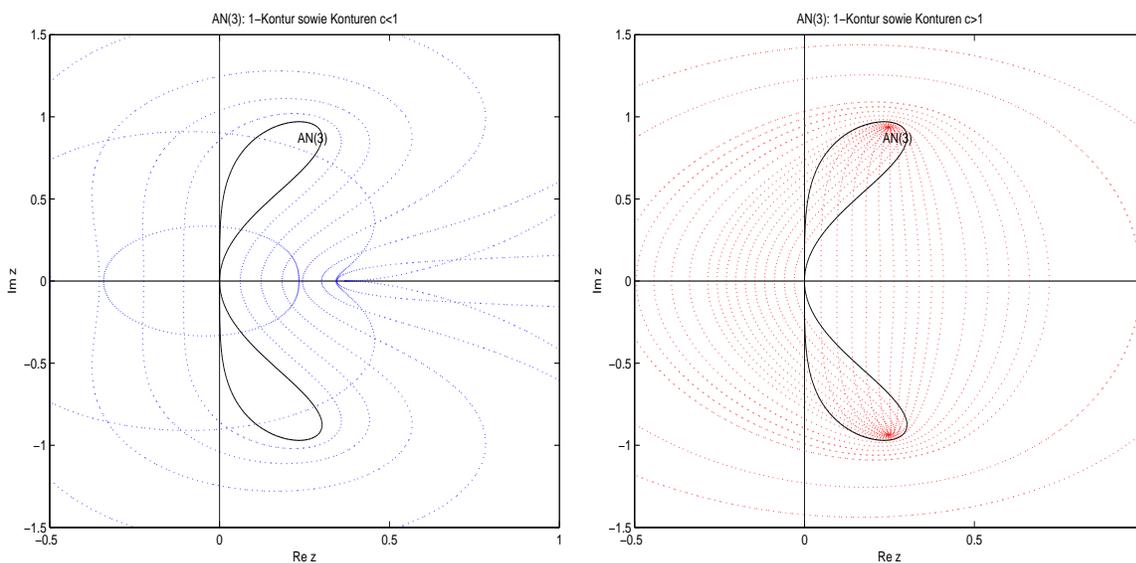


Abb. 4.28: Dateien *ny_stab3a.ps*, *ny_stab3b.ps*, AN(3) mit Konturen $\partial\mathcal{E}$
 bei $c = 0.1(0.1)1$ (links) und bei $c = 1(0.05)1.5(0.1)2(0.5)3$ (rechts)

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 gleichen Betrag sind.

So gelten $\xi_1(z) = \overline{\xi_3(z)}$ bei $z < -0.800475$ sowie $\xi_2(z) = \overline{\xi_3(z)}$ bei $z > 0.367430$.

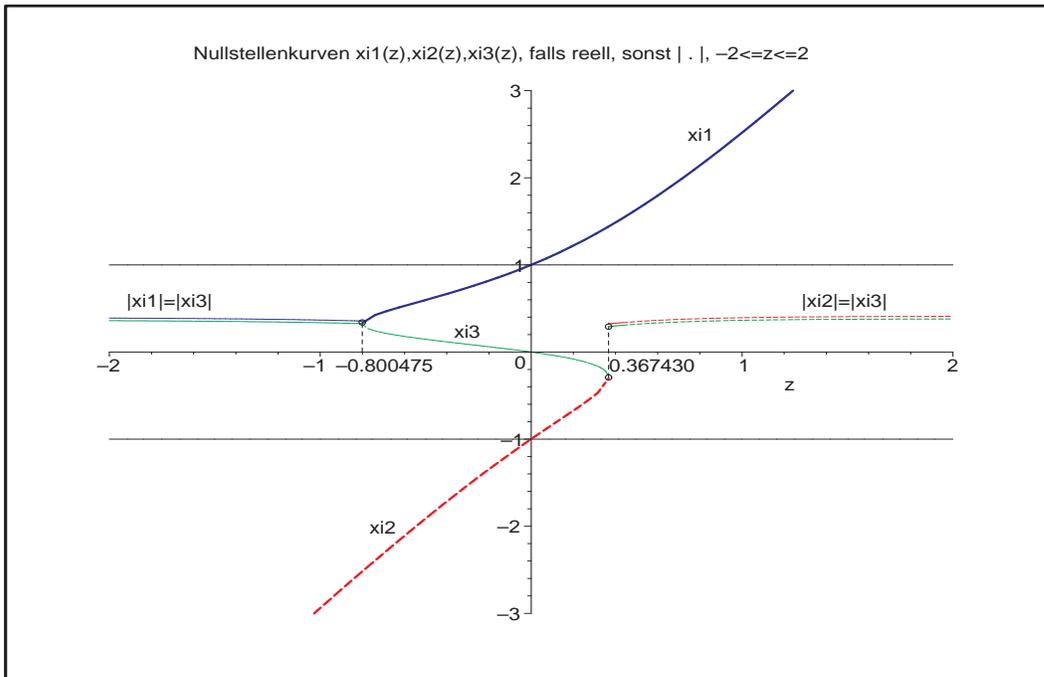


Abb. 4.29: Datei *ny_stab05.ps*, AN(3), Nullstellenkurven $\xi_{1,2,3}(z)$, $z \in [-2, 2]$, dort wo $\xi_k(z) \in \mathbb{C}$, ist $|\xi_k(z)|$ gezeichnet

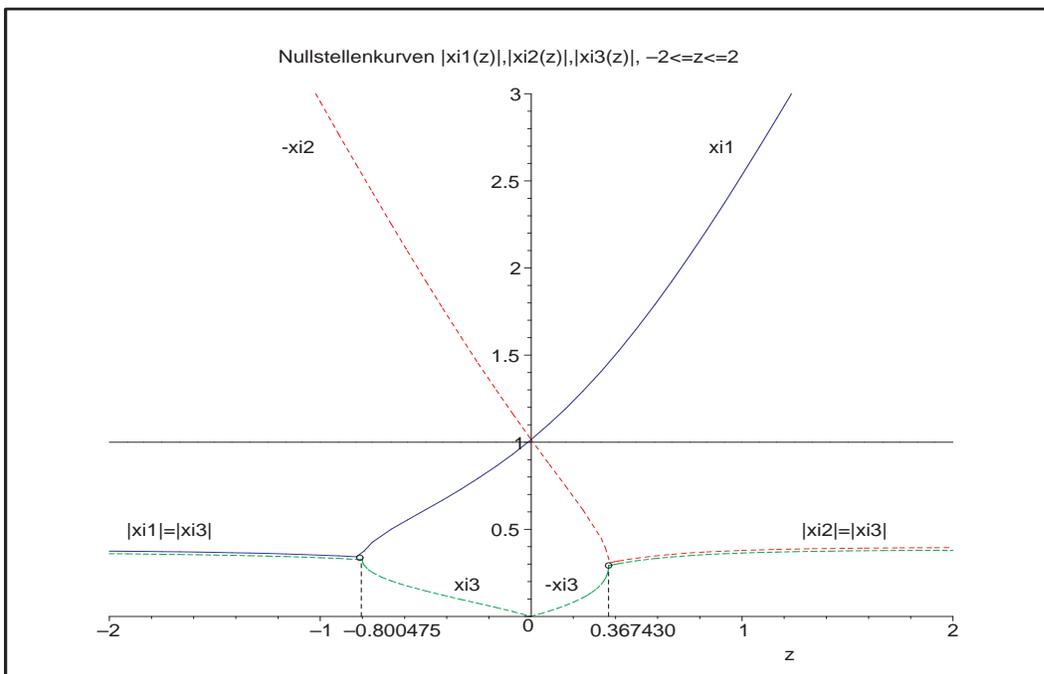


Abb. 4.30: Datei *ny_stab06.ps*, AN(3), Nullstellenkurven $|\xi_{1,2,3}(z)|$, $z \in [-2, 2]$

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 [3.17] zu $\rho(\xi) = \xi^4 - \xi^2$ ist erfüllt, denn seine vier Nullstellen sind $1, 0, 0, -1$ und vom Betrag ≤ 1 .

Weitere Untersuchungen ergeben eine Randkontur $\partial\tilde{\mathcal{E}}$ in Form einer Doppelschleife zu einem Gebiet $\tilde{\mathcal{E}}$ in \mathbb{C}^+ und Bemerkungen wie zu AN(3). AN(4) hat ebenfalls ein leeres Stabilitätsgebiet. In der Abb. 4.27 ist die Randkontur $\partial\tilde{\mathcal{E}}$ von AN(4) schon eingetragen worden.

Weitere Konturen zeigen die folgenden Grafiken.

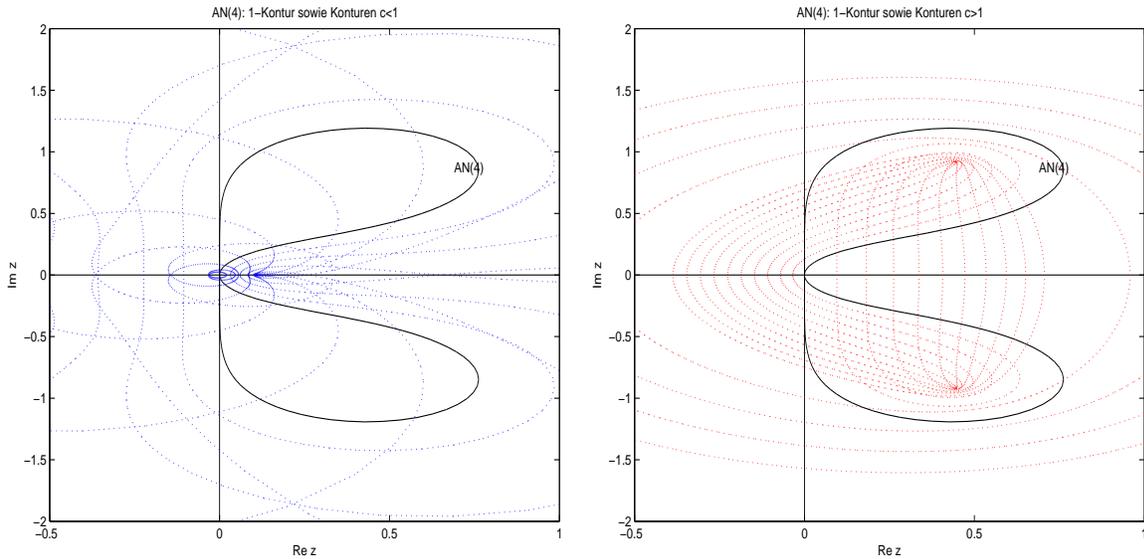


Abb. 4.31: Dateien *ny_stab4a.ps*, *ny_stab4b.ps*, AN(4) mit Konturen $\partial\mathcal{E}$ bei $c = 0.1(0.1)1, 0.55$ (links) und bei $c = 1(0.1)2(0.5)4$ (rechts)

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 Nullstelle $\xi_3(z)$. 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.

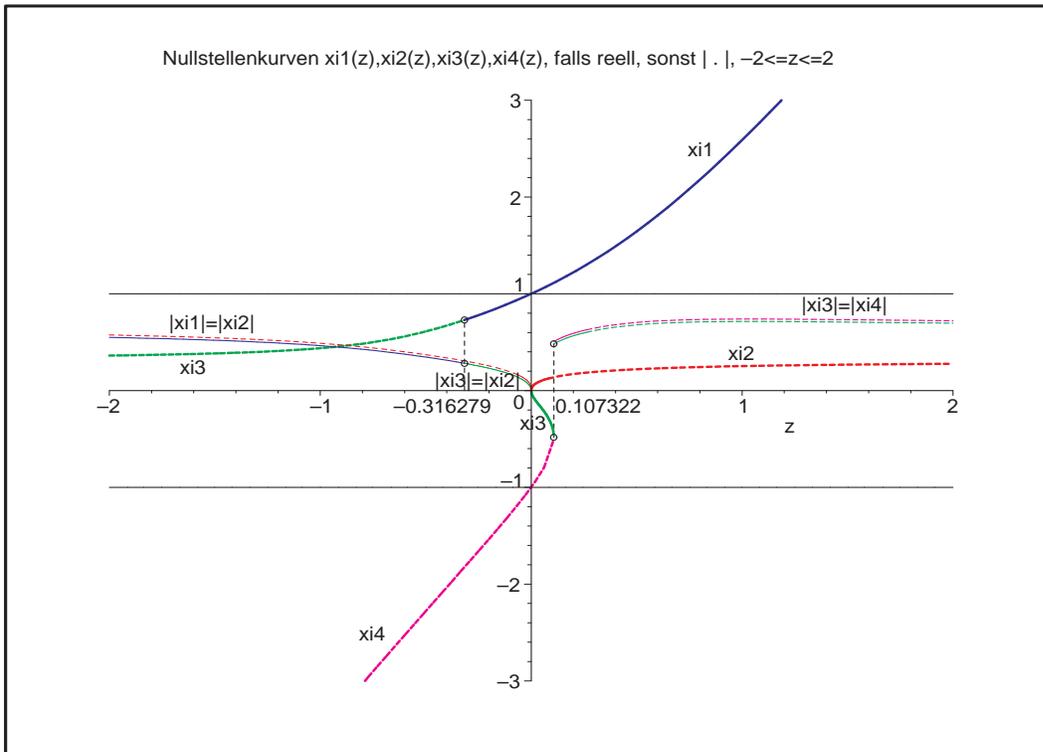


Abb. 4.32: Datei *ny_stab07.ps*, Nullstellenkurven $\xi_{1,2,3,4}(z)$, $z \in [-2, 2]$, dort wo $\xi_k(z) \in \mathbb{C}$, ist $|\xi_k(z)|$ gezeichnet

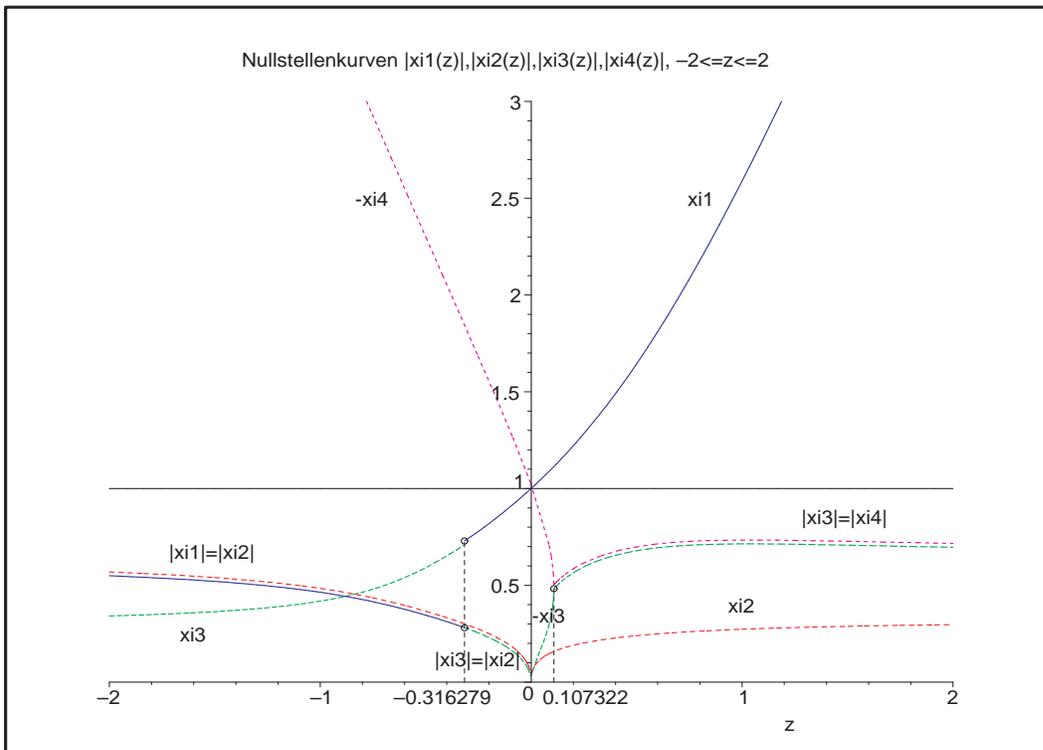


Abb. 4.33: Datei *ny_stab08.ps*, Nullstellenkurven $|\xi_{1,2,3,4}(z)|$, $z \in [-2, 2]$, dort wo $\xi_k(z) \in \mathbb{R}^-$, ist $|\xi_k(z)| = -\xi_k(z)$ gezeichnet

(4) Implizites n -stufiges BDF-Verfahren n -ter Ordnung

Die Koeffizienten der BDF(n), $n = 1, 2, \dots, 6$, entnehmen wir der Tabelle 3.6.

```
% 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;
    ...
end
```

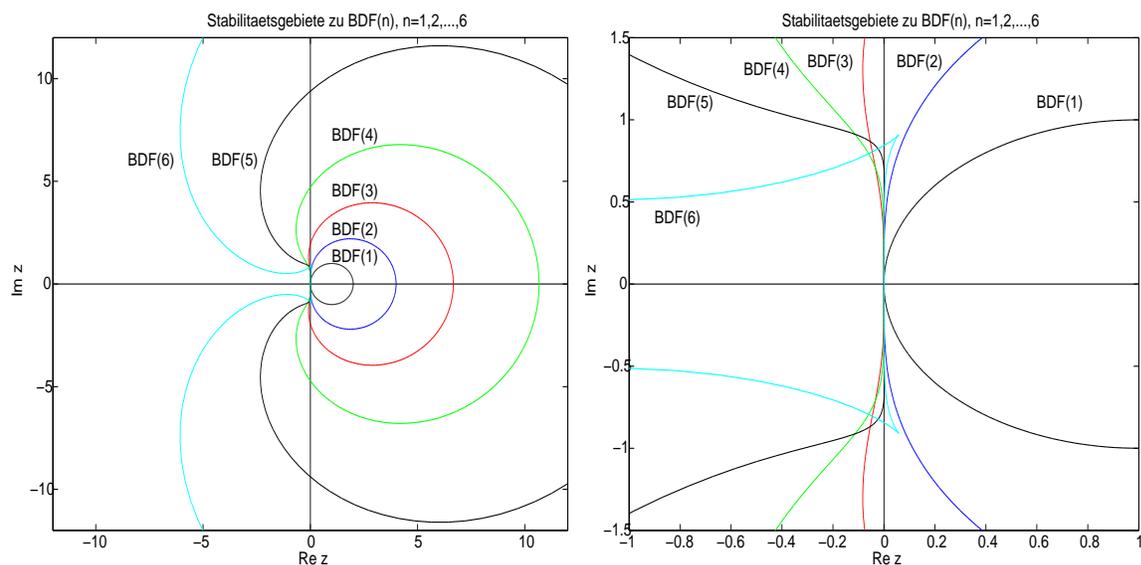


Abb. 4.34: Dateien *bdf_stab1.ps*, *bdf_stab2.ps*, Stabilitätsgebiete \mathcal{E} (rechts mit Vergrößerung nahe $z = 0$) zu BDF(n), $n = 1, 2, \dots, 6$

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-Formeln können auf entsprechende Beschränkungen der Integrationsschrittweite h führen.

Maple-Berechnung und Grafik

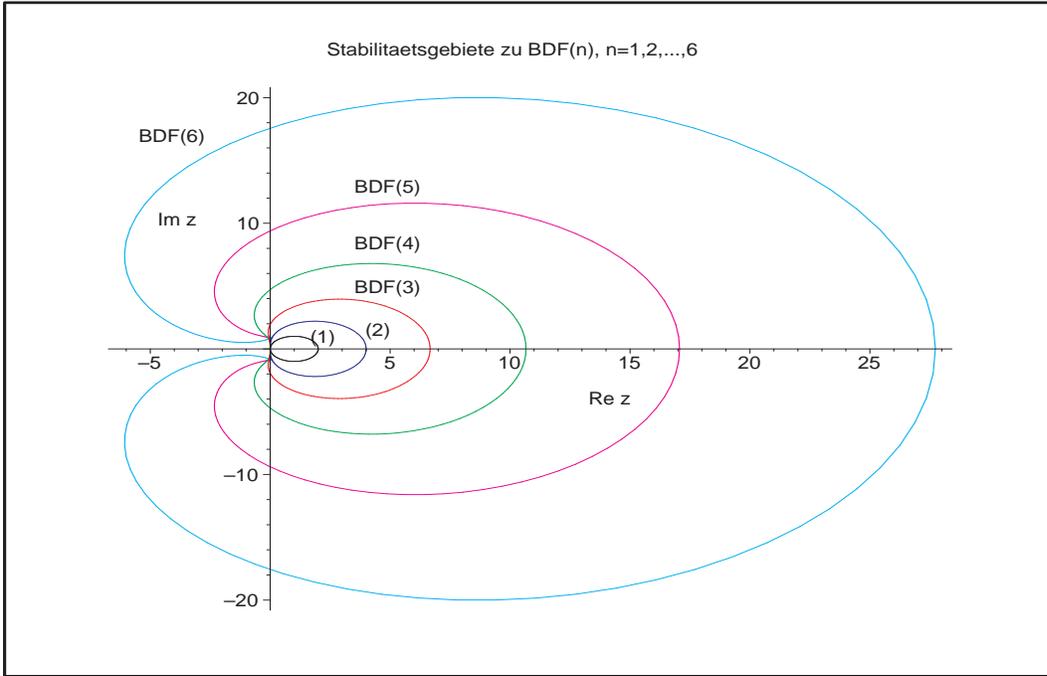


Abb. 4.35: Datei *bdf_stab01.ps*, Stabilitätsgebiete \mathcal{E} zu BDF(n), $n = 1, 2, \dots, 6$, außerhalb der Ränder $\partial\mathcal{E}$

Die BDF-Formeln 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 beginnt.

Zur A-Stabilität in den Definitionen [4.3] und [4.8] machen wir die folgende Ergänzungen.

[4.9] Definition (a) Ein MSV (3.49) heißt $A(\alpha)$ -stabil, falls für einen Winkel α mit $0 < \alpha \leq \frac{\pi}{2}$ gilt

$$\mathcal{E} \supseteq \mathbb{C}_\alpha = \{z \in \mathbb{C}^- : |\pi - \arg(z)| < \alpha\}, \tag{4.28}$$

$A(90^\circ)$ -Stabilität ist damit äquivalent zur A-Stabilität,

(b) $A(0)$ -stabil, falls ein α_0 existiert, so dass 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, so dass

$$\mathcal{E} \supseteq \mathbb{C}_\gamma = \{z \in \mathbb{C}^- : \Re z \leq \gamma\}, \tag{4.29}$$

(d) steif-stabil, falls das MSV $A(0)$ -stabil und $A(\gamma)$ -stabil ist.

Zur A-Stabilität gibt es einen allgemeinen Satz.

[4.10] Satz G. Dahlquist²² 1956, 1963

Betrachtet werden r -stufige lineare MSV (3.49) ($r \geq 1$).

- (1) Kein explizites MSV ist A-stabil.
- (2) A-stabile implizite MSV habe höchstens die Konvergenzordnung 2.

Für eine höhere Konvergenzordnung ist nur $A(\alpha)$ -Stabilität erreichbar.

Bezüglich des Satzes und der Definition [4.9] lassen sich die ersten BDF(n) wie folgt einordnen.

BDF(n)	A-stabil	A(α)-stabil	A(γ)-stabil	A(0)-stabil
n		α	γ	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. 4.1: Stabilität von BDF(n), $n = 1, 2, \dots, 6$ (Dezimalzahlen abgeschnitten)

Das BDF(2) ist A-stabil, was mittels der charakteristischen Polynome

$$\begin{aligned} \phi(z) &= \rho(\xi) - z\sigma(\xi) = (3\xi^2 - 4\xi + 1) - z(2\xi^2) \\ &= (3 - 2z)\xi^2 - 4\xi + 1 = (3 - 2z) \left(\xi^2 - \frac{4}{3-2z}\xi + \frac{1}{3-2z} \right) \end{aligned}$$

oder der Nullstellen dieser gezeigt wird. Die 1-Kontur liegt in \mathbb{C}^+ , weil die zu $\xi = e^{i\varphi}$, $\varphi \in [0, 2\pi)$, gehörigen Werte $z = \rho(\xi)/\sigma(\xi)$ nur positiven Realteil haben, die einzige Ausnahme bildet $\varphi = 0$ mit $\xi = 1$ und $z = 0$. Wir rechnen also

$$\begin{aligned} z &= \frac{\rho(\xi)}{\sigma(\xi)} = \frac{3\xi^2 - 4\xi + 1}{2\xi^2} = \frac{3e^{2i\varphi} - 4e^{i\varphi} + 1}{2e^{2i\varphi}} = \frac{3}{2} - 2e^{-i\varphi} + \frac{1}{2}e^{-2i\varphi} \\ &= \frac{3}{2} - 2[\cos(\varphi) - i\sin(\varphi)] + \frac{1}{2}[\cos(2\varphi) - i\sin(2\varphi)] \\ &= \frac{3}{2} - 2\cos(\varphi) + \frac{1}{2}(2\cos^2(\varphi) - 1) + i[2\sin(\varphi) - \sin(\varphi)\cos(\varphi)] \\ &= [1 - \cos(\varphi)]^2 + i\sin(\varphi)[2 - \cos(\varphi)] \\ &= \Re z + i\Im z, \end{aligned}$$

so dass $\Re z = [1 - \cos(\varphi)]^2 > 0$, $\varphi \neq 0$.

²²Germund Dalquist (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

$$\xi_{1,2} = \frac{2 \pm \sqrt{1 + 2z}}{3 - 2z}.$$

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 insbesondere 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.

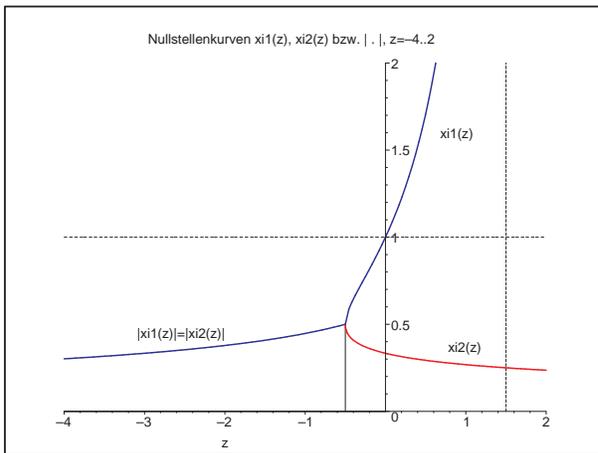


Abb. 4.36:

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 vorgegebene 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 $\rho(\xi)/(\xi - 1)$, dessen weitere reelle oder konjugiert komplexe Nullstellen betragsmäßig kleiner Eins sind.

BDF(n)	reduziertes Polynom $\rho(\xi)/(\xi - 1)$ 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. 4.2: Nullstellen zu 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 $z = \alpha_r/\beta_r$ und spezielle Punkte hervorgehoben.

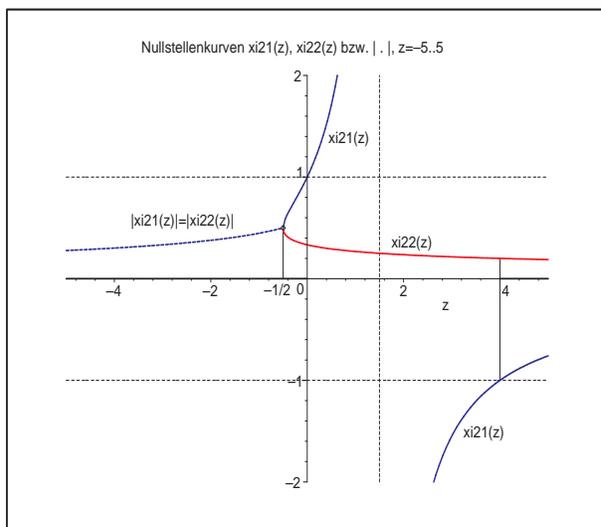


Abb. 4.37:

Datei *bdf_stab22.ps*,
BDF(2): Verlauf
der Nullstellenfunktionen
 $\xi_{1,2}(z)$ bzw. $|\xi_{1,2}(z)|$,
Asymptote bei $z = \frac{3}{2}$

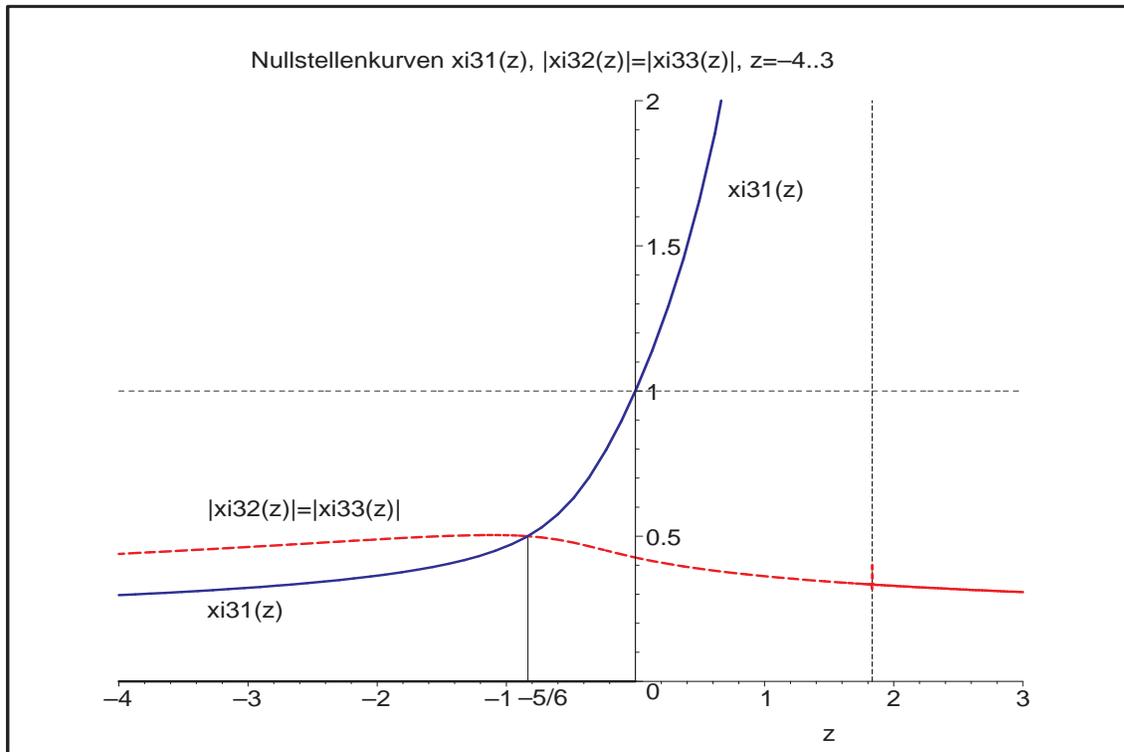


Abb. 4.38: Datei *bdf_stab23.ps*, BDF(3): Verlauf der Nullstellenfunktionen $\xi_1(z) \in \mathbb{R}$ und $|\xi_2(z)| = |\xi_3(z)|$ bei $\xi_{2,3}(z) \in \mathbb{C}$, Asymptote bei $z = \frac{11}{6}$

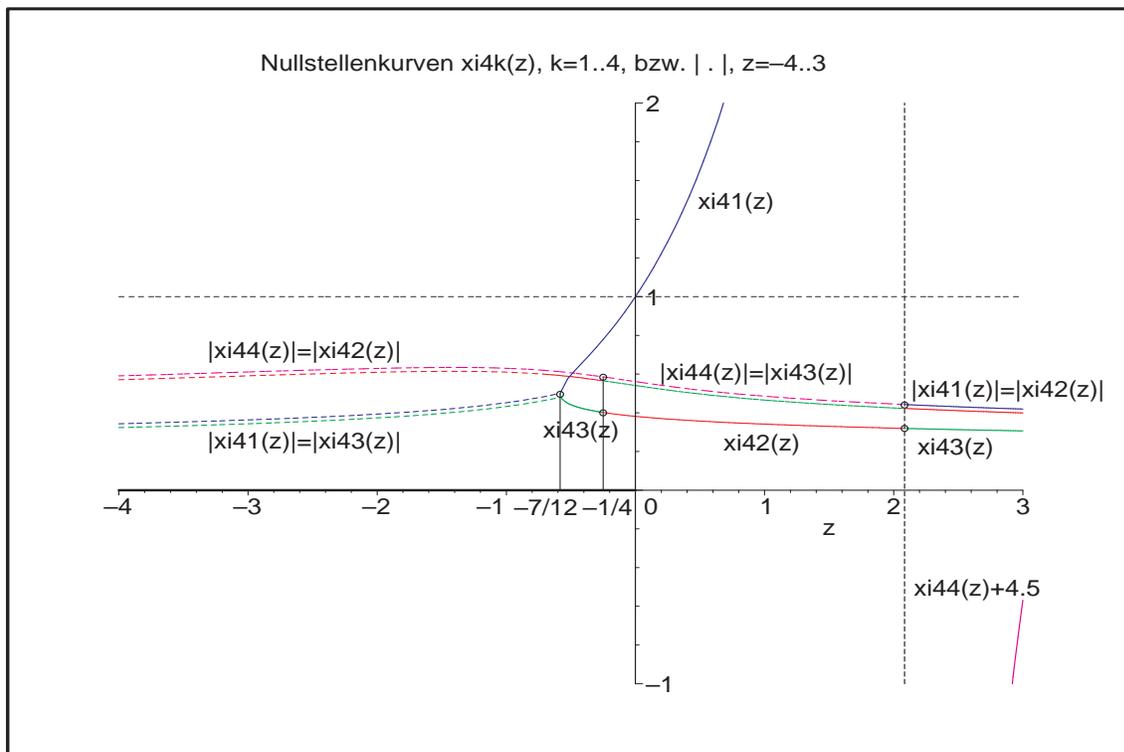


Abb. 4.39: Datei *bdf_stab24.ps*, BDF(4): Verlauf der Nullstellenfunktionen $\xi_k(z) \in \mathbb{R}$ und $|\xi_k(z)|$ bei $\xi_k(z) \in \mathbb{C}$, Asymptote bei $z = \frac{25}{12}$

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. die einseitigen Grenzwerte

$$|\xi_1(-\frac{7}{12}-)| = |\xi_3(-\frac{7}{12}-)| = \xi_1(-\frac{7}{12}+) = \xi_3(-\frac{7}{12}+) = \frac{1}{2},$$

$$\xi_3(-\frac{1}{4}-) = \xi_2(-\frac{1}{4}+) = 0.401, \quad |\xi_4(-\frac{1}{4})| = |\xi_2(-\frac{1}{4}-)| = |\xi_3(-\frac{1}{4}+)| = 0.585,$$

$$|\xi_4(\frac{25}{12}-)| = |\xi_3(\frac{25}{12}-)| = |\xi_1(\frac{25}{12}+)| = |\xi_2(\frac{25}{12}+)| = 0.442, \quad \xi_2(\frac{25}{12}-) = \xi_3(\frac{25}{12}+) = 0.319.$$

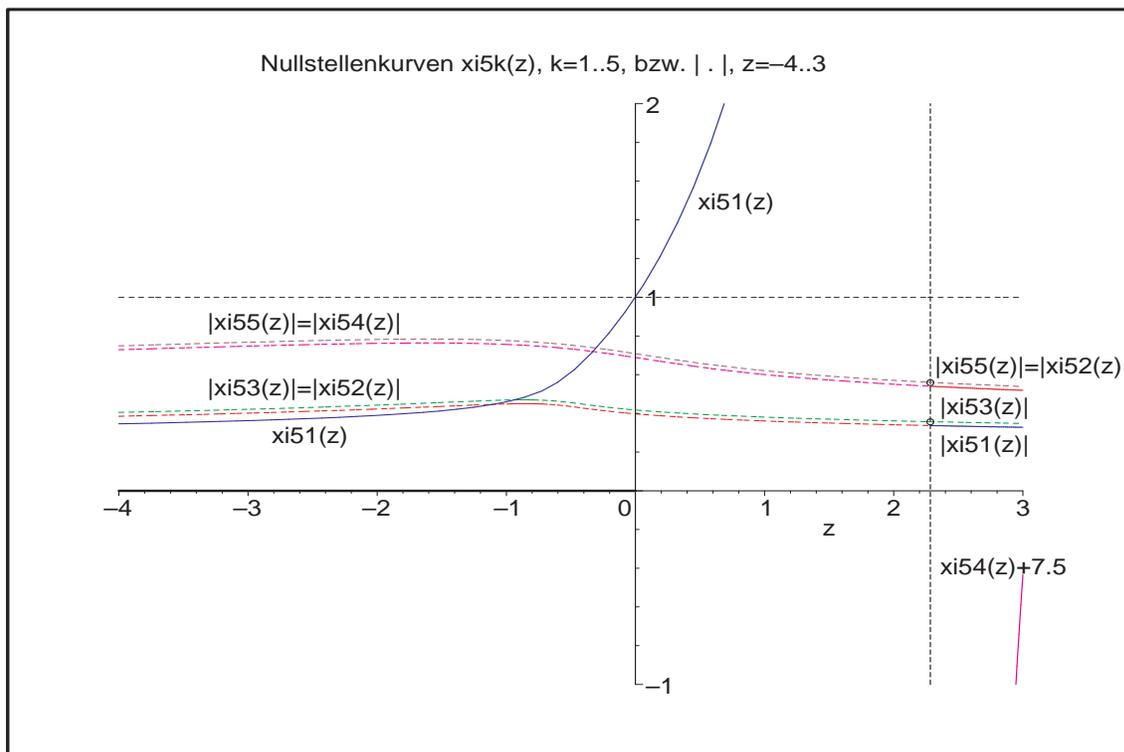


Abb. 4.40: Datei *bdf_stab25.ps*, BDF(5): Verlauf der Nullstellenfunktionen $\xi_k(z) \in \mathbb{R}$ und $|\xi_k(z)|$ bei $\xi_k(z) \in \mathbb{C}$, Asymptote bei $z = \frac{137}{60}$

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 bezüglich der Art der Stabilität und illustrieren die zugehörigen Gebiete.

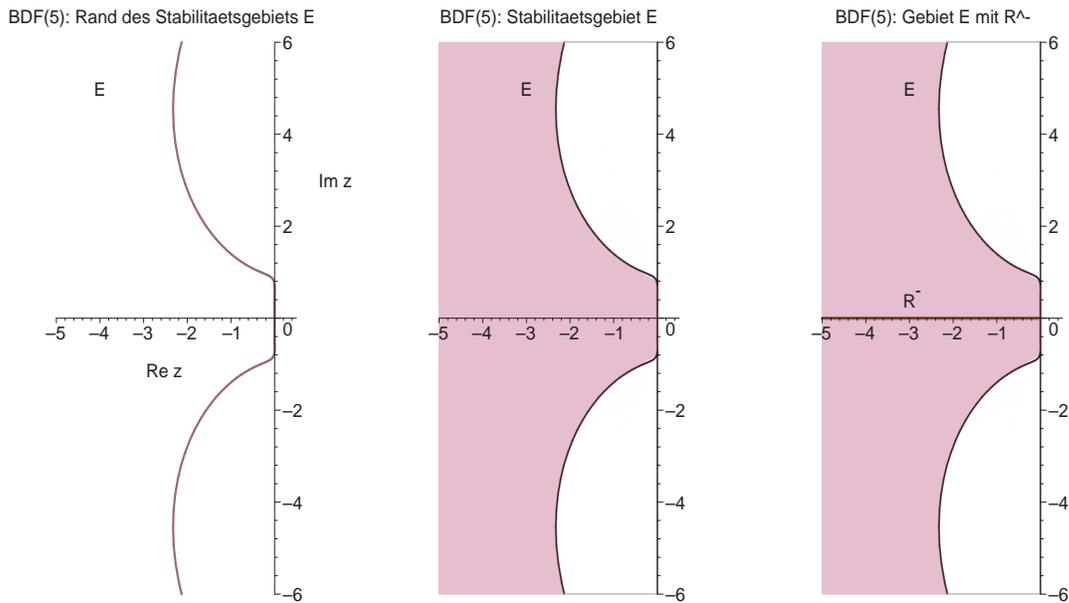


Abb. 4.41: Dateien *bdf_stab02.ps*, *bdf_stab03.ps*, *bdf_stab04.ps*, BDF(5):
Stabilitätsgebiet \mathcal{E} , Rand $\partial\mathcal{E}$ sowie mit Stab.intervall $\mathbb{R}^- = (-\infty, 0)$

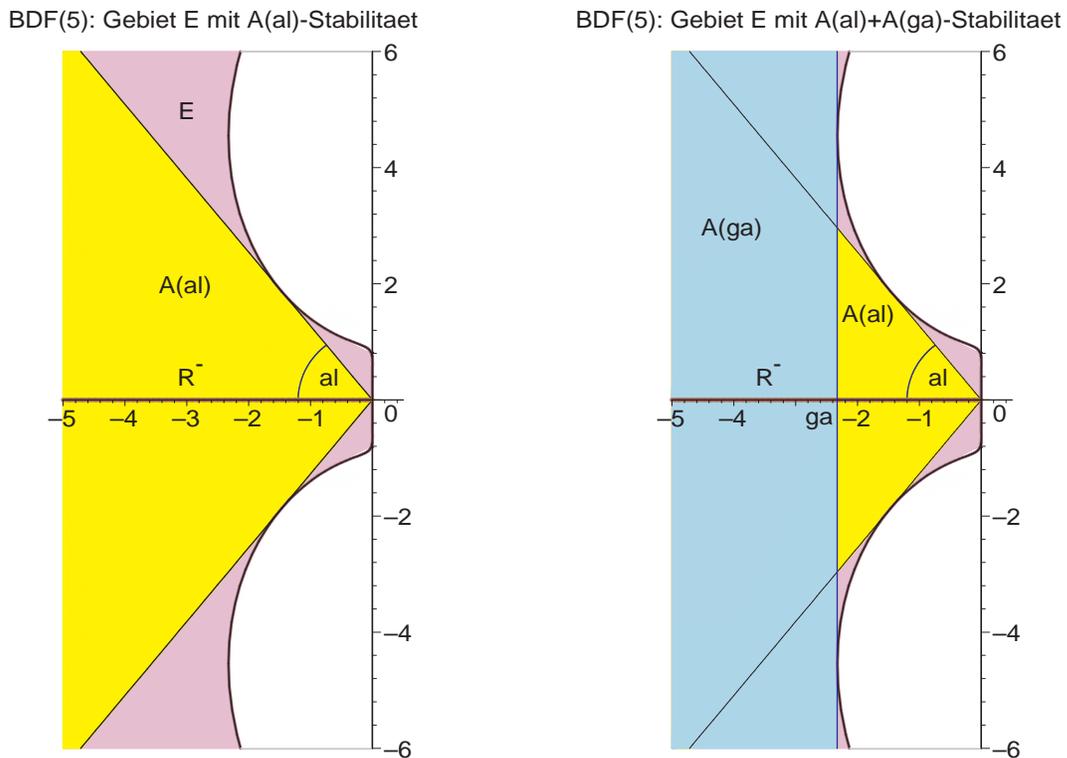


Abb. 4.42: Dateien *bdf_stab05.ps*, *bdf_stab06.ps*,
BDF(5): Stabilitätsgebiet \mathcal{E} sowie $A(\alpha)$ - und $A(\gamma)$ -Stabilität

Verbleiben noch die Nullstellenfunktionen von BDF(6).

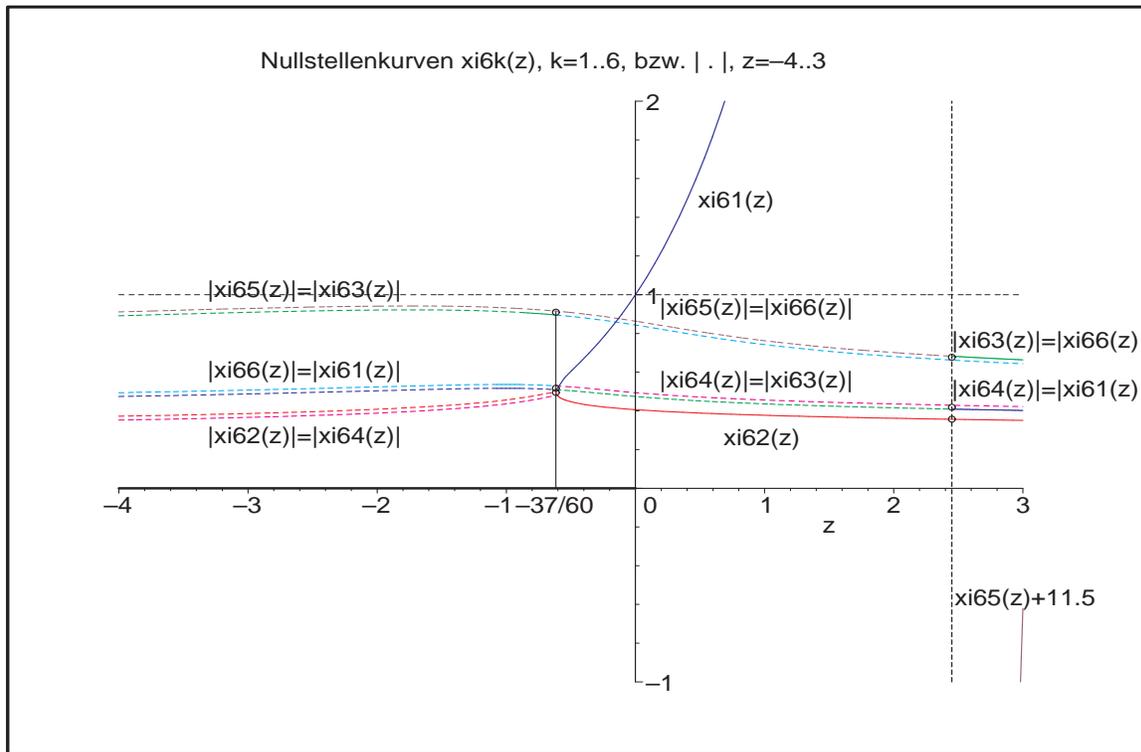


Abb. 4.43: Datei *bdf_stab26.ps*, BDF(6): Verlauf der Nullstellenfunktionen $\xi_k(z) \in \mathbb{R}$ und $|\xi_k(z)|$ bei $\xi_k(z) \in \mathbb{C}$, Asymptote bei $z = \frac{147}{60}$

Die Nullstellenkurven zu BDF(6) sind in ihrem Verhalten ähnlich wie bei BDF(4), nur dass noch ein Paar konjugiert komplexer hinzukommt.

Ansonsten gilt bez. der Nullstellen das bisher Gesagte.

$$\begin{aligned}
 |\xi_2(-\frac{37}{60}-)| &= |\xi_4(-\frac{37}{60}-)| = \xi_1(-\frac{37}{60}+) = \xi_2(-\frac{37}{60}+) = \frac{1}{2}, \\
 |\xi_6(-\frac{37}{60}-)| &= |\xi_1(-\frac{37}{60}-)| = |\xi_4(-\frac{37}{60}+)| = |\xi_3(-\frac{37}{60}+)| = 0.509, \\
 |\xi_5(-\frac{37}{60})| &= |\xi_3(-\frac{37}{60}-)| = |\xi_6(-\frac{37}{60}+)| = 0.915, \\
 \xi_2(\frac{147}{60}) &= 0.357, \\
 |\xi_4(\frac{147}{60})| &= |\xi_3(\frac{147}{60}-)| = |\xi_1(\frac{147}{60}+)| = 0.409, \\
 |\xi_6(\frac{147}{60})| &= |\xi_5(\frac{147}{60}-)| = |\xi_3(\frac{147}{60}+)| = 0.683.
 \end{aligned}$$

[4.11] Übung Schreiben Sie ein Programm zur Berechnung der komplexwertigen Funktion $z(\varphi)$, definiert durch (4.27). Stellen Sie die Stabilitätsgebiete für weitere MSV grafisch dar.

4.3 Integration steifer Systeme

Die Überlegungen der Abschnitte 4.1 und 4.2 liefern Obergrenzen an die Schrittweite h , unter welchen Instabilitäten unterdrückt werden. Günstig ist es natürlich bei geringen Genauigkeitsanforderungen, gleich A-stabile Verfahren zu verwenden. Bei hoher Genauigkeit, also kleiner Toleranz, ist die Schrittweite zumeist sehr klein zu wählen, so dass man damit i. Allg. unter der maximal zulässigen Grenze bleibt.

In der Test-DGL (4.1) 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 [4.4] (a) mit der Formel (4.12) ein wichtiges Ergebnis erhalten.

In homogenen linearen Systemen mit einer Matrix, die Eigenwerte mit betragskleinen Imaginärteilen hat, bestimmt der Eigenwert λ_i mit dem betragsgrößten negativen Realteil die Schrittweite (vergl. hierzu auch die Merkregel [1.12]).

Im Folgenden wollen wir an Beispielen das Konzept einer Schrittweitensteuerung entwerfen, welche zu einer möglichst genauen Approximation unter möglichst 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 [4.3]). 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 eventuell Monotonieeigenschaften erfüllt werden sollen, so führt das gegebenenfalls zu einer weiteren Einschränkung der Schrittweite.

gDGL mit einem moderaten Lösungsverlauf werden als nicht steif eingestuft. Die Steifheit dagegen drückt sich in einer starken Veränderung des Lösungsverlaufs (Lösungsfunktion $x(t)$, Ableitung $x'(t)$, $|f_x(t, x)|$ u. ä.) aus.

Wir betrachten zuerst gDGL 2. Ordnung bzw. SysgDGL, die nicht steif sind, um dann den Charakter der Steifheit besser zu deuten. Berechnungen und Grafiken sind in Maple.

[4.12] Beispiele

(a) Gegeben sei die homogene lineare gDGL 2. Ordnung

$$x''(t) + 5x'(t) + 6x(t) = 0. \tag{4.30}$$

Die charakteristischen Wurzeln sind $\lambda_1 = -2$ und $\lambda_2 = -3$, das Fundamentalsystem $\{e^{-2t}, e^{-3t}\}$, so dass 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);
```

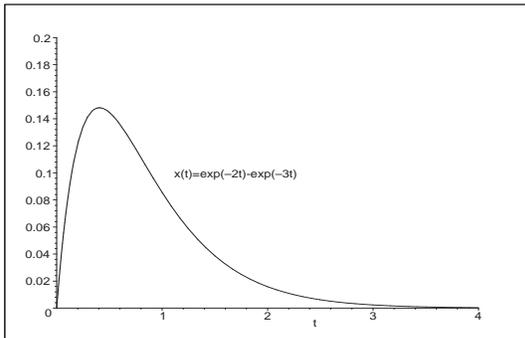


Abb. 4.44:

Datei *nsteif01.ps*, 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);
```

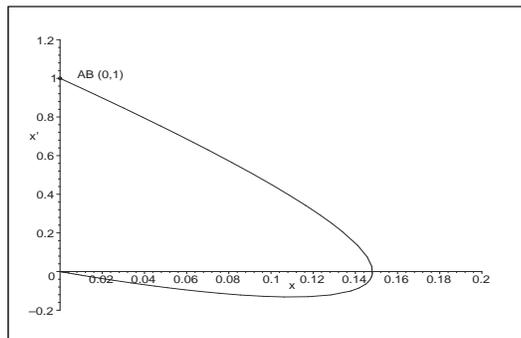
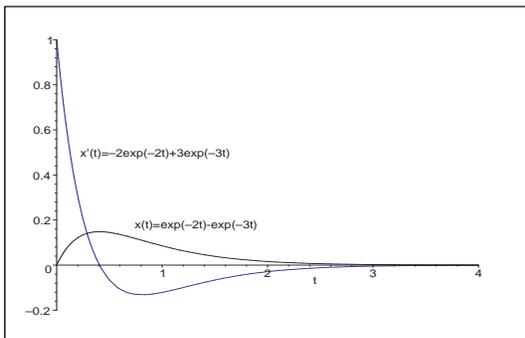


Abb. 4.45: Dateien *nsteif02.ps*, *nsteif03.ps*,

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

$$x''(t) + x'(t) - x(t) = 0, \quad x(0) = 1, \quad x'(0) = 0. \tag{4.31}$$

Die charakteristischen Wurzeln sind $\lambda_{1,2} = \frac{1}{2}(-1 \pm \sqrt{5})$.

Die Lösung und ihre Ableitung lauten

$$\begin{aligned} x(t) &= \frac{1}{10}(5 + \sqrt{5}) e^{t(-1+\sqrt{5})/2} + \frac{1}{10}(5 - \sqrt{5}) e^{t(-1-\sqrt{5})/2}, \\ &= 0.723\,606\,798 e^{0.618033\,988t} + 0.276\,393\,202 e^{-1.618033\,988t}, \\ x'(t) &= \frac{\sqrt{5}}{5}(e^{t(-1+\sqrt{5})/2} - e^{t(-1-\sqrt{5})/2}) \\ &= 0.447\,213\,595 (e^{0.618033\,988t} - e^{-1.618033\,988t}). \end{aligned}$$

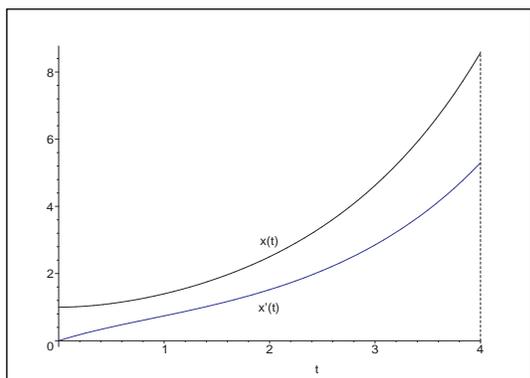


Abb. 4.46:
Datei *nsteif04.ps*,
AWP $x''(t) + x'(t) - x(t) = 0$,
 $x(0) = 1, x'(0) = 0$, sowie
Lösung $x(t)$ und Ableitung $x'(t)$ in $[0, 4]$

Es gilt $x(4) = 8.573\,522\,382$, $x'(4) = 5.297\,772\,797$, so dass man bei $t \in [0, 4]$ für das Phasenportrait $(x(t), x'(t))$ den Bereich $[0, 9] \times [0, 6]$ nehmen kann.

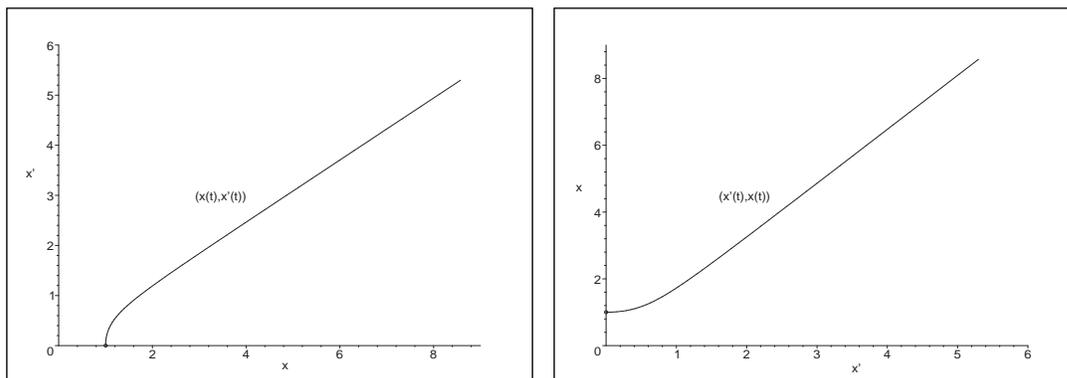


Abb. 4.47: Dateien *nsteif05.ps*, *nsteif06.ps*,
Phasenportraits $(x(t), x'(t))$ und $(x'(t), x(t))$ im jeweiligen Bereich

Man notiert die gDGL bei $x' = y$ und $\mathbf{x} = (x, x')^T$ auch als System von zwei autonomen linearen gDGL 1. Ordnung gemäß

$$\begin{aligned} x'(t) &= y(t), & x(0) &= 1, \\ y'(t) &= x(t) - y(t), & y(0) &= 0, \\ \mathbf{x}'(t) &= A\mathbf{x}(t), & A &= \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}, \mathbf{x}(0) = (1, 0)^T. \end{aligned} \tag{4.32}$$

Lösungsvarianten für (4.31) bzw. (4.32)

```
> # 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 DGL-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);
```

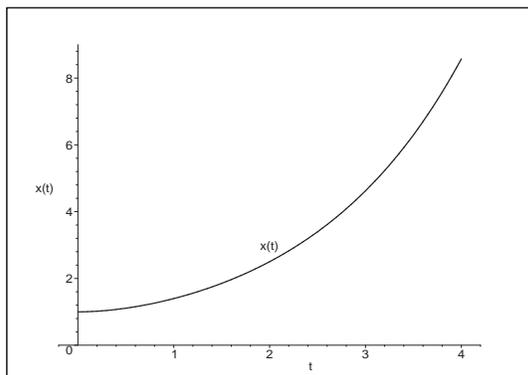


Abb. 4.48:

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 1.1 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 spezielle Kurve.

Zum AWP (4.32) 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);
```

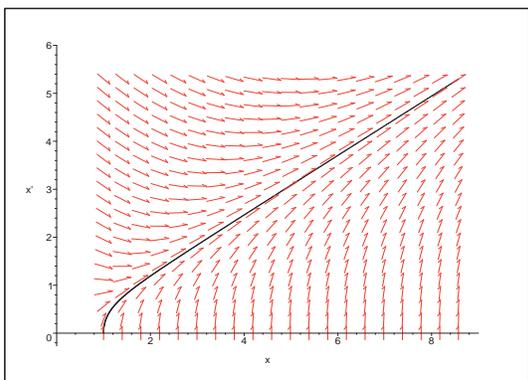


Abb. 4.49:

Datei *nsteif08.ps*,

AWP (4.32),

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 maximale 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 5 AB ausgetestet haben, zeichnen wir nun diese im Richtungsfeld (x, x') .

(x_0, x'_0)	[xmin, xmax] × [ymin, ymax]
(1, 0)	[0, 5.298] × [0, 8.574]
(1, -1)	[0.650, 3.276] × [-1, 2.022]
(0, 2)	[0, 10.596] × [1.300, 6.552]
(0, -2)	[-10.596, 0] × [-6.552, -1.300]
(-2, 2)	[-6.552, -1.300] × [-4.044, 2]

Tab. 4.3:

Bereich der Phasenkurve
mit AB $x(0) = x_0, x'(0) = x'_0$

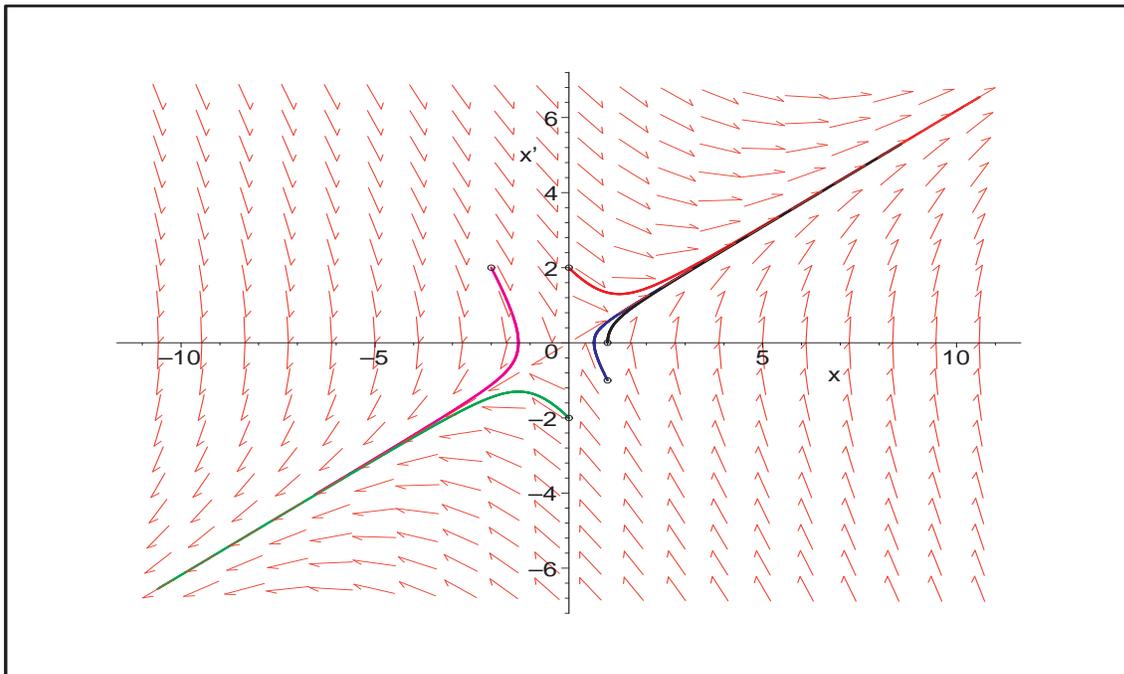


Abb. 4.50: Datei *nsteif10.ps*, AWP (4.32), Phasenkurven $(x(t), x'(t)), t \in [0, 4]$,
mit AB $(x(0), x'(0)) = (1, 0), (1, -1), (0, 2), (0, -2), (-2, 2)$
im Richtungsfeld (x, x') , Kommando `DEplot`

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],[-1,5],[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);
```

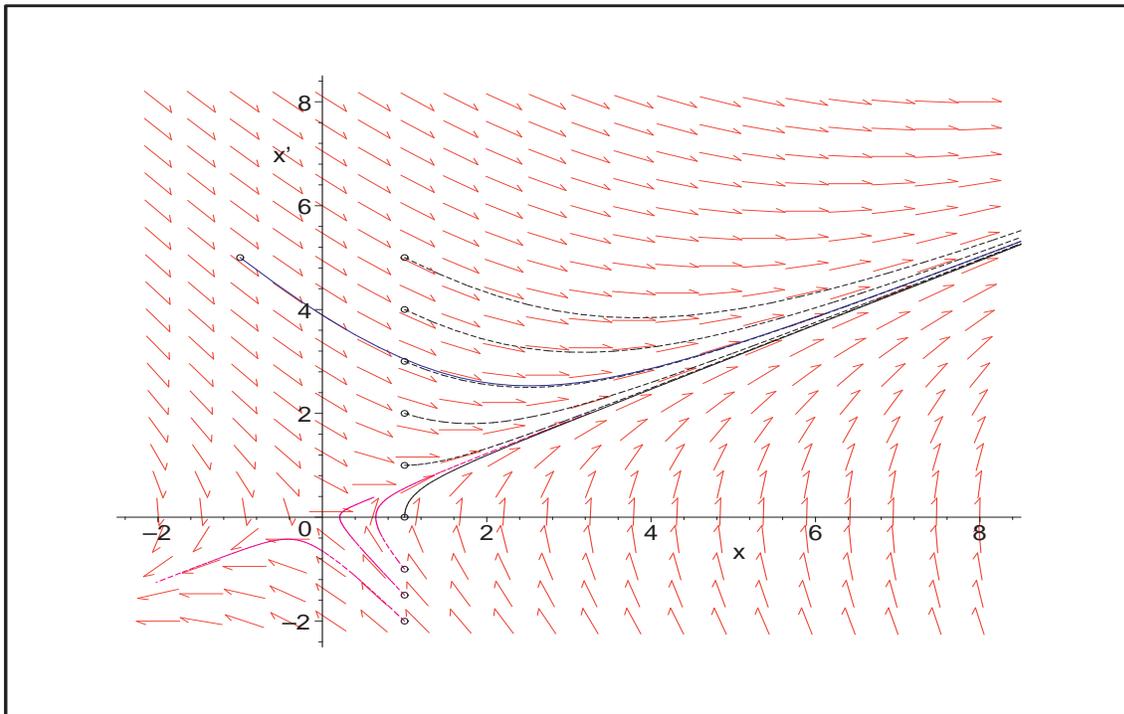


Abb. 4.51: Datei *nsteif12.ps*, AWP (4.32), Richtungsfeld $(x, x') \in [-2, 8] \times [-2, 8]$, Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$, mit AB $(1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, -1), (1, -1.5), (1, -2), (-1, 5)$, Kommando `dfieldplot`

Die Lösungstrajektorien $x(t)$ zu den genannten AB zeichnen wir in einer extra Grafik.

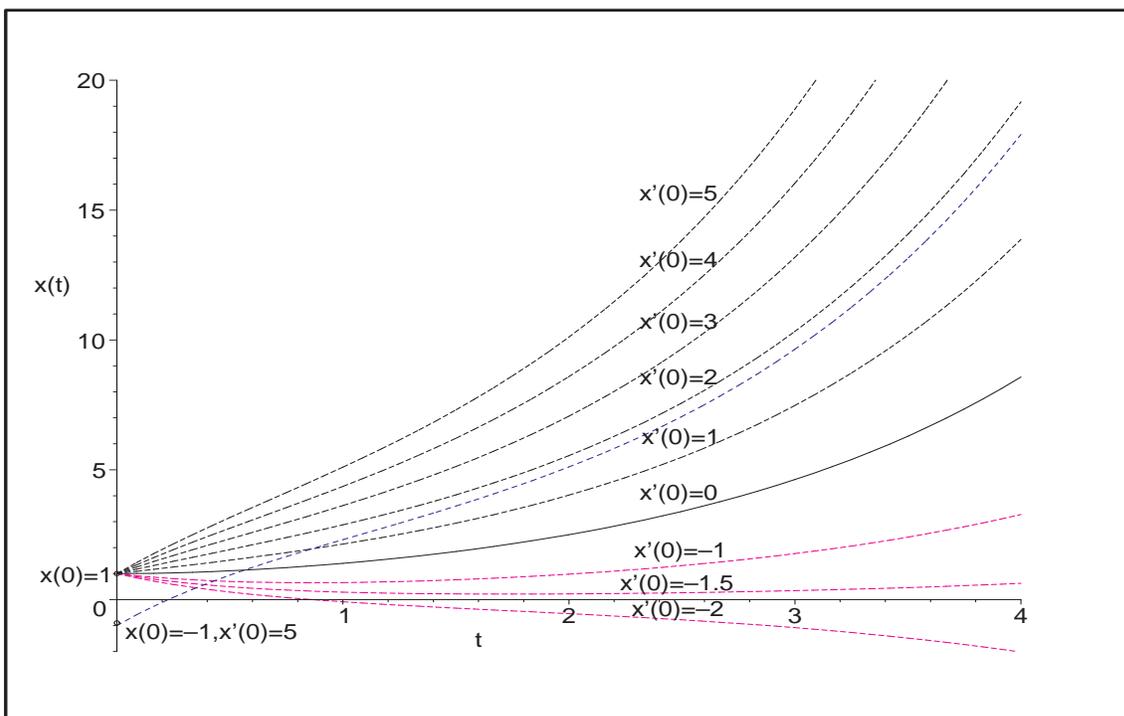


Abb. 4.52: Datei *nsteif11.ps*, AWP (4.32), Lösungskurven $x(t)$, $t \in [0, 4]$, mit AB $(1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, -1), (1, -1.5), (1, -2), (-1, 5)$

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 4.53.

```
> 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);
```

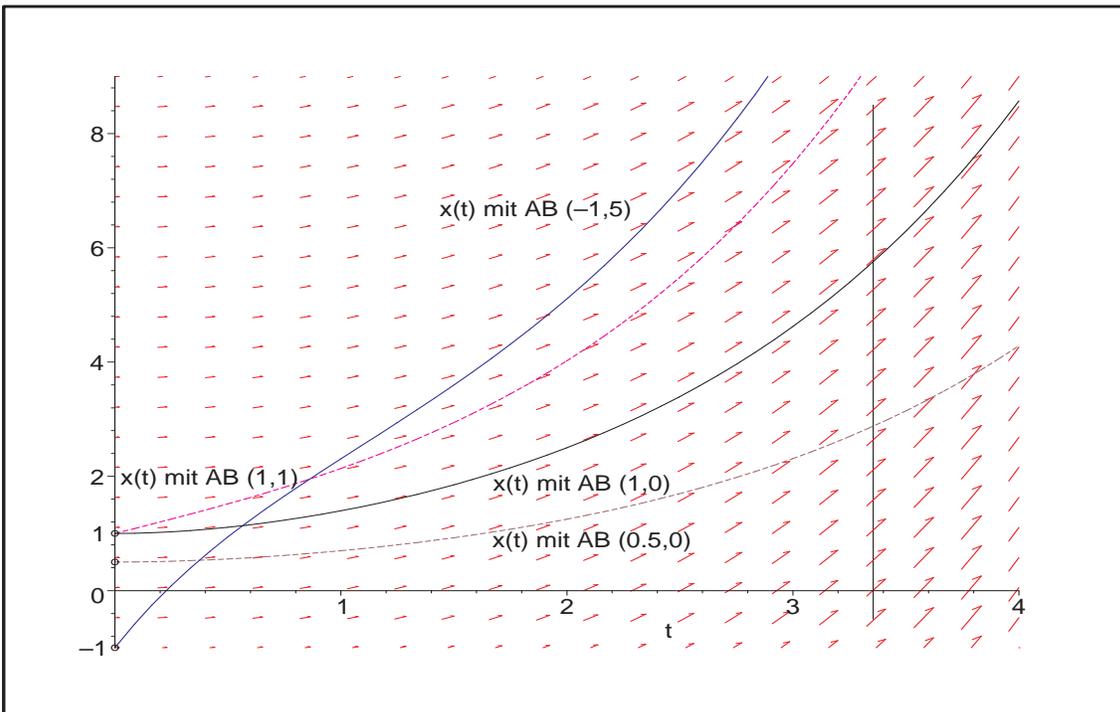


Abb. 4.53: Datei `nsteif13.ps`, AWP (4.32),
 Lösungskurven $x(t), t \in [0, 4]$, mit AB $(1, 0), (1, 1), (0.5, 0), (-1, 5)$,
 "Richtungsfeld" zur Kurve mit AB $x(0) = 1, x'(0) = 0$,
 Kommando `fieldplot`

(c) Gegeben sei die autonome inhomogene lineare gDGL 2. Ordnung als AWP

$$x''(t) + x'(t) - x(t) + 1 = 0, \quad x(0) = 1, \quad x'(0) = 0. \tag{4.33}$$

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 1, die auch die beiden AB erfüllt. Somit erhalten wir die Lösung $x(t) \equiv 1$ und ihre Ableitung 0.

Damit erübrigt sich die grafische Darstellung der Lösung und Ableitung, genauso der Phasenkurve, die in den Punkt (1, 0) zusammenfällt.

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. 4.55.

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 5 AB und verwenden als oberen Bereich $[-20, 3] \times [-12, 3]$

(x_0, x'_0)	[xmin, xmax] × [ymin, ymax]
(1, 0)	[1, 1] × [0, 0]
(1, -1)	[-4.298, 1] × [-3.276, -0.650]
(0, 2)	[0, 3.022] × [0.422, 2]
(0, -2)	[-18.169, 0] × [-11.849, -1.803]
(-2, 2)	[-14.125, -1.500] × [-9.342, 2]

Tab. 4.4:
Bereich der Phasenkurve mit AB $x(0) = x_0, x'(0) = x'_0$

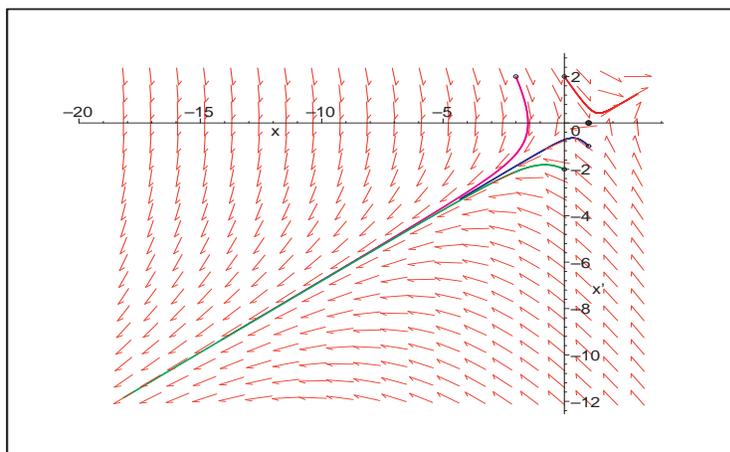


Abb. 4.54:
Datei *nsteif14.ps*,
AWP (4.33),
Phasenkurven $(x(t), x'(t))$,
 $t \in [0, 4]$, mit AB
 $(x(0), x'(0)) = (1, 0), (1, -1),$
 $(0, 2), (0, -2), (-2, 2)$
im Richtungsfeld (x, x') ,
Kommando DEplot

Das Kommando `dfieldplot` verwenden wir zur Anzeige des Richtungsfeldes (x, x') im vorgegebenen Bereich und zeichnen mehrere vorher berechnete Phasenkurven ein.

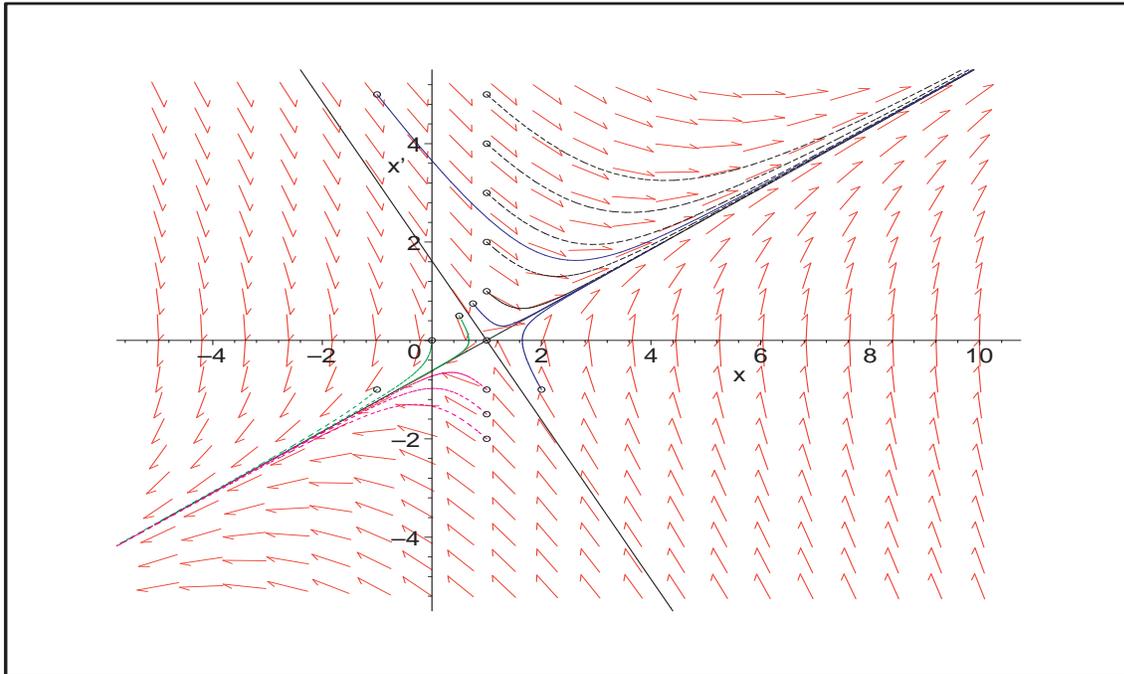


Abb. 4.55: Datei *nsteif16.ps*, AWP (4.33), Richtungsfeld $(x, x') \in [-5, 10] \times [-5, 5]$, Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$, mit AB $(1, 0)$, $(1, 1)$, $(1, 2)$, $(1, 3)$, $(1, 4)$, $(1, 5)$, $(1, -1)$, $(1, -1.5)$, $(1, -2)$, $(-1, 5)$, $(0, 0)$, $(0.5, 0.5)$, $(0.75, 0.75)$, $(2, -1)$

Die Lösungstrajektorien $x(t)$ zu den genannten AB stellen wir in einer extra Grafik dar.

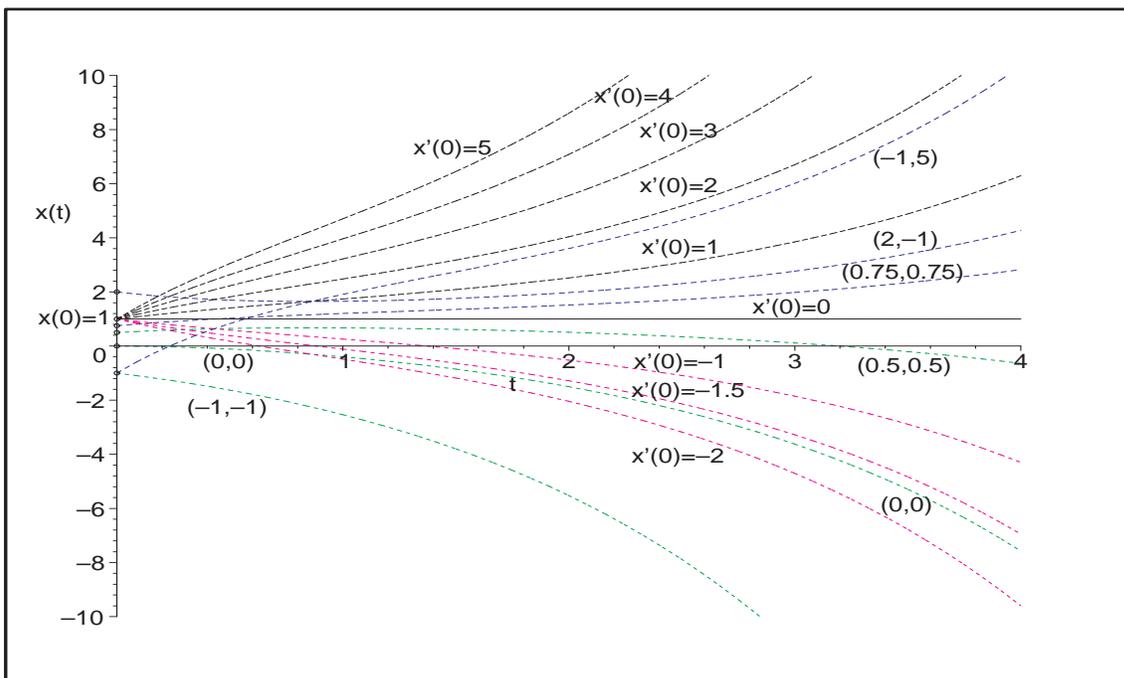


Abb. 4.56: Datei *nsteif15.ps*, AWP (4.33), Lösungskurven $x(t)$, $t \in [0, 4]$, mit AB aus vorheriger Abbildung

(d) Gegeben sei die nicht autonome inhomogene lineare gDGL 2. Ordnung als AWP

$$x''(t) + x'(t) - x(t) + t = 0, \quad x(0) = 1, \quad x'(0) = 0. \quad (4.34)$$

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 $t + 1$, so dass wir die Lösung und ihre Ableitung erhalten.

$$\begin{aligned} x(t) &= 1 + t + \frac{\sqrt{5}}{5} (-e^{t(-1+\sqrt{5})/2} + e^{t(-1-\sqrt{5})/2}), \\ &= 1 + t + 0.447213595 (-e^{0.618033988t} + e^{-1.618033988t}), \\ x'(t) &= 1 - \frac{\sqrt{5}}{10}(-1 + \sqrt{5}) e^{t(-1+\sqrt{5})/2} - \frac{\sqrt{5}}{10}(1 + \sqrt{5}) e^{t(-1-\sqrt{5})/2} \\ &= 1 - 0.276393201 e^{0.618033988t} - 0.723606797 e^{-1.618033988t}. \end{aligned}$$

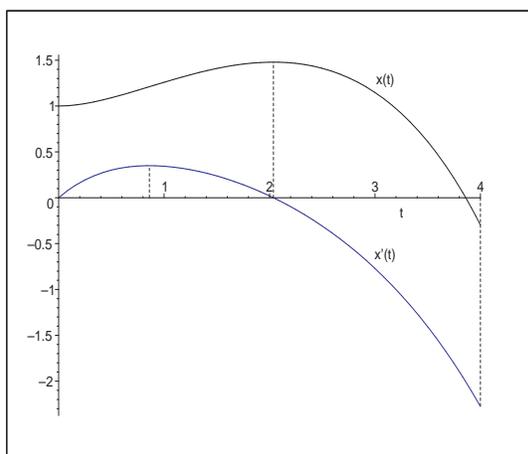


Abb. 4.57:

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 $x(4) = -0.298$, $x'(4) = -2.276$, so dass 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.

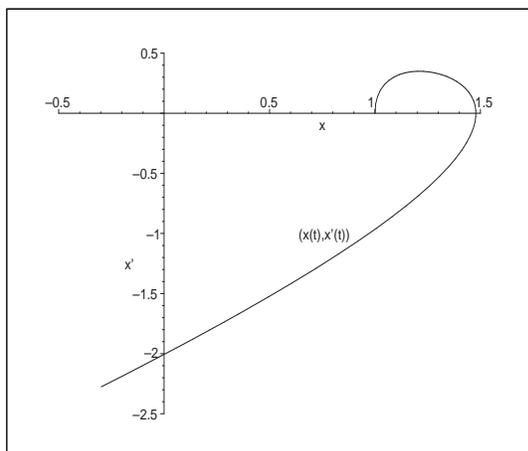


Abb. 4.58:

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

$$\begin{aligned} x'(t) &= y(t), & x(0) &= 1, \\ y'(t) &= x(t) - y(t) - t, & y(0) &= 0, \\ \mathbf{x}'(t) &= A\mathbf{x}(t) + \mathbf{g}(t), & A &= \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}, \mathbf{g}(t) = (0, -t)^T, \mathbf{x}(0) = (1, 0)^T. \end{aligned} \quad (4.35)$$

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:
  fcns:=x(t),y(t):
  init:=x(0)=1,y(0)=0:
  loe4:=dsolve({sys4,init},{fcns});

> # Loesungskomponenten als Funktionen
  loe4:=dsolve({sys4,init},{fcns}); # 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):
```

$$loe4 := \left\{ x(t) = -\frac{1}{5} e^{\left(\frac{\sqrt{5}-1}{2}t\right)} \sqrt{5} + \frac{1}{5} e^{\left(-\frac{\sqrt{5}+1}{2}t\right)} \sqrt{5} + 1 + t, \right.$$

$$\left. y(t) = -\frac{1}{5} \left(\frac{\sqrt{5}}{5} - \frac{1}{2} \right) e^{\left(\frac{\sqrt{5}-1}{2}t\right)} \sqrt{5} + \frac{1}{5} \left(-\frac{\sqrt{5}}{5} - \frac{1}{2} \right) e^{\left(-\frac{\sqrt{5}+1}{2}t\right)} \sqrt{5} + 1 \right\}$$

```
> dsolve({sys4,x(0)=1,y(0)=0},{fcns},method=laplace);
  l4:= [collect(simplify(rhs(op(1,%))),exp),
        collect(simplify(rhs(op(2,%))),exp)];
  evalf(l4[1]);
  evalf(l4[2]);
```

$$\left\{ x(t) = 1 + t - \frac{2}{5} e^{\left(-\frac{t}{2}\right)} \sqrt{5} \sinh\left(\frac{t\sqrt{5}}{2}\right), \right.$$

$$\left. y(t) = 1 + \frac{1}{5} \left(-5 \cosh\left(\frac{t\sqrt{5}}{2}\right) + \sqrt{5} \sinh\left(\frac{t\sqrt{5}}{2}\right) \right) e^{\left(-\frac{t}{2}\right)} \right\}$$

$$l4 := \left[1 + t - \frac{2}{5} e^{\left(-\frac{t}{2}\right)} \sqrt{5} \sinh\left(\frac{t\sqrt{5}}{2}\right), 1 + \left(-\cosh\left(\frac{t\sqrt{5}}{2}\right) + \frac{1}{5} \sqrt{5} \sinh\left(\frac{t\sqrt{5}}{2}\right) \right) e^{\left(-\frac{t}{2}\right)} \right]$$

$$1. + t - 0.8944271908 e^{(-0.5000000000 t)} \sinh(1.118033988 t)$$

$$1. + (-1. \cosh(1.118033988 t) + 0.4472135954 \sinh(1.118033988 t)) e^{(-0.5000000000 t)}$$

```
> 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 (4.32) erhält man bei AWP (4.35) 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.

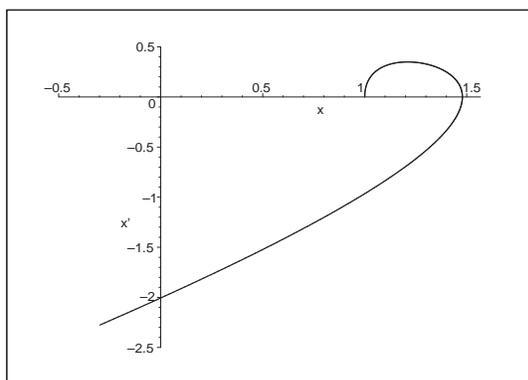


Abb. 4.59:
Datei *nsteif20.ps*,
AWP (4.35),
Phasenkurve $(x(t), x'(t))$, $t \in [0, 4]$,
ohne Richtungsfeld (x, x')

Das gleiche gilt natürlich, wenn man mehrere Phasenkurven mit DEplot darstellt.

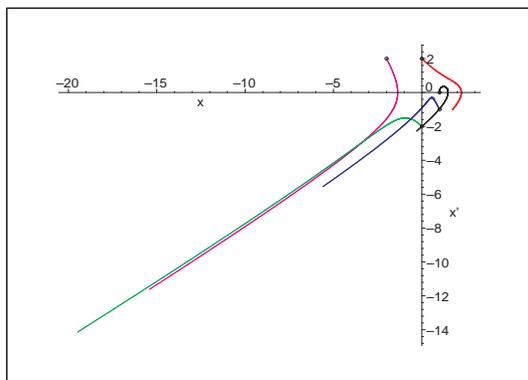


Abb. 4.60:
Datei *nsteif21.ps*,
AWP (4.35),
Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$,
mit AB $(1, 0)$, $(1, -1)$, $(0, 2)$, $(0, -2)$, $(-2, 2)$,
ohne Richtungsfeld (x, x')

Auch mit dfieldplot ist die Darstellung des Richtungsfeldes nicht möglich. Also nutzt man den üblichen Plot von Phasenkurven und Lösungstrajektorien.

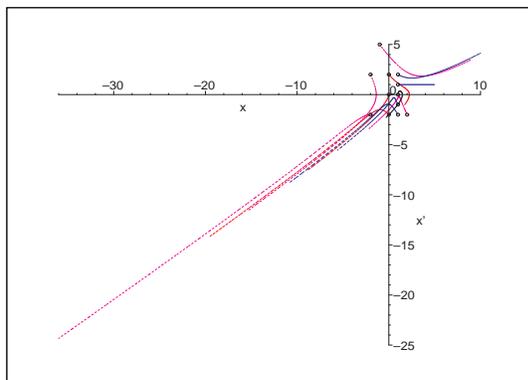


Abb. 4.61:
Datei *nsteif23.ps*,
AWP (4.35),
Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$,
mit den AB $(1, 0)$, $(1, 1)$, $(1, 2)$, $(1, -1)$, $(1, -2)$,
 $(0, 0)$, $(0, 2)$, $(0, -2)$,
 $(-2, 2)$, $(2, -2)$, $(-2, -2)$, $(-1, 5)$
im Bereich $[-36, 10] \times [-25, 5]$

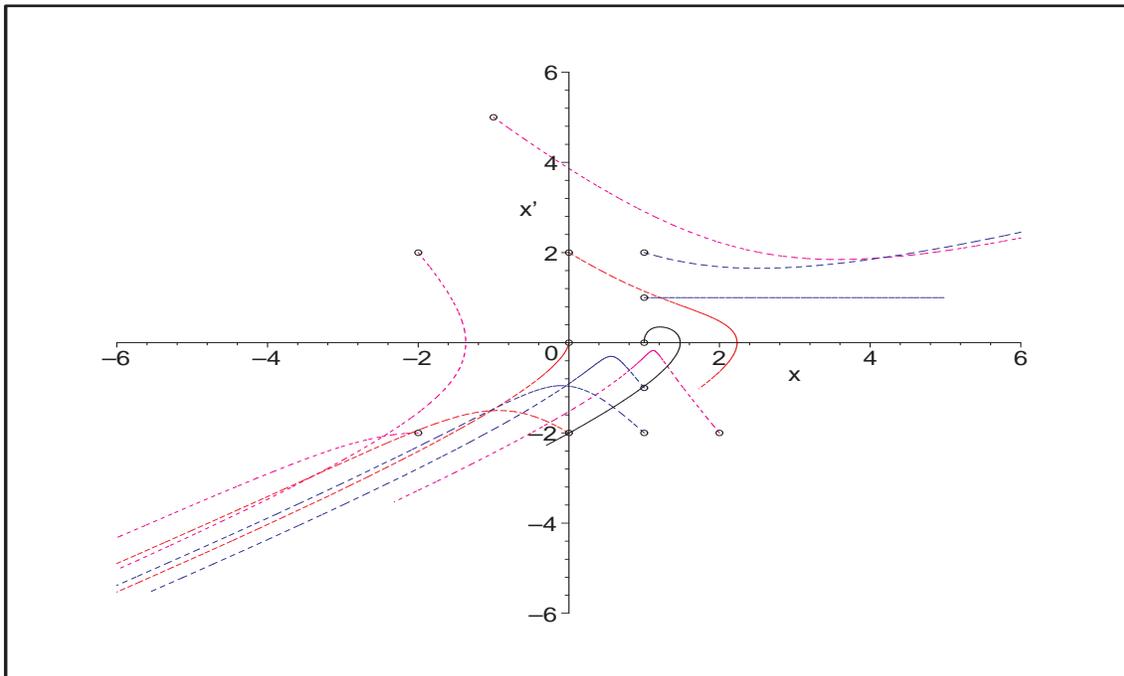


Abb. 4.62: Datei *nsteif24.ps*, AWP (4.35), Phasenkurven $(x(t), x'(t))$, $t \in [0, 4]$, zu den genannten AB im Bereichsausschnitt $[-6, 6] \times [-6, 6]$

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.

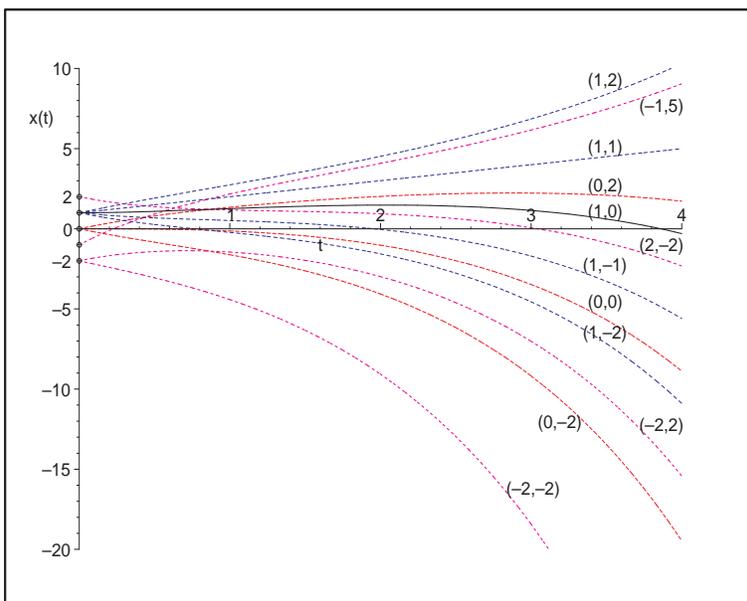


Abb. 4.63: Datei *nsteif22.ps*, AWP (4.35), Lösungstrajektorien $x(t)$, $t \in [0, 4]$, zu den genannten AB

(e) Gegeben sei die autonome inhomogenen nichtlineare gDGL 2. Ordnung als Randwertproblem (RWP) mit den Randbedingungen (RB)

$$x''(t) - x'(t)x(t) - 1 = 0, \quad x(0) = 0, \quad x(1) = 0, \quad t \in [0, 1]. \quad (4.36)$$

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

$$x''(t) - x'(t)x(t) - 1 = 0, \quad x(0) = 0, \quad x'(0) = s, \quad t \geq 0, \quad (4.37)$$

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

$$x_s(t) = -\frac{1}{2}t + \frac{1}{2}t^2 + \frac{1}{24}t^3 - \frac{1}{16}t^4 + \frac{1}{48}t^5, \quad (4.38)$$

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):

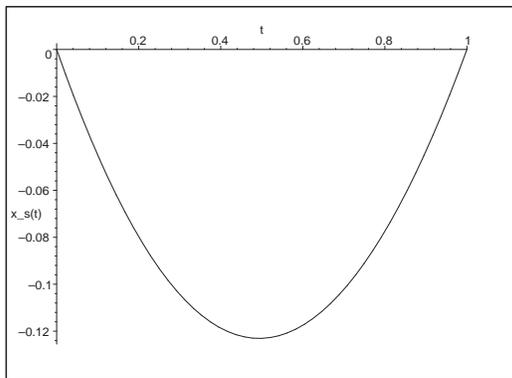
> xxt(0), xxt1(0);
  xxt(1), xxt1(1);
  simplify(eval(subs(x(t)=xxt(t),de5)));

```

$$0, \frac{-1}{2}$$

$$0, \frac{23}{48}$$

$$\frac{25}{96}t^4 + \frac{47}{192}t^5 - \frac{7}{128}t^6 - \frac{13}{576}t^7 + \frac{3}{256}t^8 - \frac{5}{2304}t^9 = 0$$

**Abb. 4.64:**Datei *nsteif25.ps*,AWP (4.37), AB mit $s = -0.5$,

Näherungslösung (4.38)

 $x_s(t)$, $t \in [0, 1]$,

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:

```

$$bsol := \text{proc}(x_bvp) \dots \text{end proc}$$

$$\left[t = 0., x(t) = 0., \frac{d}{dt}x(t) = -0.50416592791003595965 \right]$$

$$\left[t = 0.5, x(t) = -0.12498305218921320693, \frac{d}{dt}x(t) = 0.0036444537572298360201 \right]$$

$$\left[t = 1.0, x(t) = 0., \frac{d}{dt}x(t) = 0.49583407208996404024 \right]$$

Somit werden wir bei Lösung des AWP auch mit dem recht genauen Parameterwert $s = -0.504166$ für die Anfangsableitung $x'(0)$ operieren.

Wir notieren das AWP (4.37) ebenfalls als System von zwei autonomen inhomogenen nichtlinearen gDGL 1. Ordnung gemäß

$$\begin{aligned}x'(t) &= y(t), & x(0) &= 0, \\y'(t) &= x(t)y(t) + 1, & y(0) &= s, \\ \mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)) = \mathbf{f}(\mathbf{x}(t)), & \mathbf{x}(0) &= (0, s)^T.\end{aligned}\tag{4.39}$$

```
> 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 DGL-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
```

$$loe51 := \left[\begin{array}{c} \left[t, x(t), \frac{d}{dt}x(t) \right] \\ \left[\begin{array}{ccc} 0. & 0. & -.5 \\ .1 & -.0449643668968620764 & -.398989102854782064 \\ .2 & -.0797595174060793112 & -.296819209691574270 \\ .3 & -.104325523530942959 & -.194558092569997010 \\ .4 & -.118693460210456888 & -.0929559312516558128 \\ .5 & -.122953307477378551 & .0075587579093452728 \\ .6 & -.117222138246121432 & .106870514847029362 \\ .7 & -.101613507979648396 & .205162652501499842 \\ .8 & -.0762082805158981053 & .302903851009129699 \\ .9 & -.0410265004321861623 & .400841586868400102 \\ 1.0 & .00400065301760353913 & .500008002611822877 \end{array} \right] \end{array} \right]$$

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)]));
```

$$loe52 := \begin{bmatrix} & & \begin{bmatrix} t, x(t), \frac{d}{dt}x(t) \end{bmatrix} \\ \begin{bmatrix} 0. \\ .100000000000000004 \\ .200000000000000010 \\ .299999999999999988 \\ .400000000000000022 \\ .500000000000000000 \\ .599999999999999976 \\ .699999999999999954 \\ .800000000000000042 \\ .900000000000000022 \\ 1. \end{bmatrix} & \begin{bmatrix} 0. \\ -.0449646130717692994 \\ -.0797596151028408668 \\ -.104325521843663698 \\ -.118693411953866234 \\ -.122953350696438396 \\ -.117222067729801482 \\ -.101613630914293943 \\ -.0762082207520253935 \\ -.0410266681722313842 \\ .00400071376438856843 \end{bmatrix} & \begin{bmatrix} -.500000000000000000 \\ -.398988608633262132 \\ -.296819172820215360 \\ -.194558105717530100 \\ -.0929559442475329096 \\ .00755868737673635882 \\ .106870495297289452 \\ .205162623366426434 \\ .302903839647251372 \\ .400841797801013922 \\ .500008002309892952 \end{bmatrix} \end{bmatrix}$$

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);
```

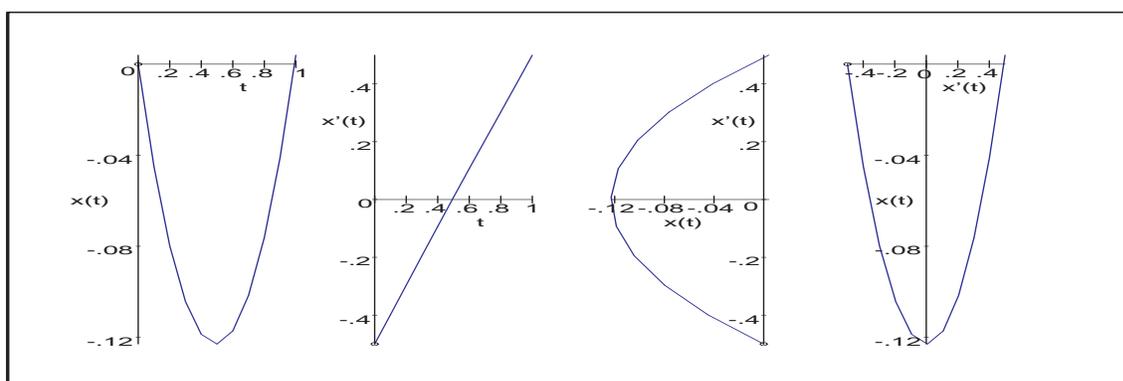


Abb. 4.65: Datei *nsteif26.ps*, AWP (4.37), AB mit $s = -0.5$, Näherungslösung $x(t)$, $t \in [0, 1]$, als Polygonzug sowie $(t, x'(t))$, $(x(t), x'(t))$, $(x'(t), x(t))$ bei 11 äquidistanten Auswertungsstellen $t_i = \frac{i}{10}$, $i = 0(1)10$, mittels `dsolve({de5,...,numeric,method=gear,...})`

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);
```

```
loes53 := proc(xrkf45) ... end proc
[ t = 0., x(t) = 0.,  $\frac{d}{dt}x(t) = -.500000000000000$  ]
[ t = .5, x(t) = -.122953350696438396,  $\frac{d}{dt}x(t) = .00755868737673635622$  ]
[ t = 1., x(t) = .00400073019471768043,  $\frac{d}{dt}x(t) = .500008005791624743$  ]
0., -.0449646130717692576, -.0797596151028408806, -.104325521843663712,
-.118693411953866178, -.122953350696438396, -.117222067729801454,
-.101613630914293915, -.0762082207520253380, -.0410266252286706490,
.00400073019471768043
```

(e2) Mit dem Maple-Kommando `phaseportrait` bestimmen wir zu (4.37) 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=classical[rk4]` 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 (4.36).

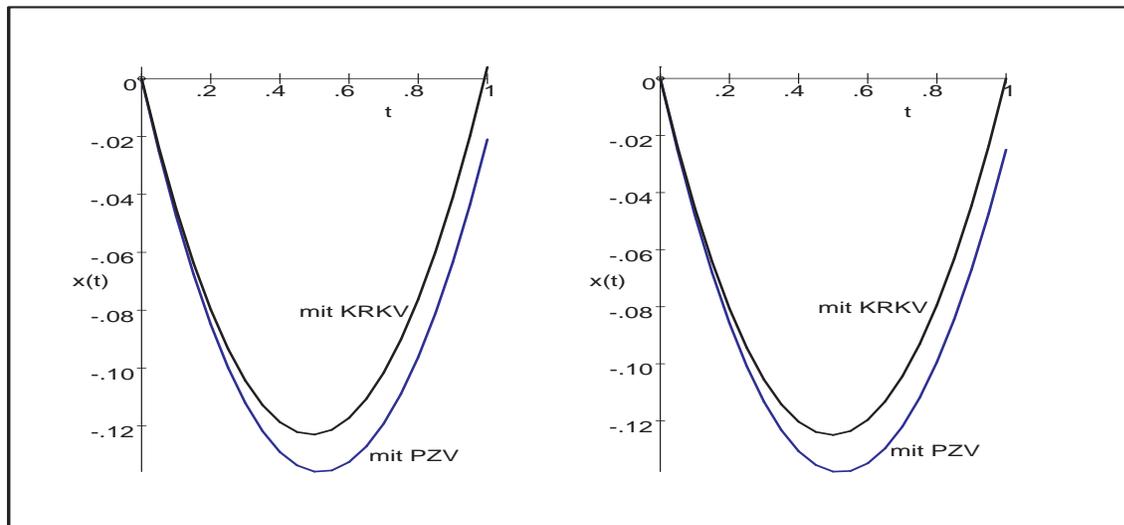


Abb. 4.66: Datei *nsteif27.ps*, AWP (4.37), AB mit $s = -0.5$ (linke Grafik) sowie $s = -0.504166$, Näherungslösungen $x(t)$, $t \in [0, 1]$, mittels `phaseportrait(de5, ..., method=classical[rk4], ...)` bzw. `phaseportrait(de5, ..., method=classical[foreuler], ...)`

(e3) Das Maple-Kommando `DEplot` hat analoge Parameter zu `phaseportrait`. Zur gDGL (4.37) 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=classical[rk4]`.

```
> 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);
```

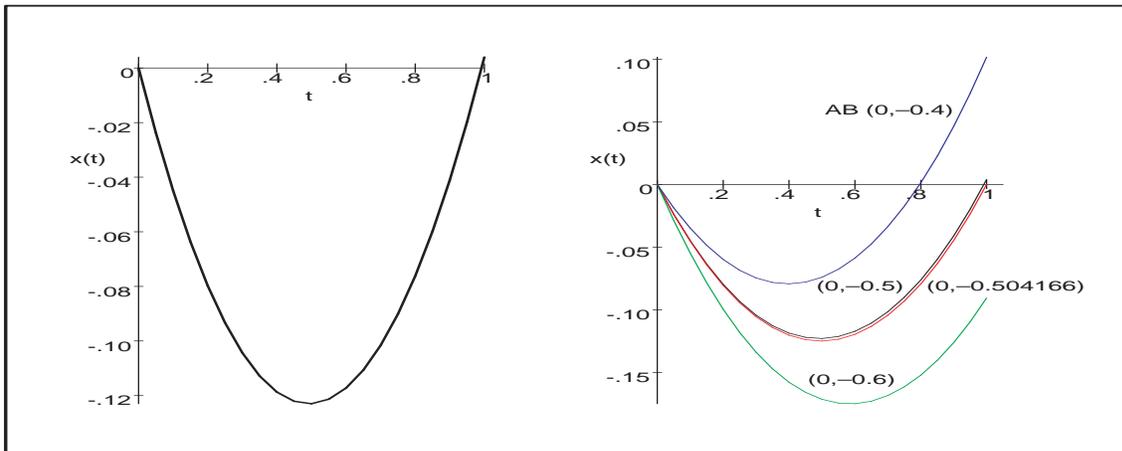


Abb. 4.67: Datei *nsteif28.ps*, AWP (4.37), AB mit $s = -0.5$ (linke Grafik) sowie $s = -0.6, -0.504166, -0.5, -0.4$, Näherungslösungen $x(t), t \in [0, 1]$, mittels `DEplot({de5},...)` bei `method=classical[rk4]`

(e4) Für das autonome SysgDGL (4.39) können wir mit den Kommandos `DEplot`, `dfieldplot`, `phaseportrait` das Richtungsfeld zeigen, ev. 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);
```

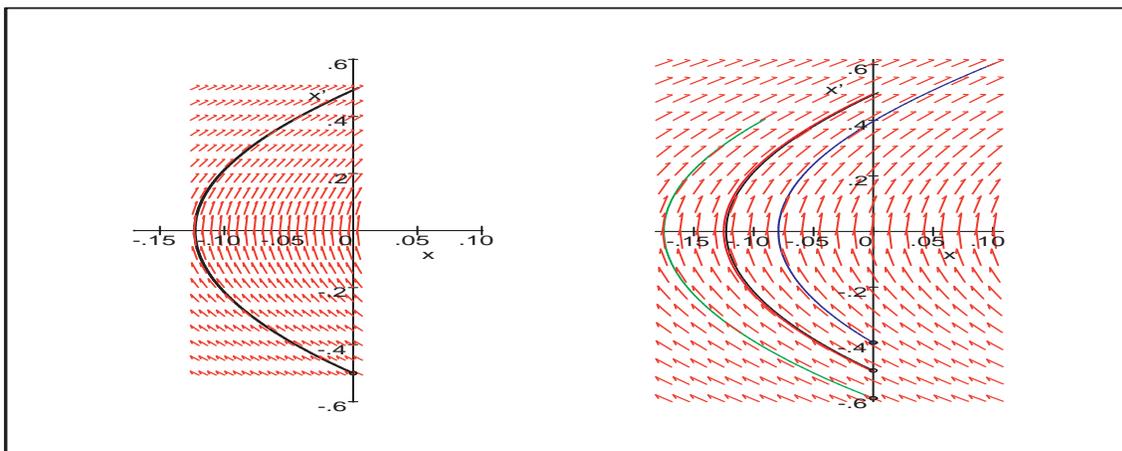


Abb. 4.68: Datei *nsteif31.ps*, AWP (4.39), AB mit $s = -0.5$ (linke Grafik) sowie $s = -0.6, -0.504166, -0.5, -0.4$, Näherungslösungen $x(t), t \in [0, 1]$, und Richtungsfeld mittels `DEplot({sys5},...)`

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']):
```

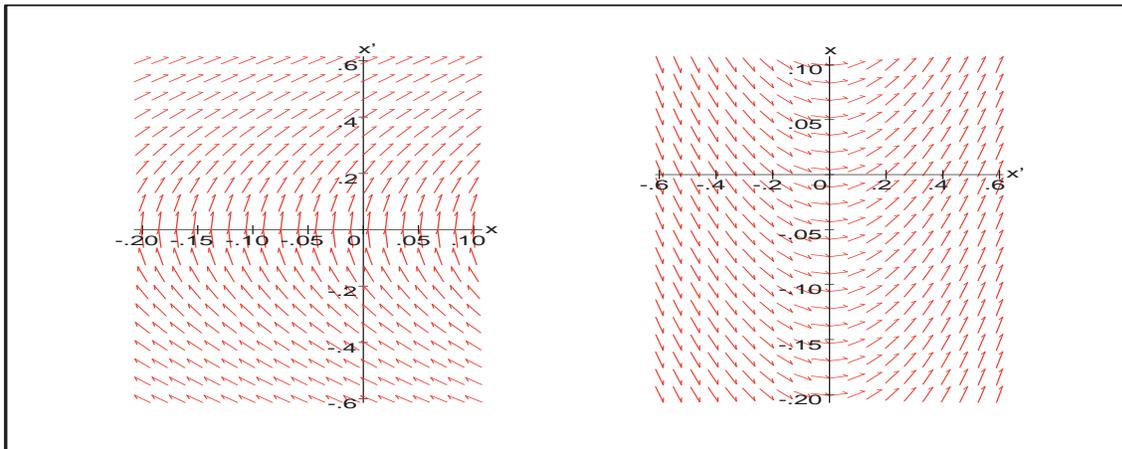


Abb. 4.69: Datei `nsteif32.ps`, AWP (4.39), AB mit $s = -0.5$,
Richtungsfeld mit `dfieldplot([sys5],...)` als (x, x') und (x', x)

Verwenden wir den Befehl `phaseportrait` für das autonome SysgDGl (4.39), so wird der Plot in Abhängigkeit vom gewünschten Szenario `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);
```

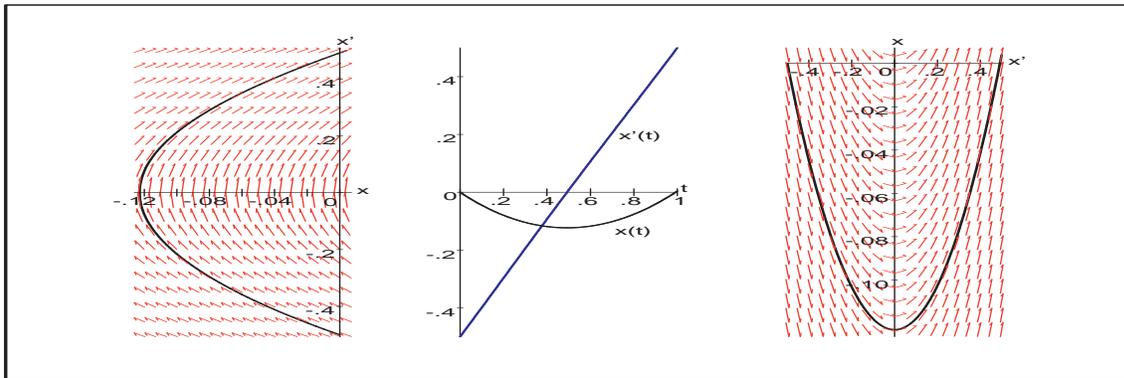


Abb. 4.70: Datei *nsteif33.ps*, AWP (4.39), AB mit $s = -0.5$,
`phaseportrait([sys5], ...)` als (x, x') , $\{(t, x), (t, x')\}$ und (x', x)

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],[fncs],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);
```

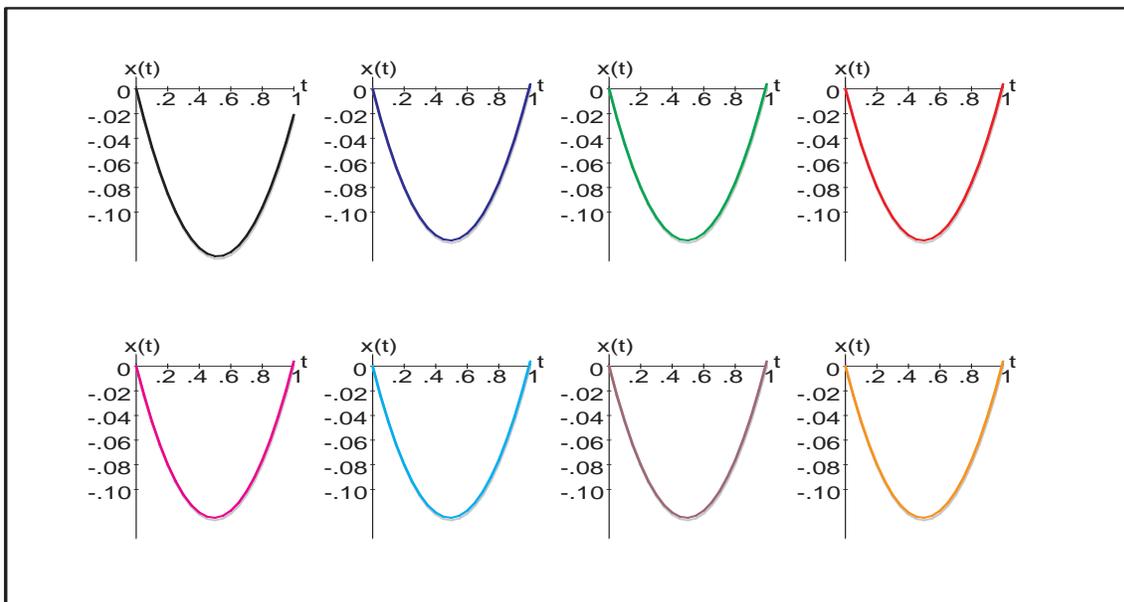


Abb. 4.71: Datei *nsteif34.ps*, AWP (4.39), AB mit $s = -0.5, -0.504166$,
`phaseportrait([sys5], ...)` für $(t, x(t))$ mit 8 klassischen Methoden ◀

Wenden wir uns der Steifheit von DGI zu.

Sie wird durch das Verhalten von Lösungen und Lösungskomponenten bzw. Lösungen zueinander charakterisiert. Diese finden wir bei gDGI 1. Ordnung als $x(t)$, bei SysgDGI als $\{x(t), y(t), \dots\}$ und gDGI der Ordnung größer Eins als $\{x(t), x'(t), \dots\}$.

gDGI 1. Ordnung nehmen dabei eine Sonderstellung ein. Zunächst ist es schon ein Unterschied, ob der Lösungsverlauf sich mehr wie $e^{\pm 1000t}$ 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 aufgrund der Kompliziertheit der gDGI 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 sinnvolle Näherungslösung erzeugen. Sie benötigen eine Schrittweitensteuerung 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 höchstens die Genauigkeitsordnung 2, was praktisch oft nicht ausreicht. Die Steifheit der gDGI zeigt sich in drastischen Veränderungen der Lösung und ihrer Ableitung, auftretender Nichtlinearitäten sowie der starken Abhängigkeit der partiellen Ableitungen der rechten Seite der gDGI von den Variablen. Eine mögliche Ansicht des Richtungsfeldes der gDGI kann nützliche Hinweise zum Lösungsverhalten geben.

Darüber hinaus haben wir in Beispiel [4.4] (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 2 zu arbeiten. Bei höherdimensionalen SysgDGI können all die genannten Probleme gleichzeitig auftreten.

[4.13] Beispiele

(a) Gegeben sei die nicht autonome nichtlineare gDGI 1. Ordnung als AWP

$$x'(t) = f(t, x(t)) = x^2(t) - t^6, \quad x(0) = -0.1, \quad t \geq 0. \quad (4.40)$$

Als erstes verweisen die partiellen Ableitungen der rechten Seite $f(t, x)$ auf mögliche Probleme und Steifheit, denn in Bemerkung [1.8] (b) haben wir schon das explodierende Verhalten der Lösung der "verwandten" gDGI $x' = x^2$ festgestellt.

Maple kann das AWP unter Verwendung von Bessel-Funktionen 1. und 2. Art lösen.

```
> dg11:=diff(x(t),t)=x(t)^2-t^6;           # Digits=16
  init:=x(0)=-0.1;
  loe1:=dsolve({dg11,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

$$loe1 := x(t) = \frac{t^3 \left(\frac{1}{160} \frac{\pi \left(10 \cdot 2^{\left(\frac{3}{4}\right)} 4^{\left(\frac{3}{4}\right)} \Gamma \left(\frac{7}{8} \right)^2 + \pi \csc \left(\frac{\pi}{8} \right) \right) 2^{1/4} 4^{1/4} \operatorname{BesselI} \left(\frac{-7}{8}, \frac{t^4}{4} \right)}{\Gamma \left(\frac{7}{8} \right)^2 \sin \left(\frac{\pi}{8} \right)} - \operatorname{BesselK} \left(\frac{7}{8}, \frac{t^4}{4} \right) \right)}{\frac{1}{160} \frac{\pi \left(10 \cdot 2^{\left(\frac{3}{4}\right)} 4^{\left(\frac{3}{4}\right)} \Gamma \left(\frac{7}{8} \right)^2 + \pi \csc \left(\frac{\pi}{8} \right) \right) 2^{1/4} 4^{1/4} \operatorname{BesselI} \left(\frac{1}{8}, \frac{t^4}{4} \right)}{\Gamma \left(\frac{7}{8} \right)^2 \sin \left(\frac{\pi}{8} \right)} + \operatorname{BesselK} \left(\frac{1}{8}, \frac{t^4}{4} \right)}$$

$$x(t) = \frac{1. t^3 (4.701307519771402 \operatorname{BesselI}(-0.8750\dots, 0.2500\dots t^4) - 1. \operatorname{BesselK}(0.8750\dots, 0.2500\dots t^4))}{4.701307519771402 \operatorname{BesselI}(0.1250\dots, 0.2500\dots t^4) + \operatorname{BesselK}(0.1250\dots, 0.2500\dots t^4)}$$

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 sind 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.

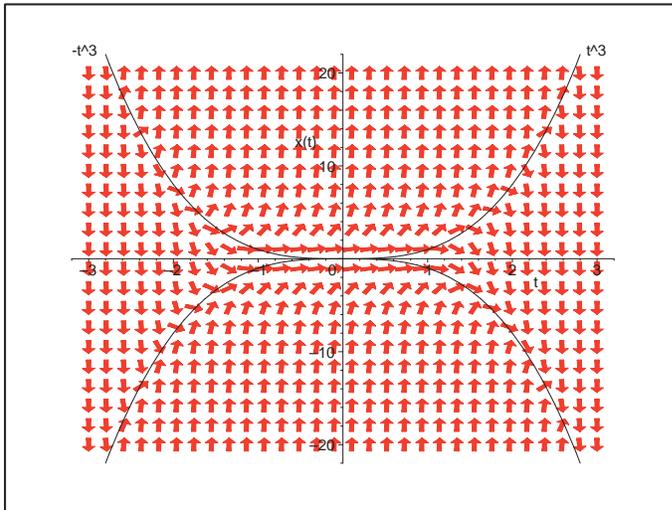


Abb. 4.72:
 Datei *nsteif35.ps*,
 Richtungsfeld zur gDGL
 des AWP (4.40),
 Trennkurven $x = \pm t^3$,
 Kommando `dfieldplot`

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 und somit Asymptote und die Funktion ist auch rechts davon definiert. Die Scheitelpunkte der unteren Lösungszweige liegen auf der Trennkurve $-t^3$. Mit kleinen AB werden die 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. 4.5:
 Polstellen zu Lösungen des AWP (4.40)
 bei verschiedenen AB $x(0) = s$

Es macht natürlich keinen Sinn, die numerische Integration des AWP über die Unstetigkeitsstellen hinweg durchzuführen.

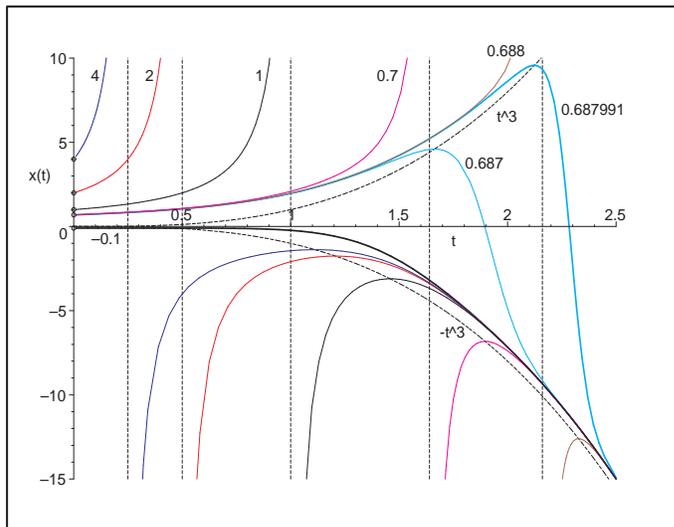


Abb. 4.73:

Datei *nsteif39.ps*,
 Lösungstrajektorien
 des AWP (4.40) mit AB
 $x(0) = -0.1, 4, 2, 1, 0.7,$
 $0.688, 0.687991, 0.687$
 und Asymptoten,
 dazu Trennkurven $x = \pm t^3$,

```
> p1:=DEplot(dg11,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);
```

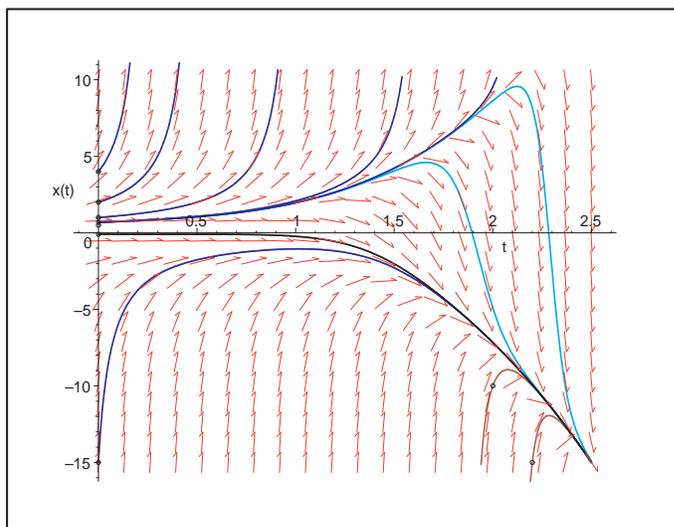


Abb. 4.74:

Datei *nsteif36.ps*,
 Richtungsfeld zur gDGL
 des AWP (4.40),
 Lösungstrajektorien 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=10-7`. Die Ergebnisse sind in Tabellenform, so dass 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
l1s1:=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,l1s1,l1s2,l1s3,l1s4],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)');
```

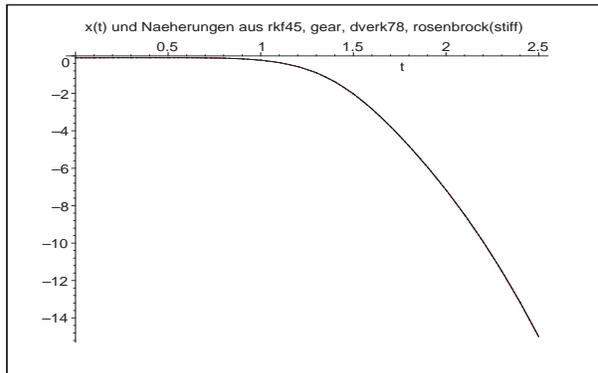


Abb. 4.75:

Datei *nsteif37.ps*,
Lösung $x(t)$, $t \in [0, 2.5]$, zum AWP (4.40)
sowie Näherungslösungen mit `rkf45`,
`gear`, `dverk78`, `rosenbrock(stiff)`,
Kommando `dsolve`

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]-l1s1[i][2],llex[i][2]-l1s2[i][2],
          llex[i][2]-l1s3[i][2],llex[i][2]-l1s4[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	-0.000000000000024	0.000000000000004	-0.00000005791164
0.4	-0.00000031864491	-0.000000000000178	0.000000000000309	-0.00000005724140
0.5	0.00000053629647	-0.00000000001295	0.0000000044261	-0.00000002839484
0.6	0.00000031546491	-0.00000000001265	0.0000000003058	-0.00000004882850
0.7	-0.00000033802165	-0.00000000001234	0.00000000059221	-0.00000005114452
0.8	-0.00000041867334	-0.00000000001208	0.0000000042353	0.00000000676051
0.9	-0.00000041703858	-0.00000000000589	0.0000000287195	0.00000000335879
1.0	-0.00000036916536	-0.00000000000595	0.0000000287422	-0.00000000593579
1.1	-0.00000030448658	0.00000000035281	0.0000000495726	-0.00000008728480
1.2	-0.00000032452329	0.00000000032602	0.0000000382640	0.00000000878593
1.3	-0.00000059348770	0.0000000047937	0.0000000437093	0.00000007083672
1.4	-0.00000080533875	0.0000000043230	0.00000001016220	0.00000004490201
1.5	-0.00000082734155	0.0000000061307	0.00000000959554	-0.00000014967193
1.6	-0.00000103610247	0.0000000068615	0.00000000648726	0.00000030306288
1.7	-0.00000140284812	0.0000000061330	0.0000000363566	-0.00000082054182
1.8	-0.00000154597091	0.0000000041456	0.0000000188713	-0.00000132398911
1.9	-0.00000232672866	0.00000000052781	0.0000000146561	-0.00000003359037
2.0	-0.00000209312064	0.00000000161115	0.0000000066557	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	-0.0000000001657	0.0000000008328	0.00000307665675

In den Abschnitten 2.1 und 2.2 haben wir die RKV unter Verwendung einer vorgegebenen festen Schrittweite in MATLAB implementiert. Sie sollen hier zur Demonstration des Einflusses der Steifheit 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] = rk1v(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_1	-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_1	-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_1	-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_1	-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_1	-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_1	-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	

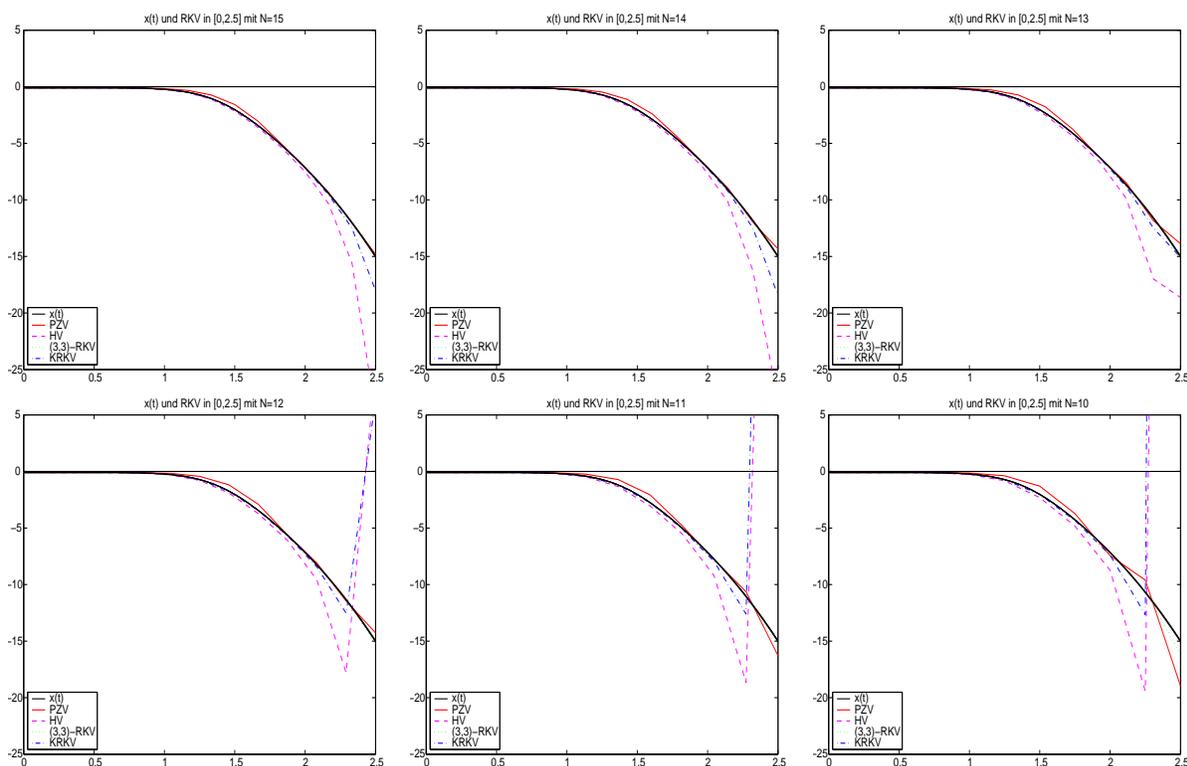


Abb. 4.76: Datei *stief_01..06.ps*, AWP (4.40), Lösung $x(t)$, $t \in [0, 2.5]$, und Näherungslösungen η_i mittels PZV, HV, (3,3)-RKV, KRKV bei einer Teilintervallanzahl $N = 15(-1)10$

Das (3,3)-RKV zeigt sich im Vergleich mit den anderen ESV am wenigsten empfindlich auf die Steifheit.

Zur Kontrolle wollen wir zum AWP (4.40) 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	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)-eta1_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 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

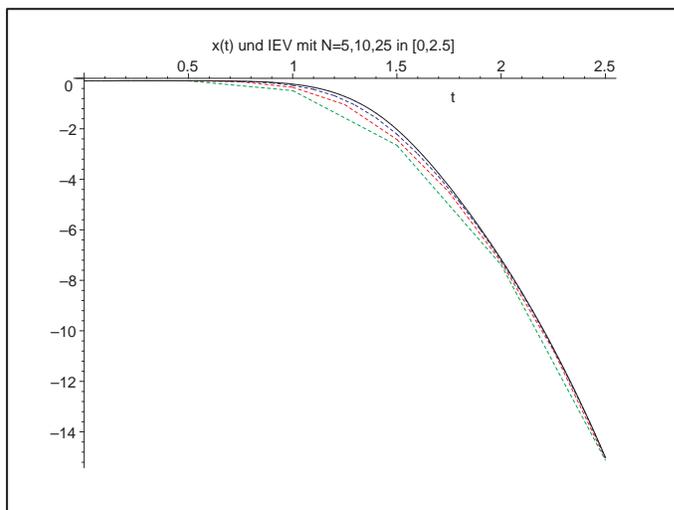


Abb. 4.77:
 Datei *nsteif38.ps*,
 AWP (4.40),
 Lösung $x(t)$, $t \in [0, 2.5]$, und
 Näherungslösungen η_i
 mittels IEV bei einer
 Teilintervallanzahl $N = 5, 10, 25$

(b) Gegeben sei als AWP die homogene autonome lineare gDGL 2. Ordnung

$$x'' + \frac{1}{100}x' + x = 0, \quad x(0) = 0, \quad x'(0) = 1, \quad t \in [0, T]. \quad (4.41)$$

Mit $x' = y$ erhält man die Darstellung als homogenes lineares SysgDGL 1. Ordnung

$$\begin{aligned} x'(t) &= y(t), & x(0) &= 0, \\ y'(t) &= -x(t) - \frac{1}{100}y(t), & y(0) &= 1, \\ \mathbf{x}'(t) &= A\mathbf{x}(t), & A &= \begin{pmatrix} 0 & 1 \\ -1 & -\frac{1}{100} \end{pmatrix}, \quad \mathbf{x}(0) = (0, 1)^T. \end{aligned} \quad (4.42)$$

Es handelt sich dabei um ein SysgDGL, wo die Steifheit durch betragskleine Realteile und betragsgroße Imaginärteile der Eigenwerte der Systemmatrix A und damit stark oszillierendes Verhalten der Lösungskomponenten verursacht wird.

Die Schrittweitenbeschränkung (4.12) ist bei $\lambda(A) = s + i\omega$, $s < 0$, $|\omega| \gg 0$,

$$h < \frac{2|\Re\lambda|}{|\lambda|^2} = -\frac{2s}{s^2 + \omega^2}$$

und kann damit $h \ll 1$ nach sich ziehen.

Wir bestimmen die exakte Lösung des AWP (4.41).

```
> dgl2:=diff(x(t),t,t)+1/100*diff(x(t),t)+x(t): # Digits=16
init:=x(0)=0,D(x)(0)=1:
loe2:=dsolve({dgl2,init},x(t));
evalf(loe2);
xx:=unapply(rhs(loe2),t): # Loesungsfunktion
```

$$loe2 := x(t) = \frac{200}{39999} \sqrt{39999} e^{\left(-\frac{t}{200}\right)} \sin\left(\frac{\sqrt{39999}t}{200}\right)$$

$$x(t) = 1.000012500234380 e^{(-0.0050000000000000000 t)} \sin(0.9999874999218740 t)$$

Mit der Funktion $x(t)$ und ihren ersten Ableitungen entstehen die folgenden Grafen.

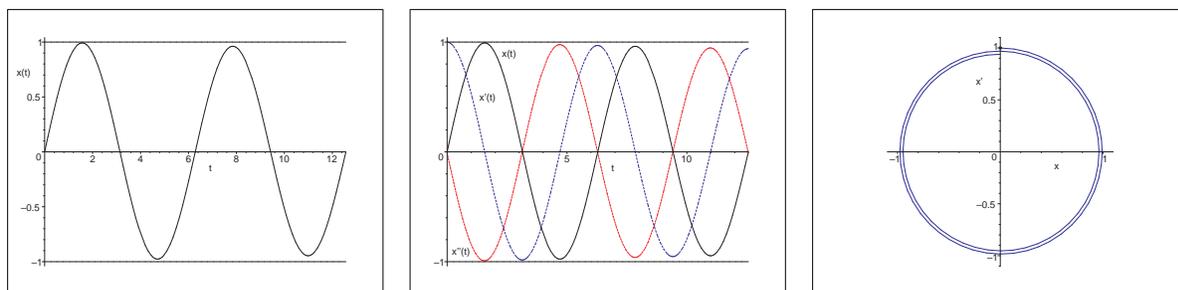


Abb. 4.78: Dateien *nsteif49..51.ps*, AWP (4.41), Lösung $x(t)$, $t \in [0, 4\pi]$,

Grafik mit $x(t)$, $x'(t)$, $x''(t)$ sowie Phasenkurve $(x(t), x'(t))$, $t \in [0, 4\pi]$

Zur Untersuchung der Steifheit und Schrittweitenbedingung nimmt man die Matrix A des SysgDGI (4.42). Ihre beiden konjugiert komplexen Eigenwerte sind

$$\begin{aligned}\lambda_{1,2} &= \frac{-1 \pm i \sqrt{39999}}{200} \\ &= -0.005 \pm i 0.9999874999218740.\end{aligned}$$

Beide haben den gleichen Betrag 1 und die Schrittweitenbedingung zum PZV ist

$$h < \frac{1}{100}.$$

Analog ist die Lösung des SysgDGI (4.42) 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)):
```

$$loe2 := \left\{ \begin{array}{l} x(t) = e^{\left(-\frac{t}{200}\right)} \left(_C1 \sin\left(\frac{\sqrt{39999}t}{200}\right) + _C2 \cos\left(\frac{\sqrt{39999}t}{200}\right) \right), \end{array} \right.$$

$$y(t) = \frac{1}{200} e^{\left(-\frac{t}{200}\right)} \left(\begin{array}{l} -_C1 \sin\left(\frac{\sqrt{39999}t}{200}\right) + _C1 \cos\left(\frac{\sqrt{39999}t}{200}\right) \sqrt{39999} \\ -_C2 \cos\left(\frac{\sqrt{39999}t}{200}\right) - _C2 \sin\left(\frac{\sqrt{39999}t}{200}\right) \sqrt{39999} \end{array} \right) \}$$

$$\{ x(t) = e^{(-0.005000000000t)} (_C1 \sin(0.9999875000t) + _C2 \cos(0.9999875000t)),$$

$$y(t) = 0.005000000000 e^{(-0.005000000000t)} (-1. _C1 \sin(0.9999875000t) + 199.9975000 _C1 \cos(0.999987500t) - 1. _C2 \cos(0.9999875000t) - 199.9975000 _C2 \sin(0.9999875000t)) \}$$

$$loe2a := \left\{ \begin{array}{l} x(t) = \frac{200}{39999} \sqrt{39999} e^{\left(-\frac{t}{200}\right)} \sin\left(\frac{\sqrt{39999}t}{200}\right), \end{array} \right.$$

$$y(t) = \frac{1}{200} e^{\left(-\frac{t}{200}\right)} \left(\begin{array}{l} -\frac{200}{39999} \sin\left(\frac{\sqrt{39999}t}{200}\right) \sqrt{39999} + 200 \cos\left(\frac{\sqrt{39999}t}{200}\right) \end{array} \right) \}$$

$$\left. \begin{aligned} x(t) &= 1.000012500 e^{(-0.005000000000 t)} \sin(0.9999875000 t), \\ y(t) &= 0.005000000000 e^{(-0.005000000000 t)} (-1.000012500 \sin(0.9999875000 t) + \\ &\quad 200. \cos(0.9999875000 t)) \end{aligned} \right\}$$

Zum AWP (4.42) 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 SysgDGL, wenn als Standardmethode `method=classical[rk4]` 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);
```

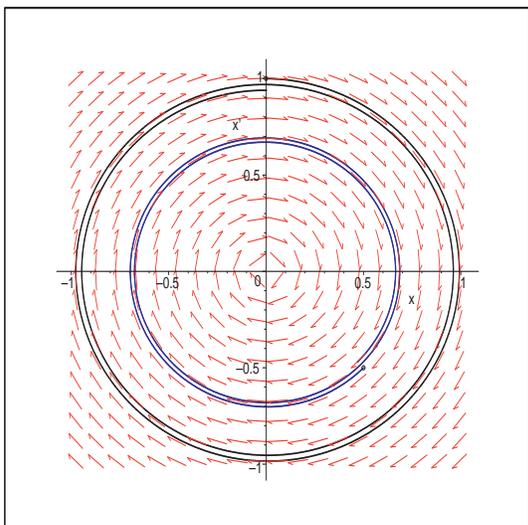


Abb. 4.79:

Datei `nsteif41.ps`,

Richtungsfeld zum AWP (4.42)

mit zwei numerischen Lösungen

zu den AB $(x(0), y(0)) = (0, 1)$,

$(x(0), y(0)) = (0.5, -0.5)$,

Kommando DEplot

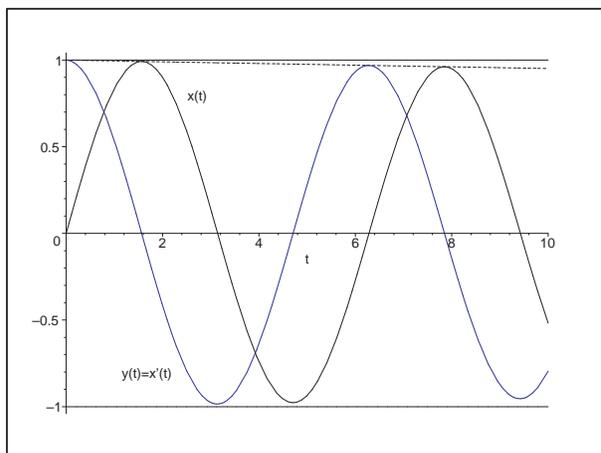


Abb. 4.80:
 Datei *nsteif42.ps*,
 AWP (4.42) mit Lösungen
 $x(t), y(t), t \in [0, 10]$,
 dazu die begrenzenden Geraden ± 1 und
 zum Vergleich die Dämpfungsfunktion
 $1.0000125 e^{-0.005t}$

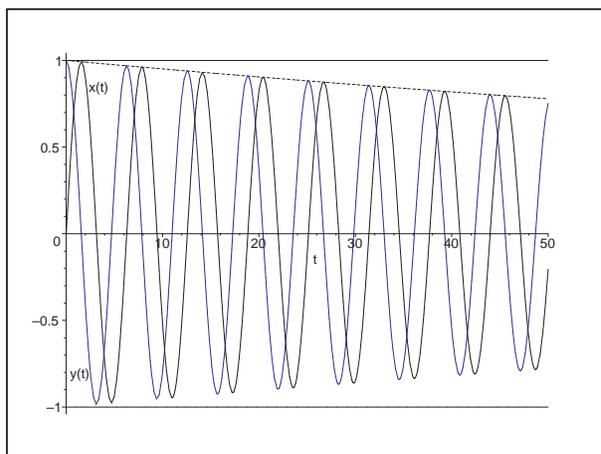


Abb. 4.81:
 Datei *nsteif43.ps*,
 AWP (4.42) mit Lösungen
 $x(t), y(t), t \in [0, 50]$

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 $h \geq 0$.

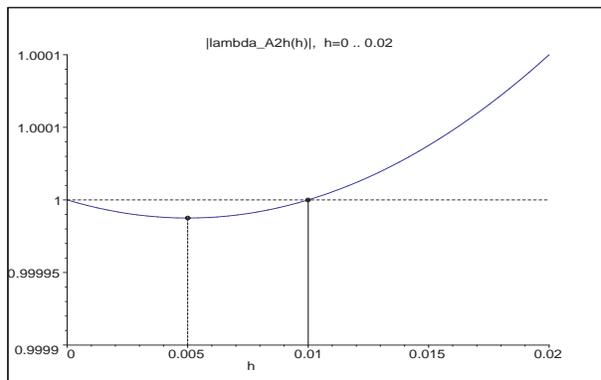


Abb. 4.82:
 Datei *nsteif44.ps*,
 Betrag der EW der Übergangsmatrix
 des PZV zum AWP (4.42)
 als Funktion von $h \geq 0$

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 $\frac{\sqrt{39999}}{200} = 0.999\,987\,500$ 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.

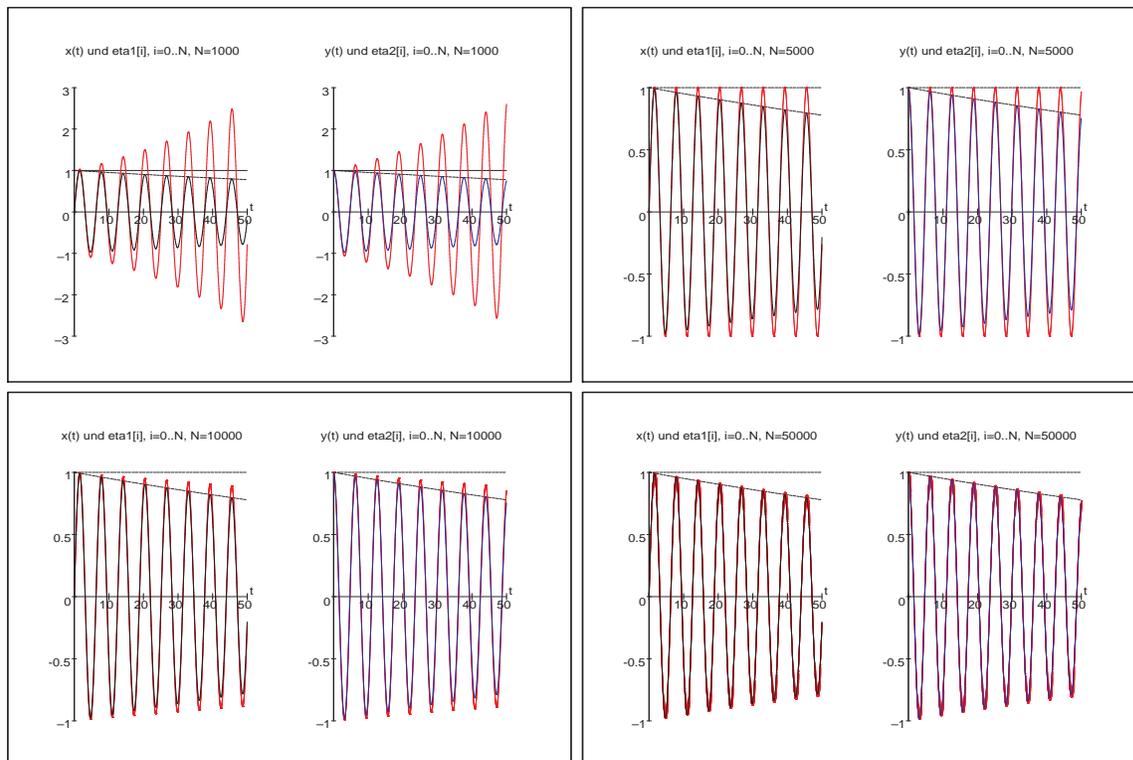


Abb. 4.83: Dateien *nsteif45..48.ps*, AWP (4.42) bei $t \in [0, 50]$,
 Näherungslösungen η_{1i}, η_{2i} mit PZV bei $h = 0.05, 0.01, 0.005, 0.001$
 im Vergleich mit den exakten Lösungskomponenten $x(t), y(t)$

4.3.1 Systeme mit ausgeprägter Steifheit

Es handelt sich dabei um ein SysgDGl, wo die Steifheit durch sehr unterschiedliche Beträge der negativen Realteile der Eigenwerte der Systemmatrix A und damit mehr oder weniger stark abnehmendes Verhalten der Lösungskomponenten verursacht wird. Ausschlaggebend ist der Eigenwert mit dem betragsgrößten negativen Realteil.

Die Schrittweitenbeschränkung (4.12) 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.

[4.14] Beispiele

(a) Gegeben sei als AWP die homogene autonome lineare gDGl 2. Ordnung

$$x'' + 100x' + x = 0, \quad x(0) = 0, \quad x'(0) = 1, \quad t \in [0, T]. \quad (4.43)$$

Mit $x' = y$ erhält man die Darstellung als homogenes lineares SysgDGl 1. Ordnung

$$\begin{aligned} x'(t) &= y(t), & x(0) &= 0, \\ y'(t) &= -x(t) - 100y(t), & y(0) &= 1, \\ \mathbf{x}'(t) &= A\mathbf{x}(t), & A &= \begin{pmatrix} 0 & 1 \\ -1 & -100 \end{pmatrix}, & \mathbf{x}(0) &= (0, 1)^T. \end{aligned} \quad (4.44)$$

Wir bestimmen die exakte Lösung des AWP (4.43).

```
> 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): # Lösungsfunktion
```

$$loe3 := \frac{1}{714} \sqrt{51} e^{((-50 + 7\sqrt{51})t)} - \frac{1}{714} \sqrt{51} e^{-(50 + 7\sqrt{51})t}$$

$$x(t) = 0.01000200060020007 (e^{(-0.01000100020005t)} - e^{(-99.98999899979995t)})$$

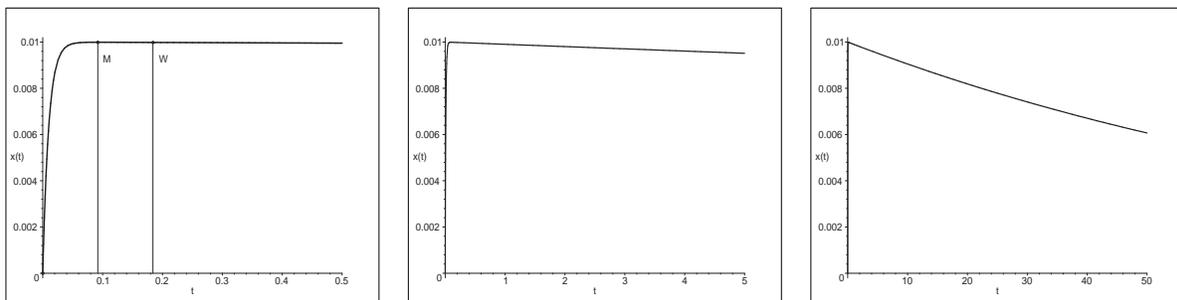


Abb. 4.84: Dateien *nsteif57..59.ps*, AWP (4.43), Lösung $x(t)$, $t \in [0, T]$, $T = 0.5, 5, 50$

Die Lösungskurve $x(t)$ hat einen vergleichsweise steilen Anfangsanstieg. An der Stelle $t_M = 0.092\,119$ ist ihr Maximum M mit $x(t_M) = 0.009\,991$ sowie bei $t_W = 0.184\,239$ ein Wendepunkt W mit $x(t_W) = 0.009\,983$ und $x'(t_W) = -0.000\,0998$.

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.

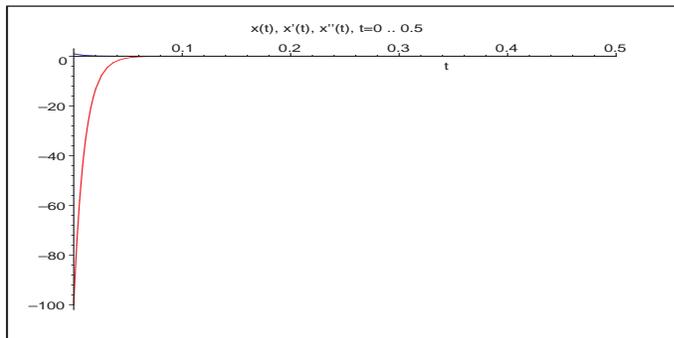


Abb. 4.85:
 Datei *nsteif60.ps*,
 AWP (4.43),
 Lösung $x(t)$, $t \in [0, 0.5]$ mit
 den ersten beiden Ableitungen

Ähnlich gelingt es, die problematischen Teile der Phasenkurve (x, x') durch die Wahl eines günstigen Zeitbereichs zu erfassen. Wir wollen uns daher systematisch an die Problemzone der "unteren rechten Ecke" herantasten.

```
> 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..0.5], opt1,view=[0..0.01,-0.1..1]);
plot([xx(t),xx1(t),t=0..0.50], opt1,view=[0..0.01,-0.1..1]);
plot([xx(t),xx1(t),t=0.06..0.20],opt1,view=[0.008..0.010,-0.0005..0.0020]);
```

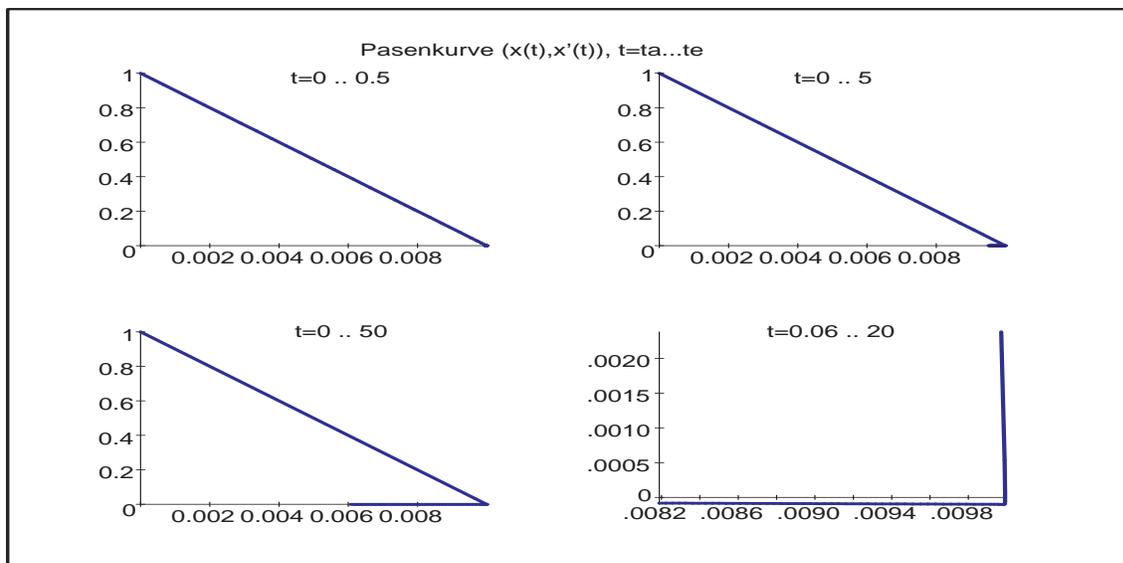


Abb. 4.86: Datei *nsteif61.ps*, Phasenkurve $(x(t), x'(t))$ in Zeitintervallen $[t_a, t_e]$

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 (4.44). Ihre beiden negativen reellen Eigenwerte sind

$$\lambda_{1,2} = -50 \pm 7\sqrt{51} = -0.010\,001\,000\,200\,05, \quad -99.989\,998\,999\,799\,95.$$

Ihre Beträge sind sehr verschieden und die Schrittweitenbedingung zum PZV ergibt sich aus dem betragsgrößten Wert und ist

$$h < \frac{2}{|\lambda_2|} = 0.020\,002\,000\,400\,100\,02, \quad \text{also grob } h \leq 0.02.$$

Analog ist die Lösung des SysgDGI (4.44) 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)):
```

$$loe3 := \{x(t) = {}_C1 e^{((-50 + 7\sqrt{51})t)} + {}_C2 e^{-(50 + 7\sqrt{51})t},$$

$$y(t) = {}_C1 (-50 + 7\sqrt{51}) e^{((-50 + 7\sqrt{51})t)} + {}_C2 (-50 - 7\sqrt{51}) e^{-(50 + 7\sqrt{51})t} \}$$

$$\{x(t) = {}_C1 e^{(-0.01000100020005t)} + {}_C2 e^{(-99.98999899979995t)},$$

$$y(t) = -0.01000100020005 {}_C1 e^{(-0.01000100020005t)} -$$

$$99.98999899979995 {}_C2 e^{(-99.98999899979995t)} \}$$

$$loe3a := \{x(t) = \frac{1}{714}\sqrt{51} e^{((-50 + 7\sqrt{51})t)} - \frac{1}{714}\sqrt{51} e^{-(50 + 7\sqrt{51})t},$$

$$y(t) = \frac{1}{714}\sqrt{51} (-50 + 7\sqrt{51}) e^{((-50 + 7\sqrt{51})t)} - \frac{1}{714}\sqrt{51} (-50 - 7\sqrt{51}) e^{-(50 + 7\sqrt{51})t} \}$$

$$\{x(t) = 0.01000200060020007 (e^{(-0.01000100020005t)} - e^{(-99.98999899979995t)}),$$

$$y(t) = -0.0001000300100035011 e^{(-0.01000100020005t)} +$$

$$1.000100030010004 e^{(-99.98999899979995t)} \}$$

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]],
         ssize=0.01, dirgrid=[20,20], linecolor=black, labels=['x', 'x']);
```

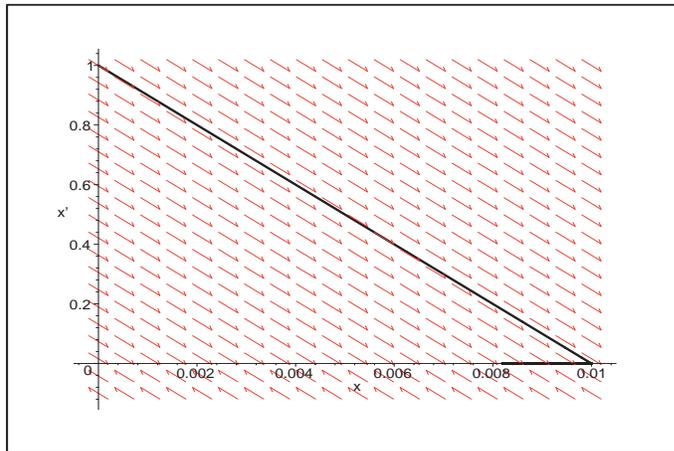


Abb. 4.87:

Datei *nsteif62.ps*,
AWP (4.44),
Richtungsfeld (x, x') und
Phasenkurve $(x(t), x'(t))$, $t \in [0, 20]$,
Kommando DEplot

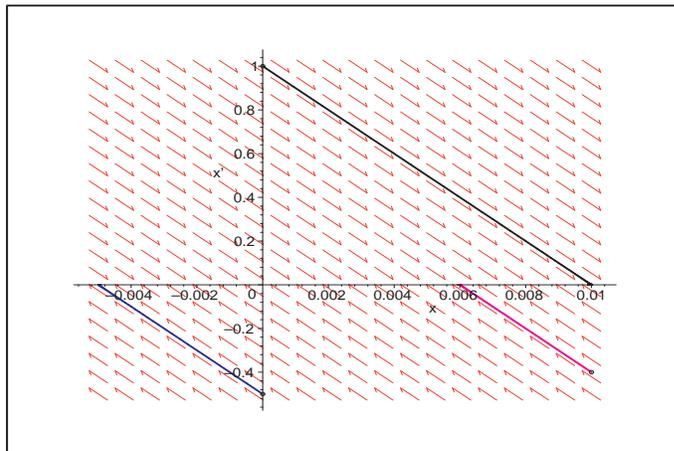


Abb. 4.88:

Datei *nsteif63.ps*,
AWP (4.44) 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 untermauern und zu beheben, indem wir uns immer mehr in die Problemzone hineinzoomen. 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);
```

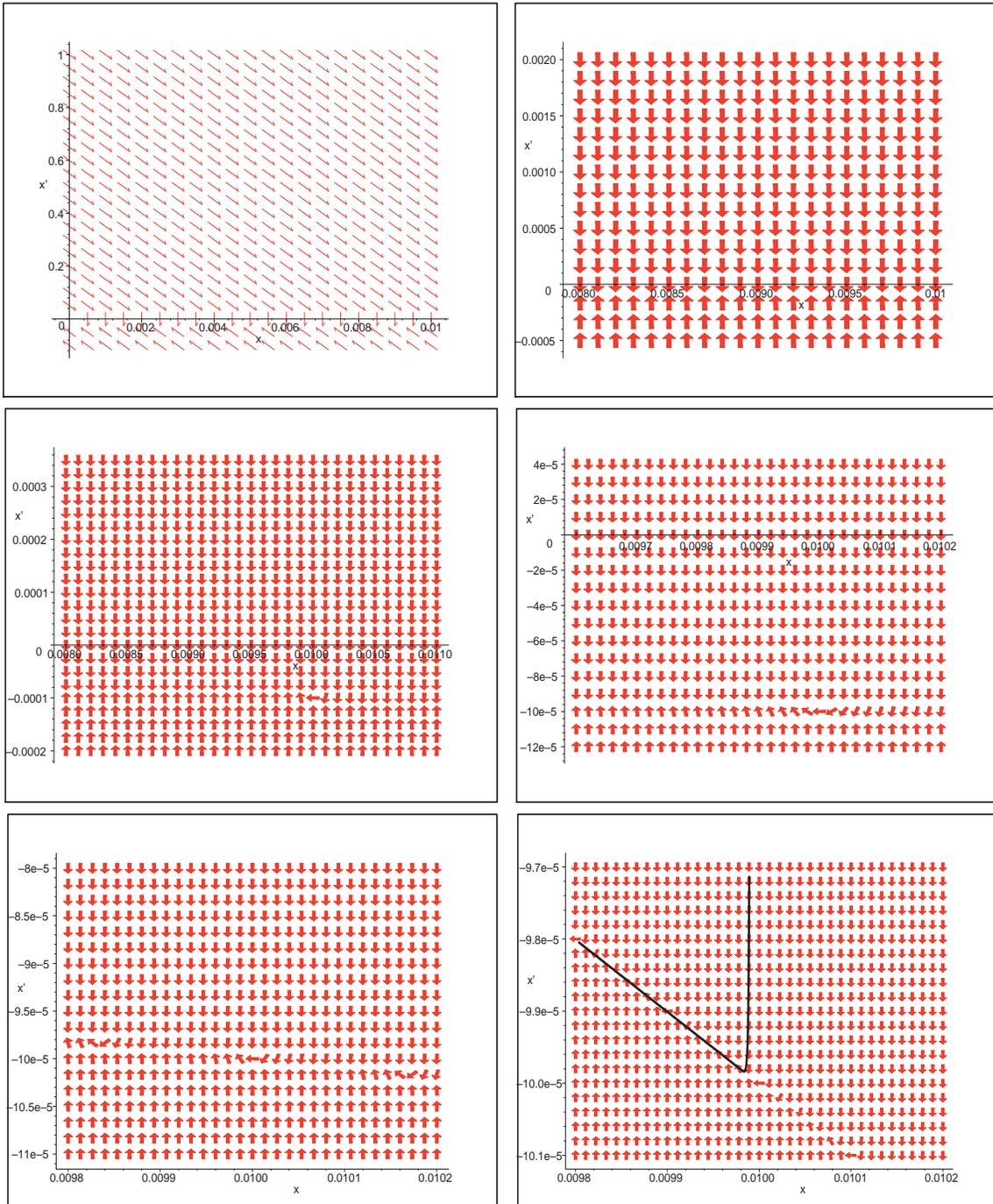


Abb. 4.89: Dateien *nsteif53.ps*, *nsteif53a.e.ps*, AWP (4.44), Richtungsfelder in ausgewählten Bereichen mit `dfieldplot`

Nimmt man das Verhalten der Lösungskomponenten $x(t)$ und $y(t)$ vom AWP (4.44) 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 $[0, t_M] = [0, 0.092119]$ 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.

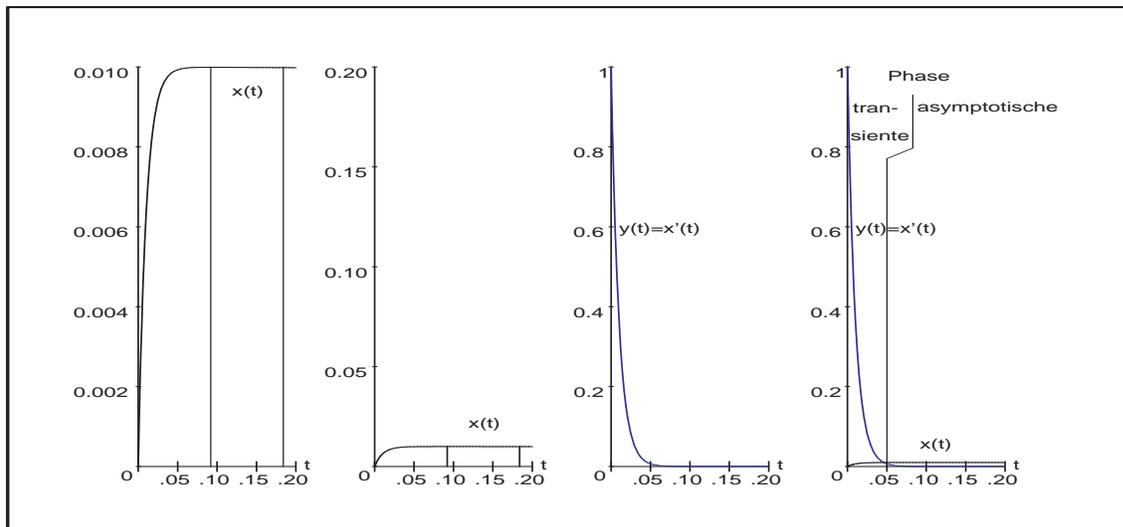


Abb. 4.90: Datei *nsteif54.ps*, AWP (4.44), Lösungen $x(t)$, $y(t)$ in $[0, 0.2]$ mit der Charakterisierung ihrer transienten und asymptotischen Phase

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.

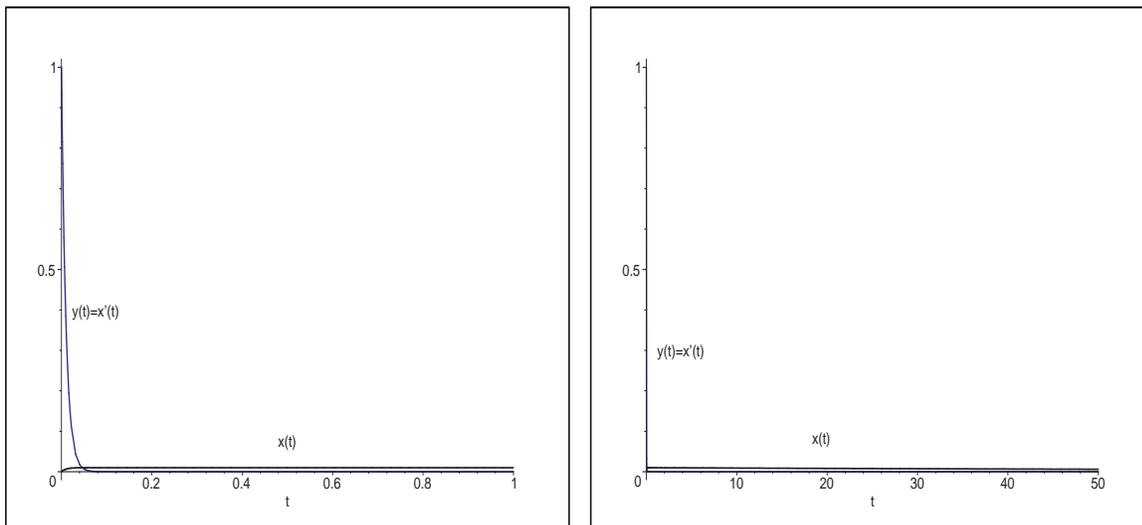


Abb. 4.91: Dateien *nsteif55..56.ps*, AWP (4.44), Lösungsverhalten von $x(t)$, $y(t)$ in $[0, 1]$ bzw. $[0, 50]$

Zum Abschluss 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 $h \approx 0.02$ 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 4.1). 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 1.2 und 2.2.

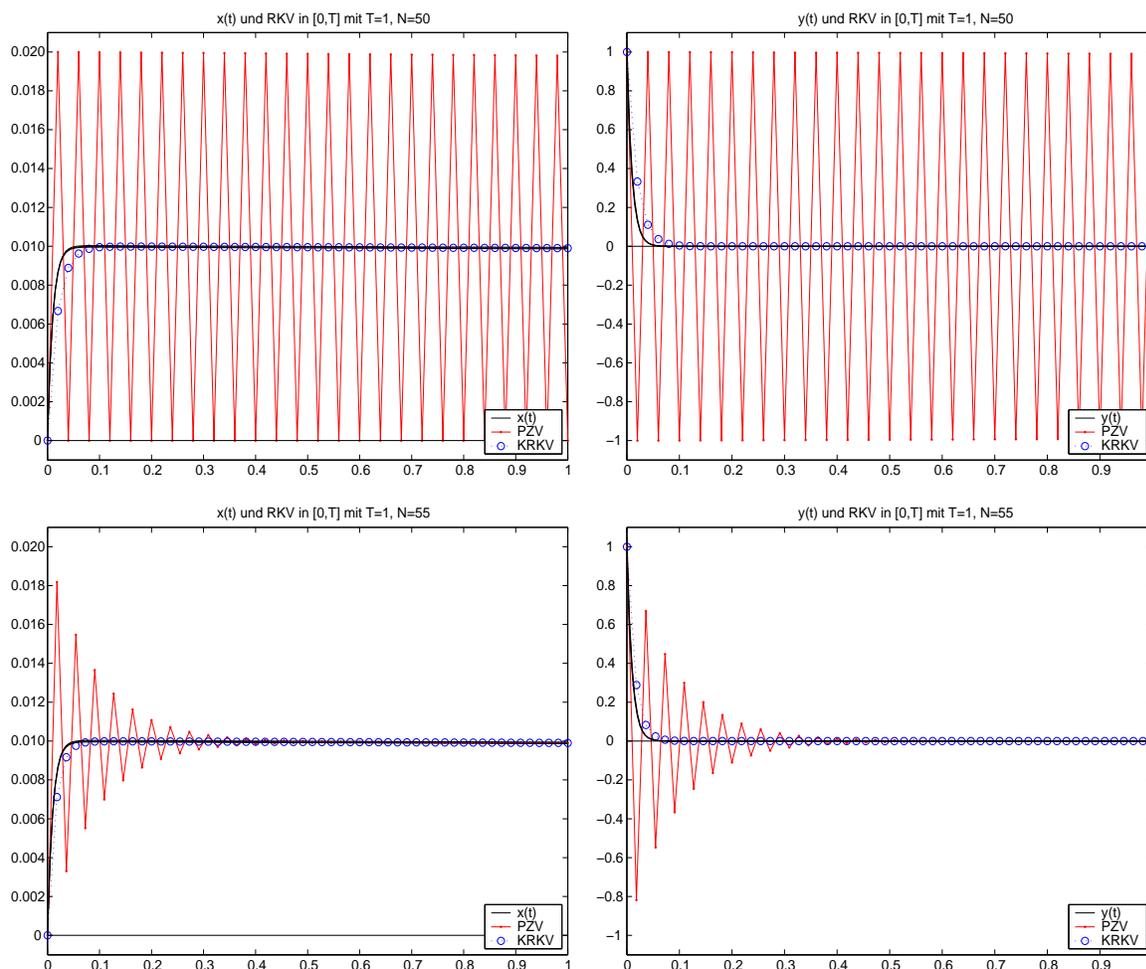


Abb. 4.92: Dateien *steif_10..13.ps*, AWP (4.44),
 Näherungslösungen η_i , $i = 0, 1, \dots, N$, aus PZV und KRKV in $[0, 1]$
 bei $N = 50, 55$ im Vergleich mit Lösungen $x(t)$, $y(t)$

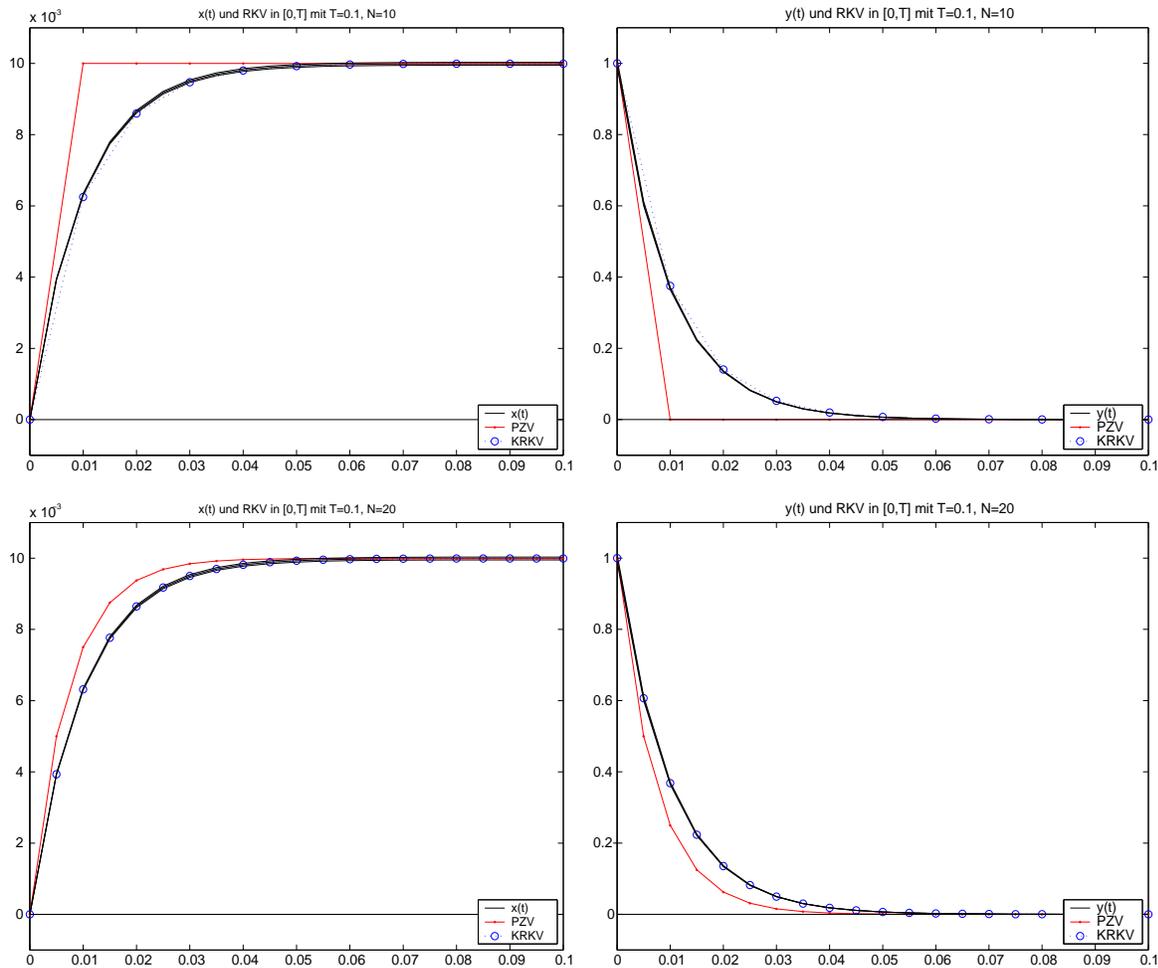


Abb. 4.93: Dateien *stef_14..17.ps*, AWP (4.44),
 Näherungslösungen η_i , $i = 0, 1, \dots, N$, aus PZV und KRKV in $[0, 0.1]$
 bei $N = 10, 20$ im Vergleich mit Lösungen $x(t)$, $y(t)$

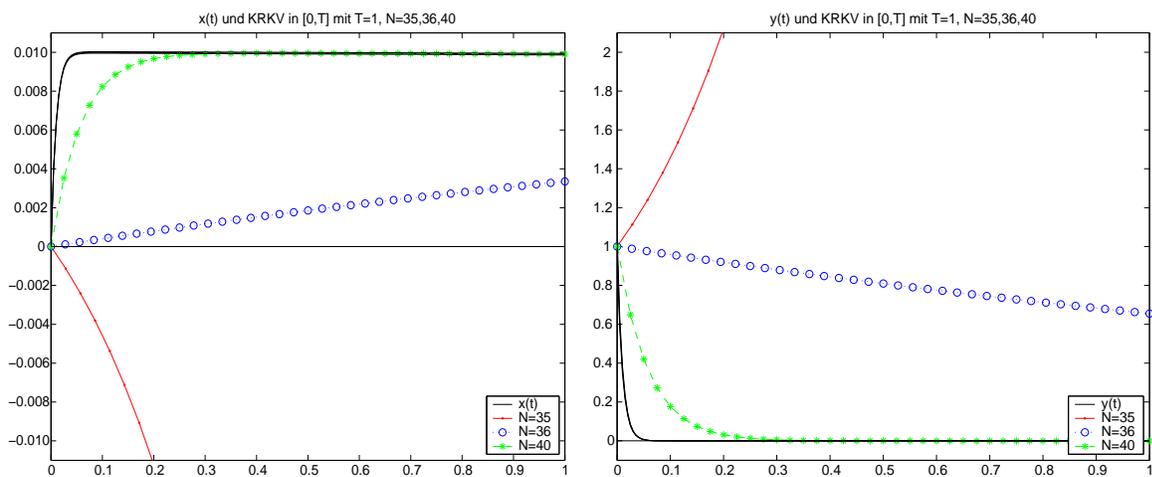


Abb. 4.94: Dateien *stef_18..19.ps*, AWP (4.44),
 Näherungslösungen η_i , $i = 0, 1, \dots, N$, aus KRKV in $[0, 1]$
 bei $N = 35, 36, 40$ im Vergleich mit Lösungen $x(t)$, $y(t)$

(b) Wir betrachten nun ein lineares SysgDGI mit mehr Steifheit als das vorherige.

Das AWP für $\mathbf{x} = (x, y)^T$ ist

$$\mathbf{x}'(t) = A \mathbf{x}(t), \quad \mathbf{x}(0) = (1, 1)^T, \quad t \geq 0, \quad \text{wobei} \quad A = \begin{pmatrix} -29\,998 & -39\,996 \\ 29\,997/2 & 19\,997 \end{pmatrix}. \quad (4.45)$$

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 $h < 2/|\lambda|_{max} = 0.000\,2$.

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 $t = 0.000\,5$ haben wir schon den Wert $0.006\,737$. 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 näherungsweise Lösung eines AWP die Schrittweitenbedingung durch "Ausblenden des restriktiven Anteils" entschärfen, weil man nicht die "zuständigen" Stellen im SysgDGI 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.

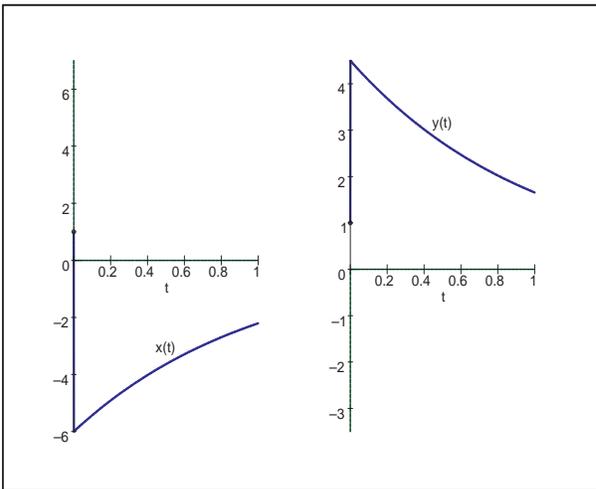


Abb. 4.95:

Datei *nsteif64.ps*,

AWP (4.45) mit AB $(x(0), y(0)) = (1, 1)$,
 Lösungen $x(t), y(t)$ in $[0, 1]$,

dazu die Lösungsanteile

$$-6e^{-t} \text{ (von } x(t) \text{ "zugedeckt")}, \quad 7e^{-10000t},$$

$$\frac{9}{2}e^{-t} \text{ (von } y(t) \text{ "zugedeckt")}, \quad -\frac{7}{2}e^{-10000t}$$

Einige Lösungsanteile zu $x(t), y(t)$ lassen sich nur bei besserer grafischer Auflösung erkennen.

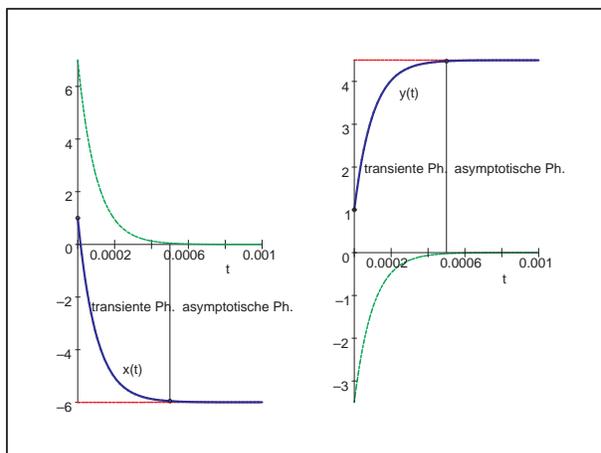


Abb. 4.96:

Datei *nsteif65.ps*,
 AWP (4.45) mit AB $(x(0), y(0)) = (1, 1)$,
 Lösungen $x(t), y(t)$ in $[0, 0.001]$,
 dazu die Lösungsanteile
 $-6e^{-t}, 7e^{-10000t}$ bzw. $\frac{9}{2}e^{-t}, -\frac{7}{2}e^{-10000t}$

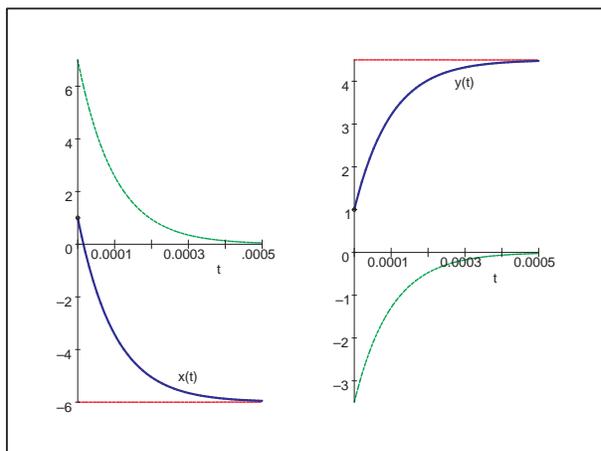


Abb. 4.97:

Datei *nsteif66.ps*,
 AWP (4.45) mit AB $(x(0), y(0)) = (1, 1)$,
 Lösungen $x(t), y(t)$ in $[0, 0.0005]$,
 dazu die Lösungsanteile
 $-6e^{-t}, 7e^{-10000t}$ bzw. $\frac{9}{2}e^{-t}, -\frac{7}{2}e^{-10000t}$

Das Zeichnen des Richtungsfeldes mit dem Kommando `DEplot` sowie der Näherungskurve zum AWP (4.45) 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.

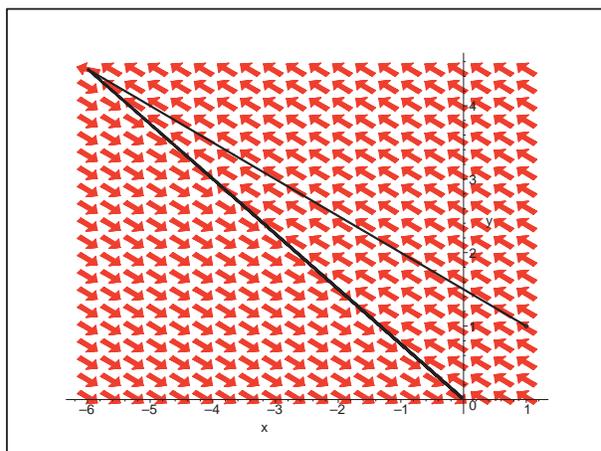


Abb. 4.98:

Datei *nsteif67.ps*,
 AWP (4.45),
 Richtungsfeld (x, y) und
 Phasenkurve $(x(t), y(t)), t \in [0, 5]$
 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. (4.87)), wollen uns jedoch nicht weiter in diesen Bereich hineinzoomen.

Zu ausgewählten Näherungsverfahren machen wir Rechnungen.

Einerseits nehmen wir Verfahren wie BDF(n) mit fester Schrittweite und BDF(5,4) mit variabler Schrittweitensteuerung (vergl. Tab. 3.6), zum anderen testen wir einige MATLAB-Prozeduren 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 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;

[aerr_x,ix] = max(abs(x(:,1))-xex(1:N1)) % Fehler und Stellen
[aerr_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. 4.6: AWP (4.45) in $[0,1]$, Schrittweitenverhalten von BDF-Methoden mit variabler Schrittweiten- und Ordnungssteuerung

Methode	Schritt-Zahl $N1$	Schrittweite h_i bei				$aerr_x(i)$	$aerr_y(i)$
		$i = 1$	$i \approx N1/2$	$t \approx 0.0005(i)$	$i = N1(i)$		
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. 4.7: AWP (4.45) in $[0,1]$, MATLAB-Prozeduren 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. Im Allgemeinen ist es sehr schwierig anzugeben, 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 (3.35)

$$\alpha_r \eta_{i+1} + \alpha_{r-1} \eta_i + \dots + \alpha_0 \eta_{i-r+1} = h \beta_r f_{i+1}, \quad i=r-1, r, \dots, \quad \eta_0, \eta_1, \dots, \eta_{r-1} \text{ geg.}$$

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.

		BDF(1)			
N	h	$aerr_x(i)$	$aerr_y(i)$	$aerr_{xN}$	$aerr_{yN}$
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. 4.8: AWP (4.45) in $[0,1]$, Fehler zu BDF(1) $\eta_{i+1} = M_1 \eta_i$ bei fester Schrittweite h

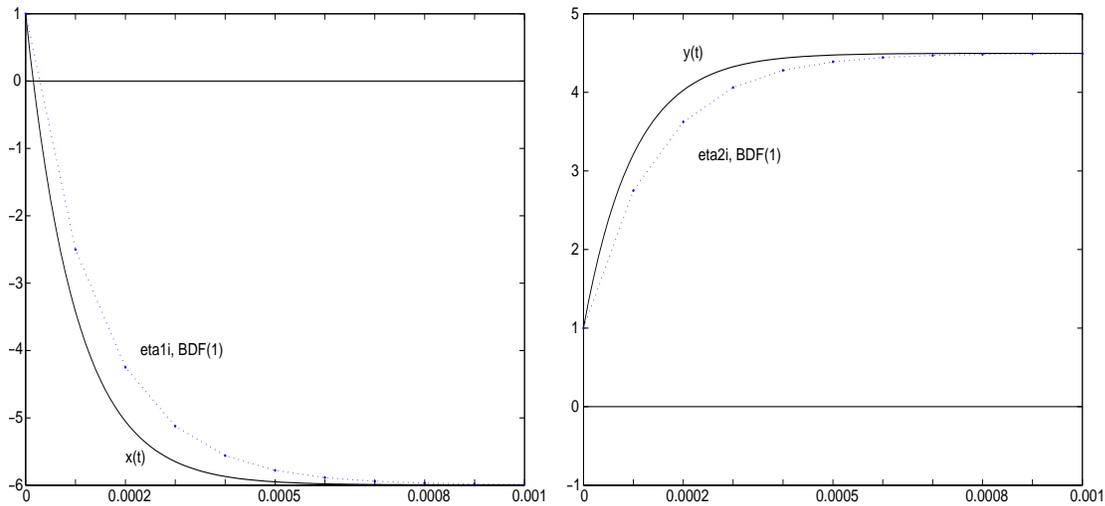


Abb. 4.99: Dateien *bdf1_3, 4.ps*, AWP (4.45) in $[0, 1]$, BDF(1) mit $h = 0.0001$, Vergleich mit exakten Lösungen $x(t)$ bzw. $y(t)$ im Bereich $t \in [0, 0.001]$

		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. 4.9: AWP (4.45) in $[0, 1]$, Fehler zu BDF(2) $\eta_{i+1} = M_2(4\eta_i - \eta_{i-1})$ bei festem h

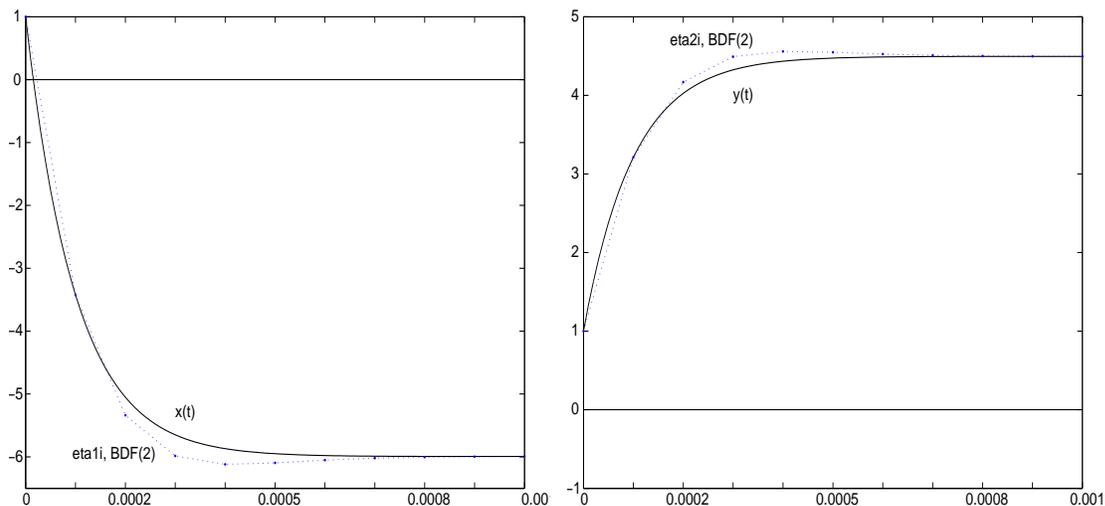


Abb. 4.100: Dateien *bdf2_3, 4.ps*, AWP (4.45) in $[0, 1]$, BDF(2) mit $h = 0.0001$, Vergleich mit exakten Lösungen $x(t)$ bzw. $y(t)$ im Bereich $t \in [0, 0.001]$

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. 4.10: AWP (4.45) 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

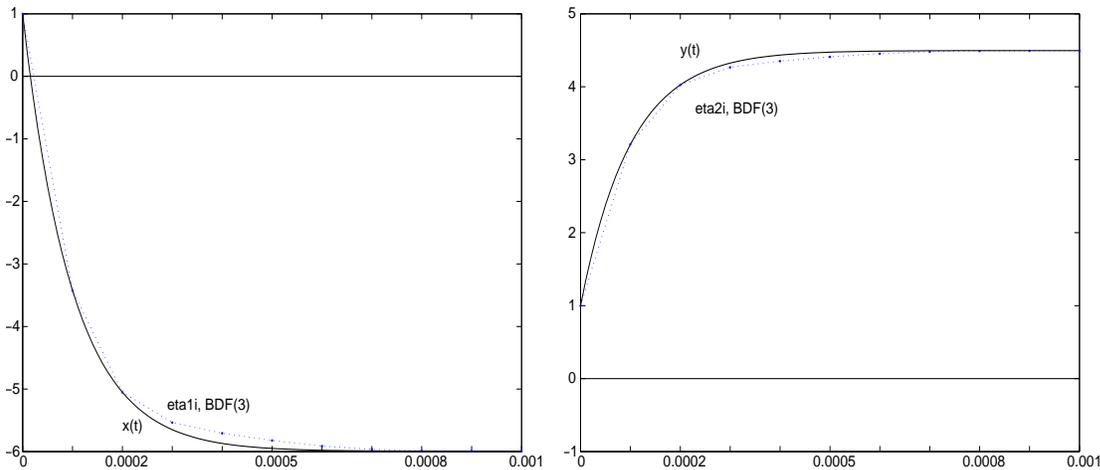


Abb. 4.101: Dateien *bdf3_3,4.ps*, AWP (4.45) in $[0, 1]$, BDF(3) mit $h = 0.000 1$, Vergleich mit exakten Lösungen $x(t)$ bzw. $y(t)$ im Bereich $t \in [0, 0.001]$

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, so dass 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 $h > 0.000 5$ überhaupt keine Stellen in der transienten Phase $[0, 0.000 5]$ erfasst werden, also darüber hinweg integriert wird. Wenn man aber mit $h < 0.000 5$ dann Stellen in diesem Bereich hat, so sollte man mit $h \ll 0.000 5$ rechnen, um zumindest die Genauigkeit $h^{1...2}$ zu retten. Ein typisches Beispiel in allen drei Tabellen ist die Schrittweite $h = 0.000 25$, 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      = -29998.000000000000  -39996.000000000000
        14998.500000000000   19997.000000000000

condA =  3.124395031180386e+005

invA   =  1.99970000001133   3.99960000002266
        -1.49985000000850  -2.99980000001699

ans    =  0.9999999999272           0
        0.00000000000728   1.00000000000728

ans    =  9.99999999943349e+003
detA   =  10000

invAs  =  1.999700000000000   3.999600000000000
        -1.499850000000000  -2.999800000000000

ans    =  1      0
        0      1

invAs1=  1.999700000000000   3.999600000000000
        -1.499850000000000  -2.999800000000000

ans    =  1.00000000000728           0
        -0.00000000000364   1.000000000000000
```

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 Gleitpunktarithmetik *double* sehr ordentliche Resultate.

Dazu kommt noch der Unterschied, ob man bei der Übereinstimmung von AA^{-1} mit der Einheitsmatrix die Inverse A^{-1} 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) zur einem Verfahren mit einer dualen Schrittweitensteuerung (Halbierung/ohne Veränderung/Verdopplung) verwendet werden (vergl. Abschnitt 2.4). 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 allen 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. 4.95) 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 sind

```
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
```

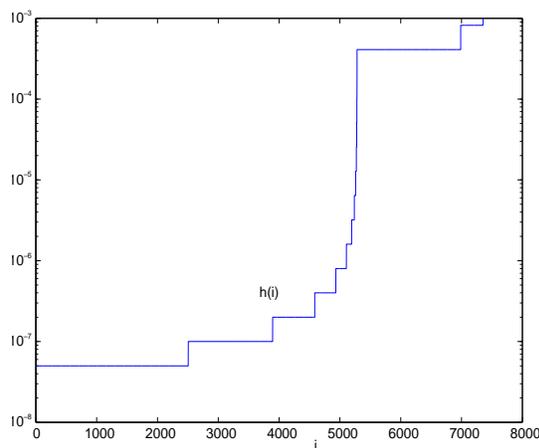


Abb. 4.102:

Datei *bdf12_1.ps*,
 Zum AWP (4.45) 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 Zeitpunkte $t(i)$, $i = 1, \dots, N$, im Integrationsbereich $[0, 1]$, so fallen die meisten Werte in die kurze transiente Phase, ca. 5000. Ihr deutliches Anwachsen kommt anschließend.

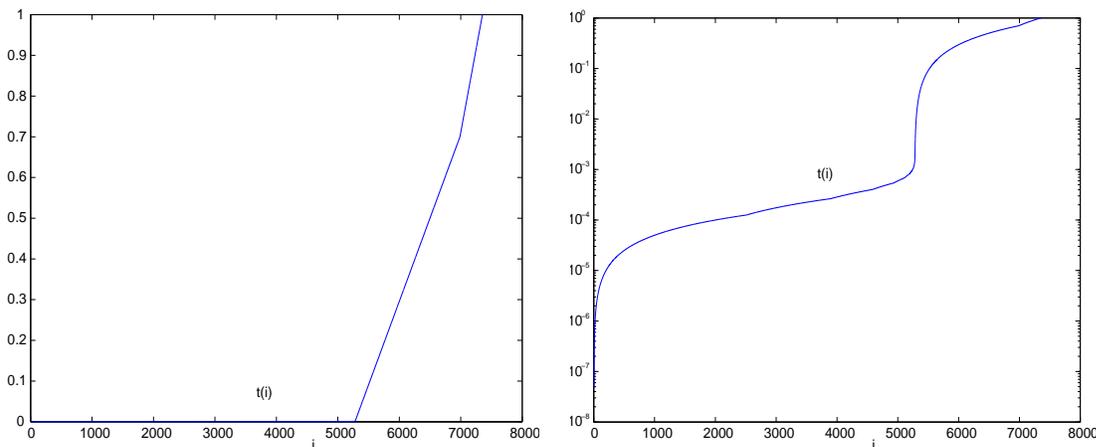


Abb. 4.103: Dateien *bdf12_2a*, *bdf12_2.ps*, Zum AWP (4.45) in $[0, 1]$,
 Verhalten der Integrationspunkte $t(i)$, $i = 1, \dots, N$, bei Anfangsschrittweite $h_0 = 5E-8$, MATLAB-Kommandos `plot(t)` und `semilogy(t)`

Für den Plot der Funktion $h(t)$ ist es sinnvoll, ihn intervallweise vorzunehmen.

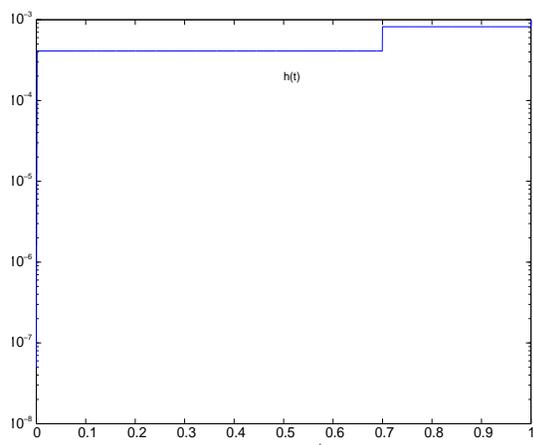


Abb. 4.104:
 Datei *bdf12_3.ps*,
 Zum AWP (4.45) in $[0, 1]$,
 Verhalten der Schrittweite $h(t)$, $t \in [0, 1]$,
 bei Anfangsschrittweite $h_0 = 5E-8$,
 MATLAB-Kommando `semilogy(t,h)`

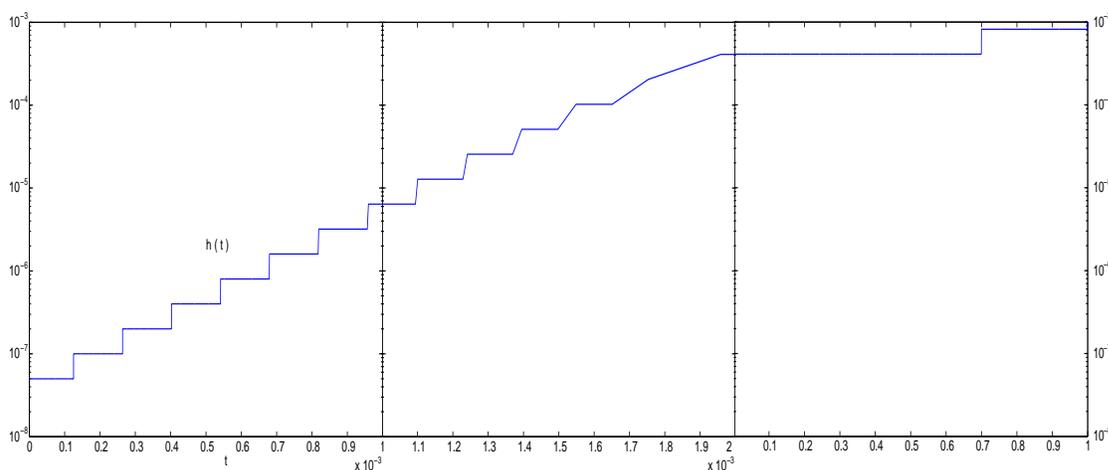


Abb. 4.105: Dateien *bdf12_3a, b, c.ps*, Zum AWP (4.45) in $[0, 1]$,
 Schrittweite $h(t)$, $t \in [0, 1]$, intervallweise zusammengesetzt
 mit $[0, 0.001] \cup [0.001, 0.002] \cup [0.002, 1]$

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 Schrittweitensteuerung 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_1 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} ist 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.

Der modifizierte Algorithmus ist 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 % bei Verdopplung von h bleibt eta(i-1)
    hi = 2*hi;
    verd = 1;
else
    xim2 = xim1;
    xim1 = xhi; % wenn h unveraendert, dann eta(i)
end;
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
[aerr_x,ix] = [3.892179423203857e-007, 5427],   d.h. i=5426
[aerr_y,iy] = [1.946107186512336e-007, 5141]
aerr_xN = 1.924794363894478e-007
aerr_yN = 1.443595769590189e-007

```

Die grafischen Darstellungen der Ergebnisse sind denen in den Abbildungen 4.102 – 4.105 sehr ähnlich. ◀

4.3.2 Steife Systeme und Linearisierung

Wir wollen nun eine 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.

[4.15] Beispiel Wir betrachten das lineare SysgDGL für $\mathbf{x} = (x_1, x_2, x_3)^T$

$$\mathbf{x}' = A \mathbf{x}, \quad \mathbf{x}(0) = (4, 13, 1)^T, \quad (4.46)$$

mit

$$A = \begin{pmatrix} -0.5 & 32.6 & 35.7 \\ 0 & -48.0 & 9.0 \\ 0 & 9.0 & -72.0 \end{pmatrix}. \quad (4.47)$$

Die Systemmatrix A besitzt die drei reellen Eigenwerte $\lambda_1 = -75$, $\lambda_2 = -45$ und $\lambda_3 = -\frac{1}{2}$. Die allgemeine Lösung

$$\begin{aligned} x(t) &= C_1 e^{-t/2} - C_2 e^{-45t} + C_3 e^{-75t}, \\ y(t) &= C_2 e^{-45t} + C_3 e^{-75t}, \\ z(t) &= \frac{1}{3} C_2 e^{-45t} - 3 C_3 e^{-75t}, \end{aligned} \quad (4.48)$$

sowie die exakte Lösung mit der AB

$$x(t) = 15e^{-t/2} - 12e^{-45t} + e^{-75t}, \quad y(t) = 12e^{-45t} + e^{-75t}, \quad z(t) = 4e^{-45t} - 3e^{-75t} \quad (4.49)$$

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 (4.12) 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.

Nach der Merkregel [1.12] sollte man die etwas schärfere Schrittweitenbedingung $h < 1/\max_i |\lambda_i| = 1/75 = 0.013333$ verwenden, um eventuell oszillierendes Verhalten der Näherungslösung zu vermeiden.

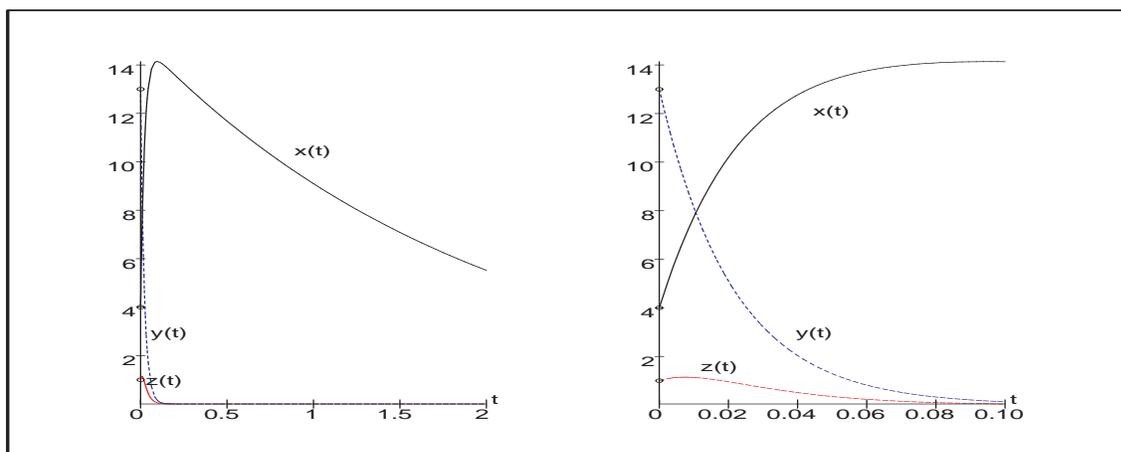


Abb. 4.106: Datei *nsteif70.ps*, Lösungen $x(t), y(t), z(t)$ zum AWP (4.46) in $[0, T]$

Das gegebene AWP soll nun mit dem KRKV gelöst werden.

Dieses hat nach Beispiel [4.7] (c) das Stabilitätsintervall $(-2.785293, 0)$. Hierdurch und durch den betragsgrößten negativen Eigenwert λ_3 wird die Stabilitätsobergrenze für die Schrittweite h festgelegt durch

$$h_{max} = \frac{2.785293}{|\lambda_3|} = 0.037137. \tag{4.50}$$

Diese Grenze muss im Folgenden immer berücksichtigt werden.

Wir fordern nun des weiteren, dass

$$|F(h\lambda_i) - e^{h\lambda_i}| < \varepsilon = 10^{-5}, \quad i = 1, 2, 3, \tag{4.51}$$

$$F(h\lambda) = 1 + h\lambda + \frac{1}{2}(h\lambda)^2 + \frac{1}{6}(h\lambda)^3 + \frac{1}{24}(h\lambda)^4,$$

(vergl. Definition [4.3]).

Ein Maß für die Genauigkeit für die 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 (4.51) mit λ_1 auf die Wahl $h_1 = 0.0035$. Nach 44 Zeitschritten 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 Zeitschrittwahl h_2 ergibt sich aus der Forderung (4.51) 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 (4.51) 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 (4.50) 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 Kenntniss 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.

[4.16] Beispiel Die folgenden Gleichungen beschreiben ein SysgDGL, wie es typischerweise bei der Berechnung der chemischen Reaktion dreier Substanzen auftreten kann.

$$\mathbf{x}' = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} -0.1x_1 + 100x_2x_3 \\ 0.1x_1 - 100x_2x_3 - 500x_2^2 \\ 500x_2^2 - 0.5x_3 \end{pmatrix} \quad (4.52)$$

mit den Anfangsbedingungen $\mathbf{x}(0) = (4, 2, 0.5)^T$.

Um herauszufinden, mit welcher Schrittweite mit 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

$$\tilde{\mathbf{x}}' = A \tilde{\mathbf{x}} + \begin{pmatrix} 99.6 \\ -2099.6 \\ 1999.75 \end{pmatrix}, \quad \tilde{\mathbf{x}}(0) = \mathbf{0}, \quad (4.53)$$

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 für diese 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 SysgDGL muss eine Linearisierung zur Schrittweitenbestimmung jeweils nach einigen Zeitschritten wiederholt werden.

Das AWP (4.52) ist analytisch nicht lösbar. Die numerische Lösung kann in Maple mittels `dsolve` und bei einer Methode für steife gDGL, 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 grob, die Anzahl der Funktionsauswertungen `maxfun` zu klein eingestellt u. ä. Man muss erst an einigen Parametern geeignet "drehen", damit eine numerische Lösung erzeugt wird.

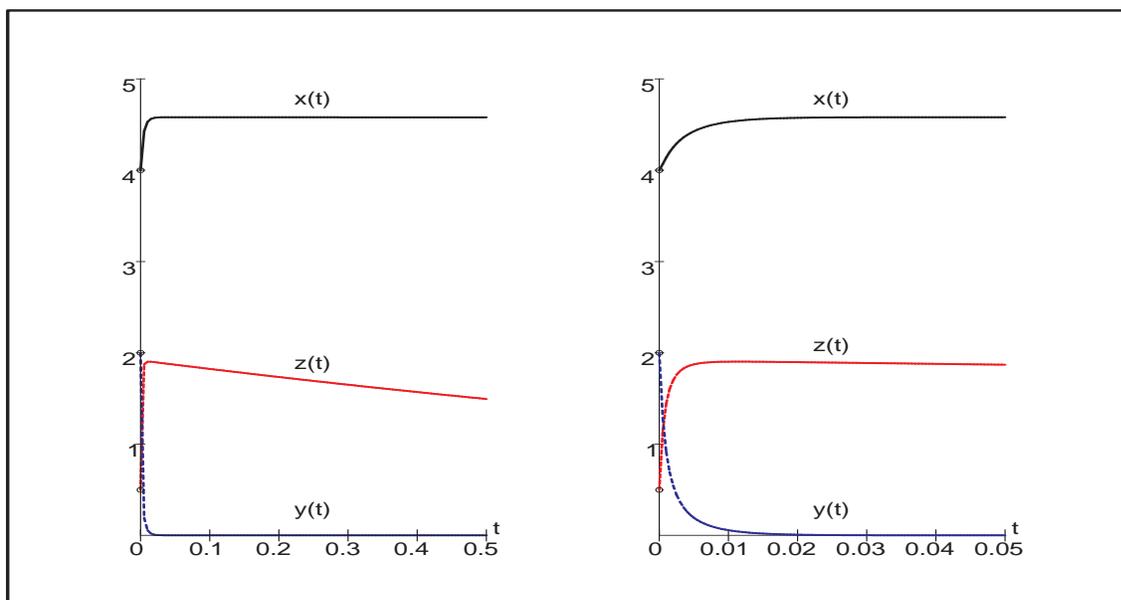


Abb. 4.107: Datei `nsteif71.ps`, numerische Lösungen $x(t), y(t), z(t)$ zum AWP (4.52) in $[0, T]$ mittels `dsolve(..., numeric, method=gear, ...)`

Die transiente Phase der Steifheit ist ungefähr das Anfangsintervall $[0, 0.0015]$.

Das linearisierte AWP (4.53) 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 gDGL 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. 107 und 108 ist im kleinen Intervall $[0, 0.001]$ zu erkennen. Weiter rechts sind die Unterschiede beträchtlich.

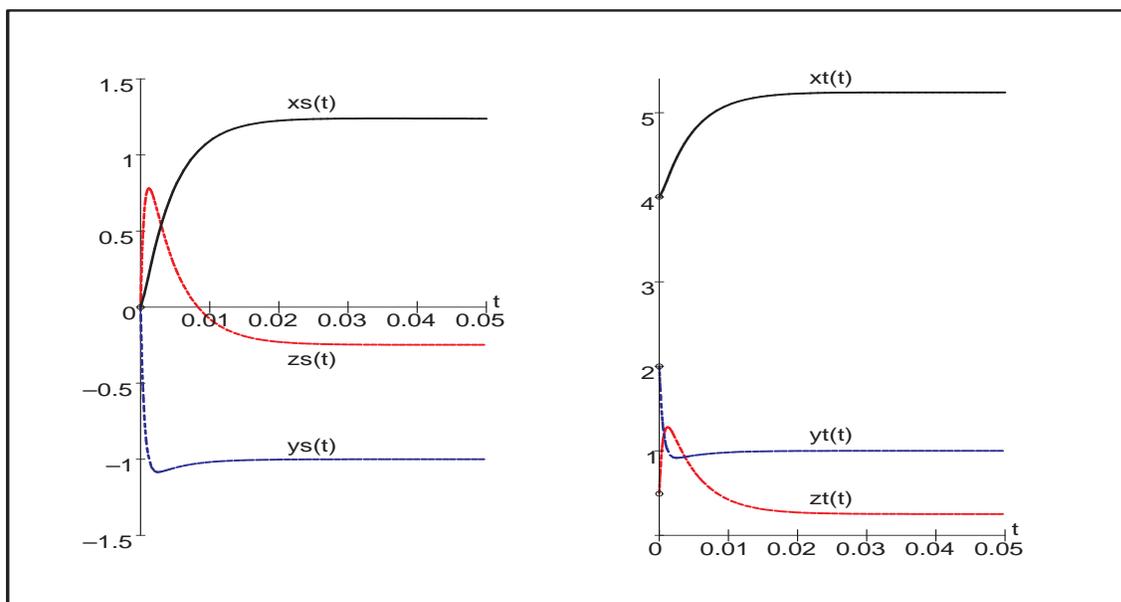


Abb. 4.108: Datei *nsteif72.ps*, exakte Lösung $\mathbf{x}\mathbf{s}(t) = \tilde{\mathbf{x}}(t) = (\tilde{x}(t), \tilde{y}(t), \tilde{z}(t))^T$ zum AWP (4.53) in $[0, 0.05]$ sowie transformierte Funktion $\mathbf{x}t(t) = \tilde{\mathbf{x}}(t) + \mathbf{x}(0)$

Genauso ist ein numerischer Vergleich zwischen $\mathbf{x}(t)$ aus der `dsolve`-Rechnung als sehr gute Näherung zu $\mathbf{x}(t)$ und der transformierte Funktion $\mathbf{x}t(t) = \tilde{\mathbf{x}}(t) + \mathbf{x}(0)$ mit $\tilde{\mathbf{x}}(t)$ aus dem linearisiertem AWP (4.53) für ausgewählte Stellen möglich. Wir nehmen dazu $t_i = i/10\,000, i = 0, 1, \dots, 15$.

$t(i)$	$x(t(i))$	$xt(t(i))$	$\tilde{x}(t(i))$	$\tilde{x}t(t(i))$	$y(t(i))$	$yt(t(i))$
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

Bei der Zeit $t \approx 0.008$ hat man die Abweichungen schon in der ersten Nachkommastelle. Generell ist es problematisch, in der transienten Phase eines steifen SysgDGL zu linearisieren. ◀

5 Übungsserien

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 dem Bestand als auch der jeweiligen Abweichung vom möglichen Höchststand.

Dies ergibt die DGL

$$y' = py(m - y), \quad y = y(t), \quad \text{Anfangsbedingung (AB) } y(0) = y_0.$$

Sie wird auch als Wachstumsgleichung oder logistische Gleichung bezeichnet.

Löse das Anfangswertproblem (AWP).

Hinweis: Trennung der Variablen, Idee der 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 DGL 2. Ordnung

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

Finde die Lösungsmenge der DGL.

Diskutiere die Anfangsbedingungen.

– Pendel wird aus der Lage $\alpha = 0$ angestoßen mit der Geschwindigkeit v_0 .

– 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 Abstands-

bedingung $|OT|^2 = |TP|^2$ 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 DGL

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

Finde die allg. Lösung dieser homogenen DGL unter Nutzung der Substitution

$u = \frac{y}{x}$ sowie mittels Partialbruchzerlegung.

d) Zweispezimodell – ökologisches Gleichgewicht bei Populationen

Dies wird beschrieben durch das Volterrasche System von zwei DGL 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

$$x\ddot{x} = \dot{x}^2 + acx^2 - cx\dot{x} + cx^2\dot{x} - acx^3,$$

$$y\ddot{y} = \dot{y}^2 + acy^2 + ay\dot{y} - ay^2\dot{y} - acy^3.$$

Löse die DGL für $a = 2$, $c = 1$ mit den AB $x(0) = 1$, $y(0) = 3$.

2. Richtungsfelder und Lösungsgebiete

Gegeben sei die explizite DGL 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 DGL eindeutig bestimmt.

Diskutiere die Richtungsfelder, Isoklinen und Lösungen von DGL.

a) $y' = 1 + y$. Was ist auf der Geraden $y = -1$?

b) $y' = \frac{1}{x + y}$. Was ist 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. Bestimme 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 DGL $y' = f(x, y)$ 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.

Diskutiere die folgende Beispiele.

a) $y' = 1 + y$, $y(x) = -1 + ce^x$.

$$-1/y' = 1 + y, \quad y(x) = \sqrt{c - 2x}.$$

$$\begin{aligned} \text{b) } y' &= y/x, \quad y(x) = cx. \\ -1/y' &= y/x, \quad y' = -x/y, \quad x^2 + y^2 = c. \end{aligned}$$

4. Phasenkurve

Für die Lösung $y = y(x)$ der DGL $y' = f(x, y)$ 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 DGI-System

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

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 DGI-System

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

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

$$y'' = f(t, y, y'),$$

wenn man diese durch die Definition $x = y' = \dot{y}$ in das System von zwei DGI

$$\begin{aligned} \dot{y} &= x = f_1(t, x, y), \\ \dot{x} &= \ddot{y} = f_2(t, y, x) = f_2(t, x, y), \end{aligned}$$

überführt und somit die Phasenkurve $(y, x) = (y, \dot{y})$ erhält.

Konstruiere für folgende DGI die Phasenkurven.

- $y' = 1 + y, \quad y(0) = \frac{1}{2}$.
- $y'' + y = 0, \quad y(0) = 0, \quad y'(0) = 1$, Pendel ohne Reibung.
- $y'' + ky' + y = 0, \quad y(0) = y_0, \quad y'(0) = y'_0$, Pendel mit Reibung $k > 0$, z.B. $k = \frac{10}{3}$.

Serie 2 - Theorie, Anwendung und Lösung

1. Aufstellen von DGI.

- Bestimme die DGI 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 DGI noch weitere Lösungen?

- Bestimme die DGI 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.

- Bestimme anhand einer Zeichnung zunächst die zugehörige DGI.

Mögliche Lösung: $yy' = 2$.

- b) Zeichne das Richtungsfeld und gib die Gebiete an, in denen die DGL definiert ist.
- c) 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? Zeichne diese in das Richtungsfeld ein.
- e) 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

Bestimme zu folgenden Kurvenscharen die DGL der orthogonalen Trajektorien

- a) Schar der Kreise $(x - c)^2 + y^2 = c^2$.
- b) Geradenschar $y = 3x + b$. Bestimme die orthogonalen Trajektorien.
- c) Parabelschar $y = ax^2$. Welche Kurven sind das?

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 DGL 1. Ordnung (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 DGI 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) $y'' = xy, y(0) = c_0, y'(0) = c_1$, Airy's Gleichung

Lösung: $y(x) = \frac{1}{6\Gamma(2/3)} (-2 \cdot 3^{5/6} c_1 \pi - 3 \cdot 3^{2/3} c_0 \Gamma(2/3)^2) \text{AiryAi}(x) + 3^{1/3} (2c_1 \pi + 3^{5/6} c_0 \Gamma(2/3)^2) \text{AiryBi}(x)$.

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 DGI höherer Ordnung mit physikalischen Anwendungen (Schwingungs-DGI) 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 DGI genügt der Anfangsbedingungen $y(0) = y'(0) = 0$?

Wie verhält sich die Funktion $y(t)$ für sehr große t ?

$$y''(t) + 6y'(t) + 25y(t) = 3 \sin(4t)$$

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) Bestimme $\lim_{x \rightarrow \infty} y(x)$.

$$y''' + 12y'' + 6y' + y = 2e^{-x}$$

7. Es sei gegeben die lineare DGL $y'' + p_1(x)y' + p_2(x)y = q(x)$.

a) Zeige.

- Sind y_1 und y_2 partikuläre Lösungen dieser DGL, so ist $y_3 = y_1 - y_2$ eine partikuläre Lösung der zugehörigen homogenen DGL.
- Ist y_1 eine partikuläre Lösung der DGL, so ist auch $y_1 + y_h$ eine Lösung für jedes y_h , welches Lösung der zugehörigen homogenen DGL ist.

b) Die obige DGL habe nunmehr speziell die Form $y'' + 5y' + 6y = 6x^2 + 10x$.

Zeige.

- Als eine partikuläre Lösung der inhomogenen linearen DGL lässt sich ein Polynom 2. Grades angeben. Welches?
- Die zugehörige homogene DGL hat Lösungen der Form $y = e^{kx}$ mit geeigneten Werten für k . Bestimme diese. Wieviele linear unabhängige Lösungen lassen sich auf diese Weise finden?
- Gib die insgesamt gefundene Lösungsmenge an.

8. Diskutiere die Lösungen $y(t)$ der parametrischen DGL (Parameteraufgaben).

a) Schwingungsgleichung ohne Einwirkung äußerer Kräfte.

$$y'' + 2hy' + k^2y = 0$$

b) DGL der erzwungenen Schwingung $y(t)$ mit Einwirkung äußerer Kräfte.

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) 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.

Zeige, dass diese Lösung Nullstellen und Extremstellen besitzt (ggf. numerische Berechnung).

$$y'''' + 3y''' + 3y'' + y' = 10e^{-2t}$$

9. Betrachtung von nichtlinearen DGL.

a) Diskutiere die Lösung der DGL, die ein relatives Extremum an der Stelle e hat.

Von welcher Art ist dieses Extremum?

$$x^2 y' - y^2 - xy = 0$$

b) Diskutiere die Lösungsschar der DGL.

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

c) Ricatti'sche DGL $y' = x - y^2$, $y(0) = 0$, $0 \leq x \leq 1$.

10. Gegeben ist die nichtlineare DGL $y' = y^2 x^3 + xy$.

- a) Bestimme den Typ der DGL und transformiere sie in eine lösbare Form.
 b) Ermittle das allgemeine Integral. **Hinweis:** $z' + xz = -x^3$.
 c) Zeige, dass alle Lösungen symmetrisch zur y -Achse sind.
 d) Bestimme diejenige Integralkurve, die durch den Punkt $P(0, 1)$ geht.
 e) 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 0 besitzt.
 Von welcher Art ist dieses?

11. Lösung von nichtlinearen DGL 1. Ordnung.

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

Lösung: $y(x) = \frac{2}{x}$

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

Lösung: $y(x) = \sqrt{2x + 1}$

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

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

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

Lösung: $y(x) = \arctan(x)$

e) $y' = -200xy^2$, $y(-1) = \frac{1}{101}$

Lösung: $y(x) = \frac{1}{1 + 100x^2}$

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

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

g) Ricatti'sche DGL $y' = 1 - y^2$, $y(0) = 0$, $0 \leq x \leq 1$.

Transformiere die DGL auf eine Gleichung 2. Ordnung und löse diese.

Hinweis: 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 DGL 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 DGL

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

Der Parameter ε legt die Einschwingdauer fest.

$$\text{Lösung für } \varepsilon = 0: y(t) = \dot{y}_0 \sin(t) + y_0 \cos(t)$$

$$\text{AB: } y(0) = 0, \dot{y}(0) = -0.05, y(0) = 2, \dot{y}(0) = 0$$

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

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

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) Emden's 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 DGL

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

l) Mathieu's DGL für ein Pendel mit zeitabhängiger Länge $l(t)$

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

$$a = 1.0, q = 0.1 \quad \text{aufschaukelnde Schwingung}$$

$$a = 1.2, q = 0.1 \quad \text{gedämpfte Schwingung}$$

Serie 3 - Einzschritverfahren

1. Gegeben sei die explizite DGL 1. Ordnung (AWP)

$$y' = f(x, y), \text{ AB } y(0), x \geq 0, f \in C^2(\mathbb{R}^2).$$

Für seine Lösung wenden wir das explizite Einzschritverfahren (ESV) von Euler-Cauchy (Polygonzugverfahren, PZV) an

$$y_{n+1} = y_n + hf(x_n, y_n), n = 0, 1, \dots, x_n = nh, y_0 = y(0).$$

- a) Löse die DGL $y' = y$ 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? Diskutiere den Fehler.
 - Das Integrationsintervall sei $[0,1]$ und $h = 1/N$.
Welcher Term beschreibt y_N ?
Welche Grenzwerte ergeben sich für beide AB für $N \rightarrow \infty$?
- b) Welche Lösung liefert das PZV für die DGL $y' = f(x), y(a) = 0, x \in [a, b]$?
- 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 DGL?
 - Angenommen, die DGL beschreibt den freien Fall eines Körpers.
Wie verhält sich die gefundene Lösung im Vergleich zur Wirklichkeit?
 - 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 DGL? 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 Stokeschen Widerstandsgesetzes $R(v) = K_1v$ (K_1 Materialeigenschaften des Mediums/Geometrie des Körpers) ergibt sich aus dem Kraftgesetz

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

- Die abgeleitete Modellgleichung für die Geschwindigkeit $v(t)$ sei $\dot{v} = c + k_1v, c > 0$.
- 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 Widerstand-

gesetzes $R(v) = K_2 v^2$ (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 $v(0) = v_0$, $k_2 < 0$, $c > 0$.

Experimentiere mit dem PZV bei verschiedenen Startwerten und Schrittweiten (auch relativ große). 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 der rechten Seite der DGL anwenden.

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

$$\ddot{z} = -g,$$

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

$$\ddot{z} = -3g + 2\frac{g}{r_E}z,$$

- $z_0 = r_E$, $g_0 = g$,

$$\ddot{z} = -\frac{gr_E^2}{z^2}.$$

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

- Welche konstante Funktion ist Lösung der 2. Modellgleichung?

Finde ihre allgemeine Lösung.

- Berechne die Fallzeit t_F und Endgeschwindigkeit $\dot{z}(t_F)$ bei einer Fallhöhe H und gegebenen AB für die Modellgleichungen 1, 2.

- 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. Löse das AWP mit dem PZV

$$y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0, 1, \dots, N-1, \quad x_n = nh, \quad y_0 = y(0),$$

$$h = \frac{x_N - x_0}{N}, \quad [x_0, x_N] = [a, b].$$

Berechne den Fehler $|y(x_N) - y_N|$, $N = 5, 10, 20$.

Woran erkennt man die Konvergenzordnung 1 des PZV?

Vergleiche den Wert $|y(x_N) - y_N|$ mit der Formel für die Abschätzung des globalen Diskretisierungsfehlers $|e_N|$ gemäß

$$|e_n| \leq (e^{K(x_n-x_0)} - 1) \frac{M_2}{2K} h, \quad n = 1, 2, \dots, N, \quad e_0 = 0,$$

$$K = \max_{a \leq x \leq b} \left| \frac{\partial f}{\partial y}(x, y(x)) \right|,$$

$$M_2 = \max_{a \leq x \leq b} |y''(x)|.$$

Werte die Güte der Abschätzung $|e_n|$. Welche geometrische Deutung haben die Größen K und M_2 . Welchen Einfluss hat das auf die Wahl der Schrittweite h ?

a) $y' = x - y^2$, $y(0) = 0$, $0 \leq x \leq 1$, Ricatti-DGL, $y(x) = \frac{\sqrt{3} \operatorname{AiryAi}(1,x) + \operatorname{AiryBi}(1,x)}{\sqrt{3} \operatorname{AiryAi}(x) + \operatorname{AiryBi}(x)}$,
 $y(1) = 0.455\,544\,526\,081\,874$.

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. Löse das AWP mit dem PZV gemäß Aufgabe 2.

Die rechte Seite der DGL 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 $\Delta y_n = y_{n+1} - y_n$. Zeige, dass $|\Delta y_n| \leq (1 + hK)^n |\Delta y_0|$.

Werte die Abschätzung, indem man für die DGL $y' = y$, $y(0) = 1$, und $y' = -y$, $y(0) = 1$, jeweils mit der exakten rekursiven Formel $\Delta y_n = C(h)\Delta y_0$ vergleicht.

Serie 4 - Explizite Einschrittverfahren

1. Die DGL $y' = y(1 - y)$ 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) 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) Wende das PZV

$$y_{n+1} = y_n + h y_n (1 - y_n), \quad n = 0, 1, 2, \dots,$$

mit $y_0 \geq 0$ 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) Teste auf analoge Weise für diese DGL das klassische Runge-Kutta-Verfahren

4. Ordnung (KRKV).

2. Gebe eine geometrische Interpretation der ESV an (Schema mit Zuwächsen hk_i).

a) Euler-Cauchy-Verfahren (PZV)

$$\begin{aligned}k_0 &= f(x_n, y_n), \\ y_{n+1} &= y_n + hk_0.\end{aligned}$$

b) Collatz-Verfahren

(modif. PZV=MPZV)

$$\begin{aligned}k_0 &= f(x_n, y_n), \\ k_1 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_0\right), \\ y_{n+1} &= y_n + hk_1.\end{aligned}$$

c) Heun-Verfahren

(HV, verbess. PZV=VPZV)

$$\begin{aligned}k_0 &= f(x_n, y_n), \\ k_1 &= f(x_n + h, y_n + hk_0), \\ y_{n+1} &= y_n + \frac{h}{2}(k_0 + k_1).\end{aligned}$$

d) KRKV

$$\begin{aligned}k_0 &= f(x_n, y_n), \\ k_1 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_0\right), \\ k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right), \\ k_3 &= f(x_n + h, y_n + hk_2), \\ y_{n+1} &= y_n + \frac{h}{6}(k_0 + 2k_1 + 2k_2 + k_3).\end{aligned}$$

3. Welche Integrationsformeln vom Newton-Cotes-Typ werden durch folgende ESV verallgemeinert?

a) Collatz-Verfahren, modifiziertes PZV (1960).

b) Heun-Verfahren, verbessertes PZV (1900).

c) (3,3)-Verfahren von Kutta (1901).

d) KRKV (1895).

4. Löse das Anfangswertproblems $y' = x - y$, $y(0) = 1$, $x \in [0, 1]$.

a) Verwende die ESV von Euler-Cauchy, Collatz und Heun mit den Schrittweiten $h = 0.2, 0.1$.

b) Vergleiche die Lösungen miteinander sowie mit der exakten Lösung

$$y(x) = 2e^{-x} + x - 1.$$

c) Berechne explizit für alle drei Verfahren $y_{n+1} = F(x_n, y_n; h, f)$ die Verfahrensfunktion F . Was stellt man beim Vergleich von Collatz-Verfahren und Heun-Verfahren fest? Begründung.

5. Löse das AWP $y' = x - y$, $y(0) = 1$, $x \in [0, 1]$, $y(x) = 2e^{-x} + x - 1$, mit dem RK-Verfahren (RKV) bei einer Schrittweite $h = 0.2$.

- a) Gebe explizit das Verfahren in der Form $y_{n+1} = F(x_n, y_n; h, f)$ an.
 b) Um eine vernünftige (variable) Schrittweite für das RKV anzuwenden, ist man durch heuristische Betrachtungen zur **Schrittkenzahl** κ gelangt.

Diese ist definiert durch

$$\kappa = 2 \left| \frac{k_2 - k_1}{k_1 - k_0} \right|$$

und sollte sich zwischen 0.1 und 0.2 bewegen.

Weise nach, dass dies durch entsprechende Wahl der Schrittweite h erreichbar ist.

Zeige dazu, dass gilt

$$\kappa \approx h \left| \frac{\partial f}{\partial y} \right|.$$

- c) Vervollständige die Tabelle der Schrittkenzahlen und vergleiche.

x	$2 \left \frac{k_2 - k_1}{k_1 - k_0} \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 DGI $y' = y$ und $y' = -y$ mit der gleichen Schrittweite gerechnet werden?
 e) Man untersuche analog das AWP $y' = ty$, $y(0) = 1$, $0 \leq t \leq 2$.

6. Gegeben sei die DGI

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

- a) Finde mittels Trennung der Variablen die exakte Lösung $y(x) \geq 0$.
 b) Bestimme $f_y = \partial f / \partial y$ sowie mittels der Schrittkenzahl $\kappa = h |f_y| = 0.1$ eine Formel für die Schrittweite h . Wie groß ist die Anfangsschrittweite h_0 ?
Hinweis: Darstellung von $h = \alpha x + \beta$.
 c) 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) 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?

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 DGL $y' = f(x, y)$ betrachten wir auf Basis des PZV $\bar{y}_{n+1} = y_n + hf(x_n, y_n)$ die Verbesserung als Heun-Verfahren (Sehnentrapezregel) in verschiedenen Formen.

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}(f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})) = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_n + hf(x_n, y_n))] \\ &= \bar{y}_{n+1} + \frac{h}{2}(f(x_{n+1}, \bar{y}_{n+1}) - f(x_n, y_n)) = \bar{y}_{n+1} + \Delta, \quad \Delta \text{ Korrektur} \\ &= y_n + hk, \quad k = \frac{1}{2}(k_0 + k_1), \quad k_0 = f(x_n, y_n), \quad k_1 = f(x_{n+1}, \bar{y}_{n+1}), \\ &\quad k_i \text{ Anstiege, } hk_i \text{ Zuwächse} \\ &= y_n + h\Phi(x_n, y_n; h, f), \quad k = \Phi \text{ Inkrementfunktion} \\ &= F(x_n, y_n; h, f), \quad F \text{ Verfahrensfunktion} \\ &= E(x_n; h, f)y_n, \quad E \text{ Verstärkungs-, Transformations-, Stabilitätsfunktion.} \end{aligned}$$

Zeige. Falls $f(x, y)$ stetig differenzierbar ist, so genügt die Verfahrensfunktion $F(x, u; h, f)$ des Heun-Verfahrens einer **Lipschitz-Bedingung**.

8. Notiere das KRKV komponentenweise für die DGL 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 DGL-System 1. Ordnung.

Beispiel: $y''' = -\frac{1}{2}y''^3$, $y(0) = 1, y'(0) = 3, y''(0) = 1$, $y(x) = \frac{4}{3}(x + 1)^{3/2} + x - \frac{1}{3}$, allg. 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. Gebe die RK-Formeln an für das DGI-System 1. Ordnung

$$y' = f(x, y, z), \quad y(x_0) = y_0, \quad z' = g(x, y, z), \quad z(x_0) = z_0.$$

Nutze und vereinfache diese bei der Lösung von

a) $y'' = f(x, y, y'), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0,$

b) $y'' = f(x, y), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0.$

Löse die AWP mittels RKV.

c) $y'' - 3y' + 5y - 1 = 0, \quad y(0) = 0, \quad y'(0) = 1, \quad x \in [0, 1], \quad h = 0.2,$

das Basissystem 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) $y'' = 2x + y, \quad y(0) = y'(0) = 0, \quad x \in [0, 1], \quad h = 0.2,$

die exakte Lösung ist $y(x) = e^x - e^{-x} - 2x.$

10. Für die Lösung $y(1)$ des AWP (Ricatti-DGI) $y' = x - y^2, \quad y(0) = 0,$

$y(x) = \frac{\sqrt{3} \text{AiryAi}(1,x) + \text{AiryBi}(1,x)}{\sqrt{3} \text{AiryAi}(x) + \text{AiryBi}(x)},$ liegen die Näherungen $y_N, \quad N = 5, 10, 20, 40,$ des KRKV vor.

N	h	$y_N(x_N = 1)$	$e_N(h)$	$\frac{e_N(h)}{e_{2N}(h/2)}$
5	0.2	0.455 533 498 11	-1.10 E-5	16.1
10	0.1	0.455 543 842 70	-6.84 E-7	16.1
20	0.05	0.455 544 483 62	-4.24 E-8	16.0
40	0.025	0.455 544 523 43	-2.65 E-9	16.0

Exakter Wert (15 Stellen): $y(1) = 0.455 544 526 081 874$

- a) Ermittle Schätzungen des globalen Diskretisierungsfehlers e_N nach dem Runge-Prinzip und vergleiche diese mit dem wahren Fehler $e_N, \quad N = 10, 20, 40.$
- b) Verbessere die Näherungswerte der jeweiligen Feinrechnung durch Richardson-Extrapolation (Ergebnis: y_N^*).
- c) Ermittle experimentell die Konvergenzordnung der verbesserten Lösungen $y_N^*.$

11. 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 DGL-System 1. Ordnung mit konstanten Koeffizienten

$$y' = Ay, \quad A(m, m)\text{-Matrix}.$$

- a) Gebe die allgemeine Lösung in Matrixschreibweise an.
- b) Wende das Euler-Cauchy-Verfahren an und notiere es in der Matrixform $y_{n+1} = My_n$, $n = 0, 1, 2, \dots$, mit der Transformations- oder Verfahrensmatrix $M = M(x_n, y_n; h, A)$.
- c) Notiere das Heun-Verfahren und das KRKV ebenfalls in Matrixform und bestimme die Transformationsmatrizen M .

13. Gegeben ist das AWP (DGL-System 1. Ordnung).

Bestimme $y_1(1)$ und $y_2(1)$ näherungsweise mittels ESV bis auf eine Genauigkeit $TOL = 10^{-4}$.

a) Lineares DGL-System

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

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

$$\text{Lösung: } y_1(x) = \cosh(x^2/\sqrt{2}),$$

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

b) Nichtlineares DGL-System

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

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

c) Autonomes nichtlineares DGL-System, Ricatti-DGL

$$y_1' = y_2 - y_1^2, \quad y_1(0) = 0,$$

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

d) Steifes lineares DGL-System

$$y_1' = -29998y_1 - 39996y_2, \quad y_1(0) = 1,$$

$$y_2' = 14998.5y_1 + 19997y_2, \quad y_2(0) = 1,$$

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

$$y_2(x) = -3.5e^{-10000x} + 4.5e^{-x}, \quad \text{asymptotisch stabil.}$$

14. Schreibe ein Computerprogramm für das Euler-Cauchy-Verfahren, das ein AWP mit einer vorgegebenen Schrittweite h löst. Löse die AWP.

a) $y' = y(1 - y)$, $y(x) = y_0$, $0 \leq x \leq 10$, $y(x) = \frac{y_0 e^x}{1 + y_0(e^x - 1)}$,
 mit $h = 0.25, 0.5, 1.0, 2.0, 2.5$ und $y_0 = -1, 0, 0.5, 1, 2$.

15. Schreibe ein Computerprogramm mit ausgewählten (4,4)-RKV, die AWP für DGL 1. Ordnung bei einer Startschrittweite h mit einer gegebenen Genauigkeit $\varepsilon > 0$ lösen.

Notiere dazu folgende Programmteile als Prozeduren:

- RK-Schritt von x bis $x + h$ bei gegebenem h ,
- Schrittsteuerung durch Grob- und Feinrechnung, Algorithmusskizze:

k_h - Zuwachs bei Grobrechnung in $[x_n, x_n + h]$,

$k'_{h/2}, k''_{h/2}$ - Zuwächse bei Feinrechnung in $[x_n, x_n + \frac{h}{2}]$ bzw. $[x_n + \frac{h}{2}, x_n + h]$,

$$Err = \frac{k'_{h/2} + k''_{h/2} - k_h}{15}, \quad \text{Test auf } |Err| \in [\varepsilon/50, \varepsilon],$$

falls $|Err| > \varepsilon$, dann Schritt ab x_n mit $h/2$ wiederholen,

falls $|Err| \leq \varepsilon$, dann Schritt akzeptieren, aber bei $|Err| < \varepsilon/50$ für nächsten Schritt Verdopplung der Schrittweite,

bei Akzeptanz $y_{n+1} = y_n + k'_{h/2} + k''_{h/2} + Err$.

Motiviere die Idee der Festlegung des Toleranzintervalls $[\varepsilon/50, \varepsilon]$.

Teste für selbstgewählte 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

$\frac{2}{5}$	$\frac{2}{5}$			
$\frac{3}{5}$	$-\frac{3}{20}$	$\frac{3}{4}$		
1	$\frac{19}{44}$	$-\frac{15}{44}$	$\frac{40}{44}$	
	$\frac{55}{360}$	$\frac{125}{360}$	$\frac{125}{360}$	$\frac{55}{360}$

e) Gill-Modifikation der RK-Formel, $c = 1/\sqrt{2}$

$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	$c - \frac{1}{2}$	$1 - c$		
1	0	$-c$	$1 + c$	
	$\frac{1}{6}$	$\frac{1-c}{3}$	$\frac{1+c}{3}$	$\frac{1}{6}$

f) RKV mit minimalen Diskretisierungsfehler

$$\alpha_2 = \frac{2}{5}, \quad \alpha_3 = \frac{14 - 3\sqrt{5}}{16}, \quad \beta = \frac{1 - \alpha_2}{(\alpha_3 - \alpha_2)(6\alpha_2\alpha_3 - 4(\alpha_2 + \alpha_3) + 3)}$$

$\alpha_2 = \frac{2}{5}$	$\frac{2}{5}$			
$\alpha_3 = \frac{14-3\sqrt{5}}{16}$	$\alpha_3 - \beta_{32}$	$\beta_{32} = \frac{\alpha_3(\alpha_3 - \alpha_2)}{2\alpha_2(1 - 2\alpha_2)}$		
1	$1 - \beta_{42} - \beta_{43}$	$\beta_{42} = \frac{\beta(-1 + \alpha_2 + \alpha_3 - (2\alpha_3 - 1)^2)}{2\alpha_2}$	$\beta_{43} = \frac{\beta(1 - 2\alpha_2)(1 - \alpha_3)}{\alpha_3}$	
	$1 - \gamma_2 - \gamma_3 - \gamma_4$	$\gamma_2 = \frac{2\alpha_3 - 1}{12\alpha_2(\alpha_3 - \alpha_2)(1 - \alpha_2)}$	$\gamma_3 = \frac{1 - 2\alpha_2}{12\alpha_3(\alpha_3 - \alpha_2)(1 - \alpha_3)}$	$\gamma_4 = \frac{1}{2} + \frac{2(\alpha_2 + \alpha_3) - 3}{12(1 - \alpha_2)(1 - \alpha_3)}$

16. Schreibe ein Computerprogramm für das KRKV, das AWP für DGI-Systeme

1. Ordnung mit einer gegebenen Genauigkeit TOL löst.

Notiere dazu folgende Programmteile als Prozeduren:

- RK-Schritt von x bis $x + h$ bei gegebenem h ,
- Schrittsteuerung durch Grob- und Feinrechnung.

17. Löse mittels PZV bzw. KRKV das autonome DGI-System 1. Ordnung für $t \in [0, T]$

$$u' = -u + 25v, \quad u(0) = 0,$$

$$v' = -2v, \quad v(0) = c, \quad c > 0,$$

$$\text{Lösung: } u(t) = 25c(e^{-t} - e^{-2t}),$$

$$v(t) = ce^{-2t}, \quad \text{asymptotisch stabil.}$$

Teste $c = 0.1, 0.02, 0.01$ und $N = 200(200)1000$, $h = T/N$, $T = 10, 20$.

Das AWP sei nun erweitert durch eine moderate Nichtlinearität zu

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

$$v' = -2v + u\sqrt{u^2 + v^2}, \quad v(0) = c, \quad c > 0.$$

Teste das RKV mit den Größen

$$c = 0.1, 0.02; \quad N = 500, 700, 1000; \quad T = 6, 10,$$

$$c = 0.01, 0.0105; \quad N = 500, 700, 1000; \quad T = 15, 30$$

$$c = 0.0106; \quad N = 700, 1000, 2000; \quad T = 10, 20, 50, 54, 55, 65$$

$$c = 0.0105556757; \quad N = 700, 1000, 2000; \quad T = 40, 48, 50, 55, 56, 59, 60, 61, 68, 69, 70, 94$$

Vergleiche die Ergebnisse mit Berechnungen durch ein Computeralgebrasystem (CAS),

z.B. Maple, MATLAB, *Mathematica* oder das Programm MAYA (Teil DGI).

Serie 5 - Implizite Einschrittverfahren

1. Zeige unter entsprechenden Voraussetzungen, dass das implizite (1,1)-Euler-Verfahren (Gear(1)-Verfahren)

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \quad n = 0, 1, \dots, \quad y_0 \text{ gegeben, d.h.}$$

$$y_{n+1} = y_n + hk_1, \quad k_1 = f(x_n + h, y_n + hk_1),$$

- a) konsistent mit Ordnung 1 ist,
b) konvergent mit Ordnung 1 ist.

2. Die sogenannten θ -Verfahren sind parametrische ESV

$$y_{n+1} = y_n + h[(1 - \theta)f(x_n, y_n) + \theta f(x_{n+1}, y_{n+1})], \quad \text{d.h.}$$

$$y_{n+1} = y_n + h[(1 - \theta)f(x_n, y_n) + \theta k_1],$$

$$k_1 = f(x_{n+1}, y_{n+1}) = f(x_n + h, y_n + h[(1 - \theta)f(x_n, y_n) + \theta k_1]),$$

mit dem reellen Parameter $\theta \geq 0$. Sei $f \in C^2(\mathbb{R}^2)$.

Welche Konsistenzordnung besitzen diese Verfahren? Man beachte die Sonderfälle $\theta = 0$, $\theta = \frac{1}{2}$ (Sehnentrapezregel, Heun, HV) und $\theta = 1$.

Oft wird auch folgende Parameterkonstellation verwendet.

$$y_{n+1} = y_n + h\left[\frac{1}{2}f(x_n, y_n) + \frac{1}{2}f(x_{n+1}, \tilde{y}_{n+1})\right], \quad \text{d.h.}$$

$$y_{n+1} = y_n + h\left[\frac{1}{2}f(x_n, y_n) + \frac{1}{2}k_1\right], \quad k_1 = f(x_{n+1}, \tilde{y}_{n+1}) = f(x_n + h, y_n + hk_1).$$

Es besitzt die Konsistenzordnung 2.

3. Zu den θ -Verfahren gehören auch die so genannten **One-leg-Verfahren**

$$y_{n+1} = y_n + hf((1 - \theta)x_n + \theta x_{n+1}, (1 - \theta)y_n + \theta y_{n+1}), \quad 0 \leq \theta \leq 1.$$

Für $\theta = 0.5$ erhält man die Tangententrapezregel (Collatz, MPZV).

Untersuche seine Konsistenzordnung.

4. Zeige unter entsprechenden Voraussetzungen die Konsistenz und Konvergenz mit Ordnung 2 des impliziten (1,2)-Gauß-Verfahrens (Mittelpunktsregel)

$$y_{n+1} = y_n + hf\left(x_n + \frac{h}{2}, \frac{1}{2}y_n + \frac{1}{2}y_{n+1}\right), \quad \text{d.h.}$$

$$y_{n+1} = y_n + hk_1, \quad k_1 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right).$$

5. Das **linear-implizite** Euler-Verfahren für autonome DGI-Systeme $y' = f(y)$ lautet

$$y_{n+1} = y_n + h \left[I - h \frac{\partial f}{\partial y}(y_n) \right]^{-1} f(y_n), \quad n = 0, 1, 2, \dots,$$

mit der Einheitsmatrix I .

- a) Unter welchen Voraussetzungen ist das Verfahren konsistent?
- b) Bestimme die Stabilitätsfunktion $\mu(H) = E(\lambda, h)$, $H = \lambda h$.
Ist das Verfahren A-stabil?
- c) 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 DGI?
- b) Notiere für die DGI das implizite (1,1)-Euler-Verfahren

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \quad n = 0, 1, \dots, \quad y_0 = 0.$$

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

$$\sum_{j=0}^r \alpha_{r-j} y_{i+1-j} - h \sum_{j=0}^r \beta_{r-j} f_{i+1-j} = 0, \quad \alpha_r = 1, \quad i \geq r-1, \quad f_k = f(x_k, y_k), \quad (5.1)$$

y_0, y_1, \dots, y_{r-1} gegeben.

Eine zweite etwas anschaulichere, aber spezielle Darstellung ist

$$y_{i+p} = y_{i-q} + h \sum_{j=0}^r b_j f_{i-j}, \quad p \geq 0, \quad q \geq 0, \quad i \geq s = \max(r, q), \quad y_0, y_1, \dots, y_{s+p-1} \text{ geg.} \quad (5.2)$$

Eine dritte besondere Form ist

$$\alpha_r y_{i+1} + \alpha_{r-1} y_i + \dots + \alpha_0 y_{i+1-r} = h \beta_r f_{i+1}, \quad \alpha_r, \beta_r \neq 0, \quad i \geq r-1, \quad y_0, \dots, y_{r-1} \text{ geg.} \quad (5.3)$$

Daraus kann man die grundlegenden Klassen von MSV bei entsprechender Wahl von p, q, r ableiten.

Formel (5.2)

$p \geq 1$: 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 (5.3) : Backward Differentiation Formulas (BDF), steif-stabile Methoden von Gear.

Bei der Herleitung der Formeln (5.1), (5.2), (5.3) wird gemäß ihrem interpolativen Charakter (Newton-Cotes-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 (5.1)-(5.3) kann in eindeutiger Weise mittels Rückwärtsdifferenzen dargestellt werden.

Koeffizienten der expliziten Adams-Bashforth(n)-Verfahren n -ter Ordnung.

$$y_{i+1} = y_i + h \sum_{j=0}^r b_j f_{i-j}, \quad i = r, r + 1, \dots, \quad y_0, y_1, \dots, y_r \text{ gegeben.}$$

r	Ordnung n Stufen	Fehler- konstante $c h^n f^{(n+1)}$	K	j						
				0	1	2	3	4	5	
				$K * b_j$						
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.

$$y_i = y_{i-1} + h \sum_{j=0}^r b_j f_{i-j}, \quad i = r, r + 1, \dots, \quad y_0, y_1, \dots, y_{r-1} \text{ gegeben.}$$

r	Stufen s	Ordnung n	Fehler- konstante $c h^n f^{(n+1)}$	K	j						
					0	1	2	3	4	5	
					$K * b_j$						
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	

Koeffizienten der impliziten Milne-Simpson-Verfahren n -ter Ordnung.

$$y_i = y_{i-2} + h \sum_{j=0}^r b_j f_{i-j}, \quad i = r, r + 1, \dots, \quad y_0, y_1, \dots, y_{r-1} \text{ gegeben.}$$

r	Stufen s	Ordnung n	K	j							
				0	1	2	3	4	5	6	
				$K * b_j$							
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.

$$y_{i+1} = y_{i-1} + h \sum_{j=0}^r b_j f_{i-j}, \quad i = r, r+1, \dots, \quad y_0, y_1, \dots, y_r \text{ gegeben.}$$

r	Ordnung	K	j			
	n Stufen		0	1	2	3
			$K * b_j$			
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.

$$\alpha_r y_{i+1} + \alpha_{r-1} y_i + \dots + \alpha_0 y_{i+1-r} = h \beta_r f_{i+1}, \quad i = r-1, r, \dots, \quad y_0, y_1, \dots, y_{r-1} \text{ gegeben.}$$

r	Ordnung	β_r	j						
	n Stufen		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

a) Darstellung von MSV mit Rückwärtsdifferenzen.

- Adams-Bashforth-Formeln

$$y_{i+1} = y_i + h \sum_{j=0}^r b_j f_{i-j} = y_i + h \sum_{j=0}^r c_j \nabla^j f_i.$$

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 $\nabla^j f_i$. Überprüfe die Richtigkeit der angegebenen Koeffizienten c_j für $r = 0, 1, 2$.

- Adams-Moulton-Formeln

$$y_i = y_{i-1} + h \sum_{j=0}^r b_j f_{i-j} = y_{i-1} + h \sum_{j=0}^r c_j \nabla^j f_i.$$

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	$-\frac{1}{2}$	$-\frac{5}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{27}{1440}$

Überprüfe die Richtigkeit der angegebenen Koeffizienten c_j für $r = 0, 1, 2$.

- Milne-Simpson-Formeln

$$y_i = y_{i-2} + h \left(2f_{i-1} + \frac{1}{3} \nabla^2 f_i - \frac{1}{90} \nabla^4 f_i - \frac{1}{90} \nabla^5 f_i - \frac{37}{3780} \nabla^6 f_i \right).$$

Überprüfe die Richtigkeit bis zur Formel 4. Ordnung.

- BDF, Gear-Formeln

$$\frac{1}{r} \nabla^r y_{i+1} + \frac{1}{r-1} \nabla^{r-1} y_{i+1} + \dots + \nabla^1 y_{i+1} = h f_{i+1}.$$

Überprüfe die Richtigkeit für $r = 1, 2, 3$.

2. Leite die Konsistenzordnung des klassischen 3-stufigen Adams-Bashforth-Verfahrens (1855) her.

$$y_{n+1} = y_n + \frac{h}{12} (23f_n - 16f_{n-1} + 5f_{n-2}), \quad n = 2, 3, \dots, \quad y_0, y_1, y_2 \text{ gegeben.}$$

3. Löse das AWP (Ricatti-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: Nutze für die Anlaufrechnung das KRKV.

b) **Prädiktor-Korrektor-Verfahren** von Milne 6. Ordn. mit Schrittweite $h = 0.1$.

$$\bar{y}_{n+1} = y_{n-5} + \frac{h}{10} (33f_n - 42f_{n-1} + 78f_{n-2} - 42f_{n-3} + 33f_{n-4}),$$

$$y_{n+1} = y_{n-3} + \frac{h}{45} (14\bar{f}_{n+1} + 64f_n + 24f_{n-1} + 64f_{n-2} + 14f_{n-3}),$$

$$\bar{f}_{n+1} = f(x_{n+1}, \bar{y}_{n+1}), \quad n = 5, 6, \dots, \quad y_0, \dots, y_5 \text{ gegeben.}$$

4. 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 $n = 1(1)4$ bei einer Schrittweite $h = 0.1$.

Die benötigten Startwerte sind durch eine Anlaufrechnung vorgegeben

(3 Nachkommastellen).

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. 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 $\phi(\xi; z) = \rho(\xi) - z\sigma(\xi)$, $z = \lambda h$.Deshalb wird das Verfahren (1) auch (ρ, σ) -MSV genannt.

a) 4-stufiges Adams-Bashforth(4)-Verfahren

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}), \quad n = 3, 4, \dots$$

b) 3-stufiges Adams-Moulton(4)-Verfahren

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}), \quad n = 2, 3, \dots$$

6. Gegeben sind die linearen MSV

4-stufiges explizites Milne-Verfahren 4. Ordnung

$$y_{n+1} = y_{n-3} + \frac{h}{3}(8f_n - 4f_{n-1} + 8f_{n-2}), \quad n = 3, 4, \dots, \quad (5.4)$$

2-stufiges implizites Adams-Störmer-Verfahren 4. Ordnung (Milne-Simpson(4), Simpson-Regel)

$$y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}), \quad n = 1, 2, \dots \quad (5.5)$$

a) Bestimme die erzeugenden Polynome $\rho(\xi)$, $\sigma(\xi)$.

b) Bestimme die Konsistenzordnungen.

c) Bestimme das charakteristische Polynom $\phi(\xi; z)$ und die Konsistenzordnung des aus (5.4) und (5.5) bestehenden PC-Verfahrens von Milne.d) Überprüfe das **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 $y' = f(x, y)$

$$y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}), \quad n = 2, 3, \dots,$$

ist ein Differenzenschemaverfahren, das auf der Simpson-Regel der numerischen Integration beruht. Es gilt

$$\begin{aligned}
 y_{n+1} - y_{n-1} &= \int_{x_{n-1}}^{x_{n+1}} y' dx = \int_{x_{n-1}}^{x_{n+1}} f(x, y(x)) dx \\
 &= \frac{2h}{6}(f(x_{n-1}, y_{n-1}) + 4f(x_n, y_n) + f(x_{n+1}, y_{n+1})) \\
 &= \frac{h}{3}(6f_n + f_{n+1} - 2f_n + f_{n-1}) \\
 &= h(2f_n + \frac{1}{3}\nabla^2 f_{n+1}), \\
 y_{n+1} &= y_{n-1} + h(2f_n + \frac{1}{3}\nabla^2 f_{n+1}),
 \end{aligned}$$

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 $\nabla^3 f_{n+1} = \nabla^3 f_n = const$ 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

$$y_{n+1} = y_{n-1} + h(2f_n + \frac{1}{3}(\nabla^2 f_n + \nabla^3 f_n)) = y_{n-1} + \frac{h}{3}(8f_n - 5f_{n-1} + 4f_{n-2} - f_{n-3})$$

(explizites Nyström(4)), für das in einer Anlaufrechnung mittels RKV die Anfangswerte y_0, y_1, y_2, y_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	y_k	f_k	∇f_k	$\nabla^2 f_k$	$\nabla^3 f_k$
$n-3$			
$n-2$		
$n-1$	
n
$n+1$

Löse das AWP $y' = x - y$, $y(0) = 1$, $x \in [0, 1]$, $h = 0.1$, exakte Lösung ist $y(x) = 2e^{-x} + x - 1$, $y(1) = 2/e = 0.735\ 758\ 882\dots$, mit folgendem Ausgangstableau und vervollständige es (Rechnung mit 4 Nachkommastellen).

n	x_n	y_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					

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.

$$y_{n+1}^{(0)} = y_{n-1} + h \left(2f_n + \frac{1}{3}(\nabla^2 f_n + \nabla^3 f_n) \right) \quad (\text{Nyström(4)}),$$

$$y_{n+1}^{(k+1)} = y_{n-1} + h \left(2f_n + \frac{1}{3}\nabla^2 f_{n+1}^{(k)} \right), \quad k = 0, 1, \dots, \quad (\text{Milne-Simpson(4)}).$$

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	y_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	$y_4^{(0)}$ $y_4^{(1)}$ $y_4^{(2)}$ $y_4^{(3)}$	0.740646			
5	0.5	$y_5^{(0)}$ $y_5^{(1)}$ $y_5^{(2)}$ $y_5^{(3)}$				

9. Löse das AWP (lineare homogene DGI 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))

$$y_{n+1} = y_{n-1} + h(2f_n + \frac{1}{3}(\nabla^2 f_n + \nabla^3 f_n)) = y_{n-1} + \frac{h}{3}(8f_n + -5f_{n-1} + 4f_{n-2} - f_{n-3}).$$

a) Bestimme die exakte Lösung des AWP mittels des Ansatzes $y(x) = ce^{\lambda x}$.

b) Überführe die DGI in ein System 1. Ordnung

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

$$z' = g(x, y, z), \quad z(0) = z_0.$$

c) Finde die Darstellungen mit den Inkrementfunktionen Φ_i .

$$y_{n+1} = y_{n-1} + h\Phi_1(y'_n, y'_{n-1}, y'_{n-2}, y'_{n-3}),$$

$$y'_{n+1} = y'_{n-1} + h\Phi_2(y_n, y_{n-1}, y_{n-2}, y_{n-3}, y'_n, y'_{n-1}, y'_{n-2}, y'_{n-3}).$$

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

Ergänze die Tabelle.

n	x_n	y_n	$y'_n = f_n$	∇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 $y_n^{(0)}$ noch nicht gibt.

Im iterativen Korrektor wird auf der rechten Seite zunächst $f(x_{n+1}, \bar{y}_{n+1}^{(0)})$ benutzt und in den ev. weiteren Iterationen dann $f(x_{n+1}, y_{n+1}^{(k)})$, $k > 0$.

a) Diskutiere die nachfolgenden modifizierten PC-Verfahren 4. Ordnung.

Aus welchen Prädiktor- und Korrektor-Formeln sind sie zusammengesetzt?

- Milne, $Err = -\frac{1}{90}h^4 f^{(5)}$,

$$y_{n+1}^{(0)} = y_{n-3} + \frac{4}{3}h(2f_n - f_{n-1} + 2f_{n-2}), \text{ Milne-Prädiktor,}$$

$$\bar{y}_{n+1}^{(0)} = y_{n+1}^{(0)} + \frac{28}{29}(y_n - y_n^{(0)}), \text{ 1. Modifikator (Evaluation),}$$

$$y_{n+1} \leftarrow y_{n+1}^{(k+1)} = y_{n-1} + \frac{1}{3}h(f(x_{n+1}, y_{n+1}^{(k)}) + 4f_n + f_{n-1}), \quad k = 0, 1, \dots, K, \text{ Korrektor,}$$

$$\bar{y}_{n+1}^{(1)} = y_{n+1}^{(K+1)} + \frac{1}{29}(y_{n+1}^{(0)} - y_{n+1}^{(K+1)}), \text{ 2. Modifikator (Evaluation).}$$

- Hamming, $Err = -\frac{1}{40}h^4 f^{(5)}$,

$$y_{n+1}^{(0)} = y_{n-3} + \frac{4}{3}h(2f_n - f_{n-1} + 2f_{n-2}),$$

$$\bar{y}_{n+1}^{(0)} = y_{n+1}^{(0)} + \frac{112}{121}(y_n - y_n^{(0)}),$$

$$y_{n+1} \leftarrow y_{n+1}^{(k+1)} = \frac{1}{8}(9y_n - y_{n-2}) + \frac{3}{8}h(f(x_{n+1}, y_{n+1}^{(k)}) + 2f_n - f_{n-1}), \quad k = 0, 1, \dots, K,$$

$$\bar{y}_{n+1}^{(1)} = y_{n+1}^{(K+1)} + \frac{9}{121}(y_{n+1}^{(0)} - y_{n+1}^{(K+1)}).$$

- Adams-Bashforth-Moulton,

$$y_{n+1}^{(0)} = y_n + \frac{1}{24}h(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}),$$

$$\bar{y}_{n+1}^{(0)} = y_{n+1}^{(0)} + \frac{251}{270}(y_n - y_n^{(0)}),$$

$$y_{n+1} \leftarrow y_{n+1}^{(k+1)} = y_n + \frac{1}{24}h(9f(x_{n+1}, y_{n+1}^{(k)}) + 19f_n - 5f_{n-1} + f_{n-2}), \quad k = 0, 1, \dots, K,$$

$$\bar{y}_{n+1}^{(1)} = y_{n+1}^{(K+1)} + \frac{19}{270}(y_{n+1}^{(0)} - y_{n+1}^{(K+1)}).$$

- Hermite, $Err = \frac{31}{720}h^4 f^{(5)}$,

$$y_{n+1}^{(0)} = y_n + \frac{1}{2}h(-f_n + 3f_{n-1}) + \frac{1}{12}h^2(17y_n'' + 7y_{n-1}''),$$

$$\bar{y}_{n+1}^{(0)} = y_{n+1}^{(0)} + \frac{31}{30}(y_n - y_n^{(0)}),$$

$$y_{n+1} \leftarrow y_{n+1}^{(k+1)} = y_n + \frac{1}{2}h(f(x_{n+1}, y_{n+1}^{(k)}) + f_n) + \frac{1}{12}h^2(-(y_{n+1}^{(k)})'' + y_n''), \quad k = 0, 1, \dots, K,$$

$$\bar{y}_{n+1}^{(1)} = y_{n+1}^{(K+1)} - \frac{1}{30}(y_{n+1}^{(0)} - y_{n+1}^{(K+1)}).$$

Dieses Verfahren braucht noch die zweite Ableitung

$$y'' = \frac{d f(x, y)}{dx} = f_x + f_y y' = f_x + f_y f,$$

die für die gegebene DGL 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) Löse mit den PC-Verfahren mit der Schrittweite $h = 0.1$ das AWP

$$y' = \frac{1}{1 + \tan^2(y)}, \quad y(0) = 0, \quad x \in [0, 20], \quad \text{Lösung } y(x) = \arctan(x).$$

Führe nur **eine** Korrektoriteration 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. Bestimme für das BDF(2)-Verfahren

$$y_{n+1} = \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2}{3}hf_{n+1}$$

die erzeugenden Polynome $\rho(\xi)$, $\sigma(\xi)$ und die Konsistenzordnung.

Ist das Verfahren konvergent?

Hinweis: Wurzelkriterium.

12. Die iterative Lösung des impliziten Adams-Störmer-Verfahren 4. Ordnung

(Milne-Simpson(4)) erfolgt gemäß

$$y_{n+1}^{(k)} = y_{n-1} + \frac{h}{3}(f_{n+1}^{(k-1)} + 4f_n + f_{n-1}), \quad k = 0, 1, \dots, \quad f_{n+1}^{(k-1)} = f(x_{n+1}, y_{n+1}^{(k-1)}), \quad y_{n+1}^{(0)}$$

gegeben.

Untersuche, unter welcher Bedingung an h das Verfahren konvergiert.

Vergleiche das Ergebnis mit der Forderung aus der Schrittkenzahl des RK-Verfahrens $\kappa = h|f_y| = 0.1 \dots 0.2$.

Hinweis: Schätze die Differenz $|y_{n+1}^{(k+1)} - y_{n+1}^{(k)}|$ unter Anwendung der Lipschitz-Bedingung ab und nutze die Kontraktionseigenschaft.

13. Eine vierte Möglichkeit der Notation von impliziten MSV ist die Form

$$y_{n+1} = \sum_{j=1}^r a_j y_{n+1-j} + h \sum_{j=0}^r b_j f(x_{n+1-j}, y_{n+1-j}), \quad n \geq r-1, \quad y_0, y_1, \dots, y_{r-1} \quad \text{gegeben,}$$

bzw.

$$y_{n+1} = hb_0 f(x_{n+1}, y_{n+1}) + \sum_{j=1}^r (a_j y_{n+1-j} + hb_j f_{n+1-j}). \tag{5.6}$$

Die Bestimmung von y_{n+1} erfolgt iterativ bei gegebenen $y_{n+1}^{(0)}$

$$y_{n+1}^{(k+1)} = hb_0 f(x_{n+1}, y_{n+1}^{(k)}) + C = g(y_{n+1}^{(k)}), \quad k = 0, 1, \dots$$

a) Gebe die Fixpunktgleichung an, die dieser Iteration zugrunde liegt.

b) Untersuche die Konvergenzbedingung für das allgemeine Iterationsverfahren

(Fixpunktiteration, Picard-Verfahren).

Hinweis: Prüfe $|g'(y)| < 1$.

- c) 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.

Überprüfe hier seine Konvergenzbedingung. Inwieweit hat die Wahl der Schrittweite darauf Einfluss?

- d) 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. Schreibe eine Funktion oder Prozedur für das klassische Adams-Bashforth(3)-Verfahren.

16. Ermittle die Koeffizienten α_i, β_i der Adams-Bashforth(k)-Verfahren für $k = 1(1)10$.

Hinweis: Nutze ein CAS.

Serie 7 - Stabilität

1. Sind folgende DGI bzw. DGI-Systeme steif?

Man wähle geeignete AB sowie Parameter und löse das AWP.

- a) $y'' + 2000y' + 10^6y = 0, \quad 0 \leq x \leq 1,$

$$\text{Lösung } y(x) = (c_1 + c_2x)e^{-1000x}.$$

- b) $y'' + 100001y' + 10^5y = \sin(x), \quad 0 \leq x \leq 1,$

$$\text{Lösung } y(x) = c_1e^{-x} + c_2e^{-100000x} - \frac{100001}{20000000002} \cos(x) + \frac{99999}{20000000002} \sin(x)$$

- c) $\ddot{x} + 100.1\dot{x} + 1000x = x^2, \quad x = x(t), \quad -100 \leq t \leq 100.$

$$\begin{aligned} \text{Lösung der homogenen DGI } x_{hom}(t) &= c_1e^{-t(1001-3\sqrt{66889})/20} + c_2e^{-t(1001+3\sqrt{66889})/20} \\ &= c_1e^{-11.25563830t} + c_2e^{-88.84436170t} \end{aligned}$$

- d) $\ddot{x} + 100.1\dot{x} + 1000x = -x^2, \quad x = x(t), \quad -1 \leq t \leq 1.$

$$\text{Lösung der homogenen DGI } x_{hom}(t) = c_1e^{-t(1001-3\sqrt{66889})/20} + c_2e^{-t(1001+3\sqrt{66889})/20}$$

- e) $\ddot{x} - 100.1\dot{x} + 1000x = t^3, \quad x = x(t), \quad -100 \leq t \leq 100.$

$$\begin{aligned} \text{Lösung } x(t) &= c_1e^{t(1001+3\sqrt{66889})/20} + c_2e^{t(1001-3\sqrt{66889})/20} \\ &\quad + \frac{2408409003}{50000000000000} + \frac{2706003}{50000000000}t + \frac{3003}{10000000}t^2 + \frac{1}{1000}t^3 \end{aligned}$$

$$f) y' = \begin{pmatrix} 1 & 1 \\ \omega^2 & 1 \end{pmatrix} y, \quad \omega > 0.$$

$$\text{Lösung } y_1(t) = c_1 e^{(1+\omega)t} + c_2 e^{(1-\omega)t} \\ y_2(t) = \omega(c_1 e^{(1+\omega)t} - c_2 e^{(1-\omega)t})$$

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

$$\text{Lösung } y_1(t) = e^{-t/200} [c_1 \sin(t \sqrt{39999}/200) + c_2 \cos(t \sqrt{39999}/200)] \\ y_2(t) = 1/200 e^{-t/200} [c_1(\sqrt{39999} \cos(t \sqrt{39999}/200) - \sin(t \sqrt{39999}/200)) \\ - c_2(\sqrt{39999} \sin(t \sqrt{39999}/200) + \cos(t \sqrt{39999}/200))]$$

$$h) y' = \begin{pmatrix} 0 & 1 \\ -1 & -\frac{1}{100} \end{pmatrix} y, \quad y_1(0) = 0, \quad y_2(0) = 1.$$

$$\text{Lösung } y_1(t) = 200/\sqrt{39999} e^{-t/200} \sin(t \sqrt{39999}/200) = 1.0000125 e^{-0.005t} \sin(0.9999875 t) \\ y_2(t) = e^{-t/200} [\cos(t \sqrt{39999}/200) - 1/\sqrt{39999} \sin(t \sqrt{39999}/200)] \\ = e^{-0.005t} [\cos(0.9999875 t) - 0.005000062501 \sin(0.9999875 t)]$$

$$i) y' = \begin{pmatrix} 0 & 1 \\ -1 & -k \end{pmatrix} y, \quad k \geq 0.$$

$$\text{Lösung } y_1(t) = c_1 e^{t(-k+\sqrt{k^2-4})/2} + c_2 e^{t(-k-\sqrt{k^2-4})/2} \\ y_2(t) = c_1 (-k+\sqrt{k^2-4})/2 e^{t(-k+\sqrt{k^2-4})/2} + c_2 (-k-\sqrt{k^2-4})/2 e^{t(-k-\sqrt{k^2-4})/2}$$

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

2. Bestimme mittels der Test-DGL $y' = \lambda y$, $\lambda \in \mathbb{C}$, die Stabilitätsfunktion

$\mu(H) = E(\lambda, h)$ und den Stabilitätsbereich $\mathcal{E}(H) = \{H = \lambda h \in \mathbb{C} : |E(\lambda, h)| \leq 1\}$ folgender Verfahren.

- Euler-Cauchy-Verfahren (Rechteckregel links, PZV),
- Heun-Verfahren (Sehnentrapezregel, HV, VPZV),
- Collatz-Verfahren (Tangententrapezregel, MPZV),
- (3,3)-Verfahren von Kutta,
- KRKV.

Folgere daraus, dass das charakteristische Polynom des (r, p) -RK-Verfahrens (r Stufenzahl, p Ordnung) die Form hat

$$Q(\xi; H) = \xi - \mu(H) = \xi - (1 + \sum_{i=1}^r H^i \varepsilon_i), \quad \varepsilon_i \text{ eindeutig bestimmte Konstanten.}$$

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.

$$\bar{y}_{n+1} = y_n + hf(x_n, y_n) = (1 - \lambda h)y_n \quad (\text{PZV}),$$

$$y_{n+1}^{(k+1)} = y_n + 0.5h[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})], \quad y_{n+1}^{(0)} = \bar{y}_{n+1}, \quad k = 0, 1, \dots (\text{HV}).$$

Für $k = 0$ erhalten wir die Verstärkungsfunktion (Stabilitätsfunktion)

$$E_0 = E(\lambda, h) = 1 - \lambda h + \frac{(\lambda h)^2}{2},$$

aus der sich die Stabilitätsbedingung $|\lambda h| < 2$ ergibt.

- a) Wird durch weitere Iterationen der Korrekturformel diese Stabilitätsforderung abgeschwächt?

Hinweis: Untersuche dazu die Verstärkungsfunktion

$$E_k(z) = E_k(\lambda, h) = \frac{1 - \frac{z}{2} - 2\left(-\frac{z}{2}\right)^{k+3}}{1 + \frac{z}{2}}, \quad z = \lambda h.$$

- b) Was ist zur Konvergenz des Iterationsverfahrens zu sagen?

4. Bestimme die Stabilitätsfunktion $\mu(H) = E(\lambda, h)$ und -bereich $\mathcal{E}(H)$

- a) der impliziten (1,2)-Gauß-Formel

$$y_{n+1} = y_n + hk_1, \quad k_1 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right),$$

- b) der impliziten (2,4)-Gauß-Formel

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2),$$

$$k_1 = f\left(x_n + \frac{3 - \sqrt{3}}{6}, y_n + h\left(\frac{1}{4}k_1 + \frac{3 - 2\sqrt{3}}{12}k_2\right)\right),$$

$$k_2 = f\left(x_n + \frac{3 + \sqrt{3}}{6}, y_n + h\left(\frac{3 + 2\sqrt{3}}{12}k_1 + \frac{1}{4}k_2\right)\right).$$

5. Die Lösung das AWP (lineare homogene DGl 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) Überführe die DGl in ein System 1. Ordnung

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

$$z' = g(x, y, z), \quad z(0) = z_0.$$

- b) Notiere das PZV mit Schrittweite h komponentenweise sowie in Matrixdarstellung

$$y_{n+1} = My_n, \quad y_n = (y_n, z_n)^T \text{ mit der Transformationsmatrix } M.$$

- c) Zeige, dass das PZV genau dann stabil ist, wenn gilt $h < 0.2$.

Hinweis: Für die angenommene diskrete Lösung der Form $u_n = c\rho^n$ heißt Stabilität $|\rho| < 1$.

- d) Berechne die Eigenwerte λ der Matrix M und leite daraus die Stabilität ab.
e) Rechne einige Schritte des PZV mit $h = 1/6$.

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) Bestimme die exakte Lösung.
b) Transformiere die Aufgabe auf das entkoppelte DGI-System

$$\dot{z}_i = \lambda_i z_i, \quad z_i(0) = z_{i0}, \quad i = 1, 2, 3.$$

- c) Ermittle eine Schrittweiteschranke $h_0 > 0$ für das PZV und für das KRKV, so dass bei allen Schrittweiten $h < h_0$ das jeweilige Verfahren absolut stabil ist.

7. Die sogenannten θ -Verfahren sind parametrische ESV der Form

$$y_{n+1} = y_n + h[(1 - \theta)f(x_n, y_n) + \theta f(x_{n+1}, y_{n+1})], \quad \theta \geq 0,$$

oder als One-leg-Verfahren

$$y_{n+1} = y_n + hf((1 - \theta)x_n + \theta x_{n+1}, (1 - \theta)y_n + \theta y_{n+1}), \quad 0 \leq \theta \leq 1.$$

Sei $f \in \mathcal{C}^2(\mathbb{R}^2)$.

- a) Bestimme die Stabilitätsfunktion $\mu(H) = E(\lambda, h)$ der Verfahren.
b) Das Verfahren heißt **A-stabil** (unbeschränkt absolut stabil), falls $\mathcal{E}(H) \supseteq \mathbb{C}^- = \{z | \operatorname{Re}(z) < 0\}$ bzw. $\mathcal{E}(H)$ enthält einen "genügend" großen unbeschränkten Teil von \mathbb{C}^- .

Für welche Werte θ sind die Verfahren A-stabil?

8. Das **linear-implizite** Euler-Verfahren für DGI-Systeme lautet

$$y_{n+1} = y_n + h \left[I - h \frac{\partial f}{\partial y}(x_n, y_n) \right]^{-1} f(x_n, y_n), \quad n = 0, 1, 2, \dots,$$

mit der Einheitsmatrix I .

Bestimme die Stabilitätsfunktion $\mu(H) = E(\lambda, h)$. Ist das Verfahren A-stabil?

9. Man betrachte das DGI-System $y' = f(x, y) = Ay$ 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 $A = (\partial f / \partial y)$, die allgemeine Lösung und diskutiere die Steifheit des DGI-Systems.
- Löse das AWP für $x \in [0, 1]$.
- Berechne mit dem PZV und $h = 0.1, 0.05, 0.02, 0.01$ eine Näherung. Welche Stabilitätsbedingung muss eingehalten werden?
- Wende das A-stabile implizite Euler-Verfahren an, indem das entstehende Gleichungssystem nach den Unbekannten $(y_{1,n+1}, y_{2,n+1})$ aufgelöst wird.
- Vergleiche die Ergebnisse.

10. Bestimme für die beiden impliziten MSV

$$\begin{aligned} y_{n+1} &= \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2}{3}hf_{n+1}, \\ y_{n+1} &= y_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}), \end{aligned}$$

- das charakteristische Polynom $Q(\xi; H)$,
- den Stabilitätsbereich \mathcal{E} .
- Welche Stabilitätseigenschaft besitzen diese Verfahren?

11. Für die Test-DGI $y' = -\lambda y$, $y(x_0) = y_0$, $\lambda > 0$, untersuche man das implizite MSV von Milne-Simpson

$$\begin{aligned} y_{n+1} &= y_{n-1} - \frac{\lambda h}{3}(y_{n+1} + 4y_n + y_{n-1}) \quad \text{bzw.} \\ (1+z)y_{n+1} + 4zy_n + (z-1)y_{n-1} &= 0, \quad z = \lambda h/3, \end{aligned}$$

auf seine Stabilität.

Benutze dazu die Ansatzfunktion $y_n = c\rho^n$ für die Differenzgleichung mit der Forderung $|\rho| < 1$.

Literatur

- [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 uravnenia*. 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] BRAUN, M.: *Differentialgleichungen und ihre Anwendungen*. Springer-Verlag Berlin 1979, 1991, 1994.
- [13] BRÄUNING, G.: *Gewöhnliche Differentialgleichungen*. Fachbuchverlag Leipzig 1964.
- [14] BURRAGE, K.: *Parallel and Sequential Methods for Ordinary Differential Equations*. Oxford University Press Oxford 1995.
- [15] BUTCHER, J.C.: *The numerical analysis of ordinary differential equations. Runge-Kutta and general linear methods*. John Wiley& Sons, New York 1987-
- [16] CHERKASOVA, M.P.: *Collected Problems in Numerical Methods*. Akademie-Verlag Berlin 1972.

- [17] COLLATZ, L.: *Funktionalanalysis und numerische Mathematik*. Springer-Verlag Berlin 1964.
- [18] COLLATZ, L.: *The Numerical Treatment of Differential Equations*. 3rd Edition. Springer-Verlag Berlin 1960.
- [19] DEKKER, K.: und J.G. VERWER: *Stability of Runge-Kutta methods for stiff nonlinear differential equations*. North-Holland Amsterdam 1984.
- [20] DEMAILLY, J.-P.: *Gewöhnliche Differentialgleichungen*. Theoretische und numerische Aspekte, Lehrbuch Angewandte Mathematik. Vieweg Verlag Braunschweig 1994.
- [21] DEMUTH, M. und B.-W. SCHULZE (Hrsg.): *Differential Equations, Asymptotic Analysis, and Mathematical Physics*. Reihe Mathematical Research. Wiley-VCH Berlin 1996.
- [22] DEUFLHARD, P. und F. BORNEMANN: *Numerische Mathematik*. 2. Integration gewöhnlicher Differentialgleichungen. Walter de Gruyter Berlin 1994.
- [23] DEUFLHARD, P. und H. HOHMANN: *Numerische Mathematik*. Walter de Gruyter Berlin 1991.
- [24] 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.
- [25] FAIRES, J.D. und R.L. BURDEN: *Numerische Methoden. Näherungsverfahren und ihre praktische Anwendung*. Spektrum Akademischer Verlag Heidelberg 1994.
- [26] FORSYTHE, J., M. MALCOLM and C. MOLER: *Computer Methods for Mathematical Computations*. Prentice Hall 1979.
- [27] FORTUNA, Z., B. MACUKOW und J. WĄSOWSKI: *Metody numeryczne*. Podręczniki akademickie eit. WNT Warszawa 1982.
- [28] FRIED, I.: *Numerical Solution of Differential Equations*. Computer Science an Applied Mathematics. Academic Press New York 1979.
- [29] GOLUB, G.H. und J.M. ORTEGA: *Wissenschaftliches Rechnen & Differentialgleichungen*. Heldermann Verlag Berlin 1995. Hrsg. der dt. Übersetzung: Grigorieff, R. D.
- [30] GOLUBITSKY, M. und M. DELLNITZ: *Linear algebra and differential equations using MATLAB*. Brooks/Cole Pub. Co Pacific Grove 1999.
- [31] GOERING, H., FELGENHAUER, A., LUBE, G., ROOS, H.-G. und L. TOBISKA: *Singularly perturbed differential equations*. Mathematical Research 13. Akademie-Verlag Berlin 1983.

- [32] GRADSTEIN, I.S. und I.M. RYSHIK: *Tablicy integralow, summ, rjadow i proiswedenij*. Izdatelstwo Nauka Moskwa 1971.
- [33] GRIGORIEFF, R.D.: *Numerik gewöhnlicher Differentialgleichungen*.
Bd. I : Einschrittverfahren. Teubner Stuttgart 1972.
Bd.II : Mehrschrittverfahren. Teubner Stuttgart 1977.
- [34] GUCKENHEIMER, J. und P. HOLMES: *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. Springer-Verlag New York 1983, 1986, 1990.
- [35] HAIRER, E., NORSETT, S.P. und G. WANNER: *Solving Ordinary Differential Equations*. Vol. 1. Nonstiff problems. Springer-Verlag Berlin 1987.
- [36] HAIRER, E. und G. WANNER: *Solving Ordinary Differential Equations*. Vol. 2. Stiff and differential-algebraic problems. Springer-Verlag Berlin 1991.
- [37] HÄMMERLIN G. und K.-H. HOFFMANN: *Numerische Mathematik*. Grundwissen Mathematik 7. Springer-Verlag Berlin 1991.
- [38] HANKE-BOURGEOIS, M.: *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Teubner Stuttgart 2002.
- [39] HEMKER, P.W.: *A numerical study of stiff two-point boundary problems*. Akademische Dissertation. Mathematisches Zentrum Amsterdam 1977.
- [40] 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.
- [41] ISAACSON, E. und H. B. KELLER: *Analyse numerischer Verfahren*. Edition Leipzig 1972.
- [42] JANKOWSKA, J. und M. JANKOWSKI: *Przegląd metod i algorytmów numerycznych*. Bd. 1. WNT Warszawa 1981.
- [43] KALITKIN, N.N.: *Tshislelnye metody*. Nauka Moskwa 1978.
- [44] KAMKE, E.: *Differentialgleichungen*. Lösungsmethoden und Lösungen I. B. G. Teubner Stuttgart 1977.
- [45] KANTOROWITSCH, L. W. und G. P. AKILOW: *Funktionalanalysis in normierten Räumen*. Akademie-Verlag Berlin 1978.
- [46] KIELBASIŃSKI, A. und H. SCHWETLICK: *Numerische lineare Algebra*. DVW Berlin 1988.
- [47] KINCAID, D. und W. CHENEY: *Numerical Analysis*. Brooks/Cole Publ. Comp. Pacific Grovw 1991.
- [48] KNOBLOCH, H.W. und F. KAPPEL: *Gewöhnliche Differentialgleichungen*. B. G. Teubner Stuttgart 1974.

- [49] KNOBLOCH, J. und J. STEIGENBERGER: *Gewöhnliche Differentialgleichungen*. Skript IfMath TU Ilmenau 1996.
- [50] KORSCH, H.J.: *Chaos: a program collection for the PC*. Springer-Verlag Berlin 1998.
- [51] 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.
- [52] KOSE, K., SCHRÖDER, R. und K. WIELICZEK: *Numerik sehen und verstehen*. Ein kombiniertes Lehr- und Arbeitsbuch mit Visualisierungssoftware. Vieweg Braunschweig 1992.
- [53] KÖCKLER, N.: *Numerische Algorithmen in Softwaresystemen: unter besonderer Berücksichtigung der NAG-Bibliothek*. B. G. Teubner Stuttgart 1990.
- [54] LORENZ, E.N.: *The essence of chaos*. Paperback University of Washington Press 1995.
- [55] MAESS, G.: *Vorlesungen über Numerische Mathematik*. Bd. I, II. Akademie-Verlag Berlin 1984, 1988.
- [56] MATWIEJEW, N.M.: *Zadania z równań różniczkowych zwyczajnych*. PWN Warszawa 1976, Übers. aus dem Russ.
- [57] MICHLIN, S.G. und C.L. SMOLICKI: *Metody przybliżone rozwiązywania równań różniczkowych i całkowych*. PWN Warszawa 1970, Übers. aus dem Russ.
- [58] MIRANKER, W.L.: *Numerical Methods for Stiff Equations and Singular Perturbation Problems*. Mathematics and Its Applications 5. D. Reidel Publishing Comp. Dordrecht 1981.
- [59] MOHR, R.: *Numerische Methoden in der Technik*. Ein Lehrbuch mit Matlab-Routinen. Vieweg Verlag Wiesbaden 1998.
- [60] NEUNDORF, W.: *Numerische Mathematik*. Vorlesungen, Übungen, Algorithmen und Programme. Shaker Aachen 2002.
- [61] NEUNDORF, W.: *Gewöhnliche Differentialgleichungen*. Beispiele, Modelle, Verfahren, Software. Preprint M11/98 IfMath TU Ilmenau 1998.
- [62] OEVEL, W.: *Einführung in die Numerische Mathematik*. Spektrum Akademischer Verlag Heidelberg 1996.
- [63] OPFER, G.: *Numerische Mathematik für Anfänger*. Vieweg Studium Grundkurs Mathematik Wiesbaden 1993, 3. überarbeitete und erw. Auflage 2001.
- [64] PARKER, T.S. und L.O. CHUA: *Practical Numerical Algorithms for Chaotic Systems*. Springer-Verlag New York Berlin 1989.
- [65] PHILIPPOW, E. und W. BÜNTIG: *Analyse nichtlinearer dynamischer Systeme der Elektrotechnik*. Carl Hanser Verlag München 1992.

- [66] PLATO, R.: *Numerische Mathematik kompakt*. Grundlagenwissen für Studium und Praxis. Vieweg Verlag Wiesbaden 2000.
- [67] QUARTERONI, A., R. SACCO und F. SALERI: *Numerische Mathematik 2*. Springer-Verlag New York Berlin 2002.
- [68] RALSTON, A.: *A First Course in Numerical Analysis*. McGraw-Hill New York 1965.
- [69] RICE, J.R.: *Numerical Methods, Software and Analysis*. 2nd Edition. Academic Press Inc. Boston 1993.
- [70] ROOS, H.-G. und H. SCHWETLICK: *Numerische Mathematik*. Das Grundwissen für jedermann. B. G. Teubner Stuttgart 1999.
- [71] ROSS, SHEPLEY L.: *Differential equations*. 3 Ed. John Wiley & Sons, Inc. New York 1984.
- [72] RUTISHAUSER, H.: *Vorlesungen über Numerische Mathematik*. Bd. 1,2. Birkhäuser Verlag Basel 1976.
- [73] SCHABACK, R. und H. WERNER: *Numerische Mathematik*. Springer-Verlag Berlin 1993.
- [74] SCHWARZ, H.R.: *Numerische Mathematik*. B. G. Teubner Stuttgart 1988.
- [75] SCHWARZ, H.R., H. RUTISHAUSER und E. STIEFEL: *Numerik symmetrischer Matrizen*. B. G. Teubner Stuttgart 1972.
- [76] SCHWETLICK, H. und H. KRETZSCHMAR: *Numerische Verfahren für Naturwissenschaftler und Ingenieure*. Fachbuchverlag Leipzig 1991.
- [77] SEYDEL, R.: *Practical bifurcation and stability analysis*. Springer-Verlag New York 1994.
- [78] SEYDEL, R.: *From equilibrium to chaos*. Elsevier New York 1988.
- [79] SHAMPINE, L.F. und M.K. GORDON: *Computerlösung gewöhnlicher Differentialgleichungen*. Vieweg & Sohn Braunschweig 1984.
- [80] SPÄTH, H.: *Numerik*. Vieweg Wiesbaden 1994.
- [81] STEEB, W.-H.: *Chaos and Fractals. Algorithms and Computation*. B.I. Wissenschaftsverlag Mannheim 1992.
- [82] STOER, J.: *Numerische Mathematik I*. Heidelberger Taschenbücher 105. Springer-Verlag Berlin 1993.
- [83] STOER, J. und R. BULIRSCH: *Einführung in die Numerische Mathematik II*. Heidelberger Taschenbücher 114. Springer-Verlag Berlin 1990.
- [84] STRAMPP, W. und V. GRANZHA: *Differentialgleichungen mit Mathematica*. Vieweg-Verlag Wiesbaden 1995.

- [85] STREHMEL, K. und R. WEINER: *Numerik gewöhnlicher Differentialgleichungen*. Teubner Studienbücher Mathematik. B. G. Teubner Stuttgart 1995.
- [86] STUART, A. und A. HUMPHRIES, A.: *Dynamical Systems and Numerical Analysis*. Cambridge Monographs on Applied and Computational Mathematics 2. Cambridge Univ. Press 1996.
- [87] STUMMEL, F. und K. HAINER: *Praktische Mathematik*. B. G. Teubner Stuttgart 1982.
- [88] THOMPSON, W.J.: *Non-Linearity and Chaos in Engineering Dynamics*. John Wiley & Sons 1994
- [89] 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.
- [90] TROGER, H. und A. STEINDL: *Nonlinear stability and bifurcation theory*. Springer-Verlag 1991.
- [91] WALTER, W.: *Gewöhnliche Differentialgleichungen*. Eine Einführung. Heidelberger Taschenbücher Bd. 110. Springer-Verlag Berlin 1972.
- [92] WENZEL, H.: *MINÖL 7/1,2*. Gewöhnliche Differentialgleichungen, Band 1, 2. B. G. Teubner Verlagsgesellschaft 1973.
- [93] WERNER, H. und H. ARNDT: *Gewöhnliche Differentialgleichungen*. Eine Einführung in Theorie und Praxis. Springer-Verlag Berlin 1986.
- [94] WERNER, W.: *Mathematik lernen mit Maple*. Bd. 1,2. Ein Lehr- und Arbeitsbuch für das Grundstudium. dpunkt Heidelberg 1996, 1998.
- [95] WESTERMANN, T.: *Mathematik für Ingenieure mit Maple*. Springer 1996.

Anschrift:

PD Dr. rer. nat. habil. Werner Neundorf
Technische Universität Ilmenau, Institut für Mathematik
PF 10 05 65
D - 98684 Ilmenau

E-mail : werner.neundorf@tu-ilmenau.de

Homepage : <http://www.tu-ilmenau.de/num/team/werner-neundorf/>