

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 06/06

Modifiziertes Gradientenverfahren

Werner Neundorf und Nico Prätör

April 2006

‡MSC (2000): 65F10, 65G50, 65-05, 65Y20

Zusammenfassung

Gegenstand der Betrachtungen sind zwei Abstiegsverfahren zum Lösen von linearen bzw. nichtlinearen Gleichungssystemen:

- das Gradientenverfahren (GV),
auch bekannt als Methode des steilsten Abstiegs,
- und ein modifiziertes Gradientenverfahren (MGV).

In beiden Fällen wird das Lösen des Gleichungssystems durch das Lösen einer Extremwertaufgabe ersetzt, indem man Funktionale definiert, die ihr Minimum im Lösungspunkt des Gleichungssystems annehmen. Hieraus lassen sich nun Iterationsverfahren und Algorithmen ableiten, die im Falle der zwei Verfahren mit dem Computeralgebrasystem Maple 9 programmiert wurden.

Die beiden Verfahren werden in kleindimensionierten Beispielen praktischen Tests unterzogen, Iterationsverläufe werden anschaulich illustriert sowie die Resultate anschließend verglichen und ausgewertet.

Inhaltsverzeichnis

1	Einleitung	1
2	Gradientenverfahren	2
2.1	Theorie	2
2.2	Beispiele	5
3	Modifiziertes Gradientenverfahren	23
3.1	Linearer Fall	23
3.1.1	Theorie	23
3.1.2	Beispiele	29
3.2	Nichtlinearer Fall	43
3.2.1	Theorie	43
3.2.2	Beispiele im skalaren Fall	48
3.2.3	Beispiele im \mathbb{R}^2	60
4	Zusammenfassung	74
	Literaturverzeichnis	75

Kapitel 1

Einleitung

Das Lösen von linearen Gleichungssystemen (LGS) und nichtlinearen Gleichungssystemen (NLGS) ist eine Aufgabe, die seit Jahrhunderten Naturwissenschaftler und insbesondere Mathematiker beschäftigt. Im vergangenen Jahrhundert ist es nun gelungen, diese Aufgabe Maschinen zu übertragen, wozu jedoch erst Algorithmen für Computerprogramme geschaffen werden mussten, was in einem Teilgebiet der Mathematik, der Numerik realisiert wurde.

Gegenstand der Betrachtungen in der vorliegenden Arbeit sollen nun zwei Verfahren aus dem Bereich der Abstiegsverfahren sein, die zum Lösen von LGS bzw. NLGS verwendet werden können:

- das Gradientenverfahren (GV),
auch bekannt als Methode des steilsten Abstiegs,
- und ein modifiziertes Gradientenverfahren (MGV).

In beiden Fällen wird das Lösen des Gleichungssystems durch das Lösen einer Extremwertaufgabe ersetzt, indem man Funktionale definiert, die ihr Minimum im Lösungspunkt des Gleichungssystems annehmen. Hieraus lassen sich nun Iterationsverfahren und Algorithmen ableiten, die im Falle der zwei Verfahren mit dem Computeralgebrasystem Maple 9 programmiert wurden.

Die beiden Verfahren werden in kleindimensionierten Beispielen praktischen Tests unterzogen, Iterationsverläufe werden anschaulich illustriert sowie die Resultate anschließend verglichen und ausgewertet.

Kapitel 2

Gradientenverfahren

2.1 Theorie

Wir betrachten ein lineares Gleichungssystem

$$Ax = b \quad \text{mit} \quad A = (a_{ij}) \in \mathbb{R}^{n \times n}, \quad b \in \mathbb{R}^n \quad \text{und} \quad x \in \mathbb{R}^n. \quad (2.1)$$

Des Weiteren sei A symmetrisch und positiv definit (spd). Damit ist A regulär. Die Lösung $x^* = A^{-1}b$ des LGS entspricht nun der Minimierung des Funktionals

$$Q(x) = \frac{1}{2}x^T Ax - x^T b, \quad (2.2)$$

da im Minimum von $Q(x)$ offensichtlich $\nabla Q(x) = Ax - b = 0$ gilt und A die Hesse-Matrix von $Q(x)$ ist. Es ist $Q(x^*) = -\frac{1}{2}x^{*T}b$.

Von einem Startpunkt $x^{(0)} \in \mathbb{R}^n$ aus betrachten wir eine Abstiegsgerade g gemäß

$$g = \{x^{(0)} + \alpha p(x^{(0)}) \mid \alpha \in \mathbb{R}\} \quad (2.3)$$

in Richtung des steilsten Abstiegs, nämlich des negativen Gradienten und zugleich Residuums

$$p(x) = -Q'(x) = -\nabla Q(x) = b - Ax = r(x). \quad (2.4)$$

Nun bestimmen wir das Minimum des Funktionals Q auf der Geraden g und wählen die Minimumstelle als nächste Iterierte $x^{(1)}$.

Das Minimum wird angenommen für die Schrittzahl

$$\alpha_{min} = \frac{r(x^{(0)})^T r(x^{(0)})}{r(x^{(0)})^T A r(x^{(0)})},$$

denn für

$$\begin{aligned} q(\alpha) &= Q(x + \alpha p) = \frac{1}{2}(x + \alpha p)^T A(x + \alpha p) - (x + \alpha p)^T b \\ &= \frac{1}{2}\alpha^2 p^T A p + \alpha(p^T A x - p^T b) + \left(\frac{1}{2}x^T A x - x^T b\right) \end{aligned}$$

und man findet α_{min} als die Nullstelle der Ableitung

$$q'(\alpha) = \alpha p^T A p + p^T (A x - b) = \alpha p^T A p - p^T (b - A x) = \alpha r^T A r - r^T r.$$

Damit ergibt sich $x^{(1)} = x^{(0)} + \alpha_0 r(x^{(0)})$ sowie die weiteren Iterierten als

$$x^{(k+1)} = x^{(k)} + \alpha_k r(x^{(k)}) \quad \text{mit} \quad \alpha_k = \frac{r(x^{(k)})^T r(x^{(k)})}{r(x^{(k)})^T A r(x^{(k)})}. \quad (2.5)$$

Als einfaches Abbruchkriterium für den konvergenten, aber allgemein unendlichen Iterationsprozess können bei gegebener Toleranz $\varepsilon > 0$ und maximaler Iterationszahl *maxiter* die Bedingungen $k \geq \text{maxiter}$ und $\|r(x^{(k)})\|^2 < \varepsilon$ dienen.

Zahlreiche Informationen und Eigenschaften des GV findet der Leser in [3], [4], [6].

Die Formeln des GV lassen sich nun als Prozedur in Maple praktisch umsetzen.

```
> gv:=proc(n::posint,A::matrix,b::vector,x0::vector,
           maxiter::posint,etol::numeric,aus::name)

  local k,i,m,v,r,w,d,gamma0,gamman,x,alpha,Q,fh,fh1,fh2,xv1,rv1,xx;
  global xv,rv,lr;

  x:=evalm(x0):
  v:=evalm(A&*x):
  r:=evalm(b-v):  lr:=evalm(r):
  gamma0:=evalm(transpose(r)&*r):
  Q:=0.5*evalm(transpose(x)&*A&*x)-evalm(transpose(x)&*b);

  fh2:='%+.16e'; # Ausgabeformate einstellen
  fh1:='%+.16e';
  fh :=fh1;
  for m from 2 to n do, fh:=cat(fh,' ',fh1); end do;

  xx:=matrix(n,0,[]):
  xv1:=concat(xx,x);
  xv:=evalm(xv1);
  rv1:=concat(xx,r);
  rv:=evalm(rv1);
```

```

if aus=ja then
  printf('\n'):
  printf('Startvektor          x = [||fh||']\n',seq(x[i],i=1..n));
  printf('Residuum           r = b-Ax = [||fh||']\n',seq(r[i],i=1..n));
  printf('Funktionswert      Q(x) = '||fh1||'\n',Q);
  printf('Anfangsfehlerquadrat r'r = '||fh2||'\n\n',gamma0);
end if;

if gamma0<etol then RETURN(x,0); end if;
gamman:=gamma0;
for k from 1 by 1 to maxiter do
  w:=evalm(A&*r);
  d:=evalm(transpose(r)&*w);
  if d=0 then
    lprint('Abbruch wegen Nenner r'Ar=0'):
    RETURN(x,k-1);
  end if;

  alpha:=gamman/d;
  x:=evalm(x+alpha*r);
  v:=evalm(v+alpha*w);
  r:=evalm(b-v);  lr:=evalm(r);
  gamman:=evalm(transpose(r)&*r);
  Q:=0.5*evalm(transpose(x)&*A&*x)-evalm(transpose(x)&*b);
  xv1:=concat(xv,x);
  xv:=evalm(xv1);
  rv1:=concat(rv,r);
  rv:=evalm(rv1);

  if aus=ja then
    printf('Schritt k = %g\n',k);
    printf('Suchrichtung      p=r=b-Ax = [||fh||']\n',seq(r[i],i=1..n));
    printf('Suchschritt        alpha = '||fh1||'\n',alpha);
    printf('Iterationsvektor     x = [||fh||']\n\n',seq(x[i],i=1..n));
    printf('Residuum           r = b-Ax = [||fh||']\n',seq(r[i],i=1..n));
    printf('Funktionswert      Q(x) = '||fh1||'\n',Q);
    printf('Fehlernormquadrat   r'r = '||fh2||'\n\n',gamman);
  end if;

  if gamman<etol then printf('Abbruch, da Norm des Residuums < Epsilon');
                    break; end if;

end do;
if k>maxiter then k:=k-1; end if;
[x,k];
end:

```

2.2 Beispiele

Im Folgenden betrachten wir Beispiele zum GV zu (2.1) unter der üblichen Voraussetzung $A = A^T > 0$ sowie auch anderen Voraussetzungen über die Systemmatrix.

Beispiel 2.1 Sei

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 3/2 \\ 1 \end{pmatrix}.$$

A ist spd.

Die Lösung des Problems ist die Minimumstelle $x^* = \frac{1}{5}(1, 3)^T$ mit $Q(x^*) = -\frac{7}{10}$.

```
> n:=2:
  A:=matrix(n,n,[[2,1],[1,3]]):
  b:=vector(n,[1,2]):
  xs:=linsolve(A,b);
```

$$xs := \left[\frac{1}{5}, \frac{3}{5} \right]$$

Zunächst berechnen wir in Maple zehn Iterierte mit ausführlicher Ausgabe.

```
> Digits:=20:
  mmax:=10:
  x0:=vector(n,[1.5,1.0]):
  epsi:=1E-16:
  erg:=gv(n,A,b,x0,mmax,epsi,ja);
  evalm(erg[1]);
  erg[2];
  evalm(xv);
  cd:=coldim(xv);

Startvektor      x = [+1.5000000000000000e+00 +1.0000000000000000e+00]
Residuum         r = b-Ax = [-3.0000000000000000e+00 -2.5000000000000000e+00]
Funktionswert   Q(x) = +1.7500000000000000e+00
Anfangsfehlerquadrat r'r = +1.5250000000000000e+01

Schritt k = 1
Suchrichtung    p=r=b-Ax = [-4.9516908212560386e-01 +5.9420289855072464e-01]
Suchschritt     alpha = +2.9468599033816425e-01
Iterationsvektor x = [+6.1594202898550725e-01 +2.6328502415458937e-01]
Residuum        r = b-Ax = [-4.9516908212560386e-01 +5.9420289855072464e-01]
Funktionswert   Q(x) = -4.9698067632850242e-01
Fehlernormquadrat r'r = +5.9826950453919578e-01

Schritt k = 2
Suchrichtung    p=r=b-Ax = [-2.4859509020999704e-01 -2.0716257517499754e-01]
Suchschritt     alpha = +6.2244897959183673e-01
Iterationsvektor x = [+3.0772453909099872e-01 +6.3314601202799961e-01]
Residuum        r = b-Ax = [-2.4859509020999704e-01 -2.0716257517499754e-01]
Funktionswert   Q(x) = -6.8317679763917049e-01
Fehlernormquadrat r'r = +1.0471585142965307e-01
```

Schritt k = 3

Suchrichtung $p=r=b-Ax = [-4.1032200880071976e-02 \quad +4.9238641056086371e-02]$
 Suchschritt $\alpha = +2.9468599033816425e-01$
 Iterationsvektor $x = [+2.3446704873926046e-01 \quad +5.7209810340155106e-01]$
 Residuum $r = b-Ax = [-4.1032200880071976e-02 \quad +4.9238641056086371e-02]$
 Funktionswert $Q(x) = -6.9860594483049618e-01$
 Fehlernormquadrat $r'r = +4.1080852821126938e-03$

Schritt k = 4

Suchrichtung $p=r=b-Ax = [-2.0599839625505522e-02 \quad -1.7166533021254602e-02]$
 Suchschritt $\alpha = +6.2244897959183673e-01$
 Iterationsvektor $x = [+2.0892659717105239e-01 \quad +6.0274664528340074e-01]$
 Residuum $r = b-Ax = [-2.0599839625505522e-02 \quad -1.7166533021254602e-02]$
 Funktionswert $Q(x) = -6.9988448157645983e-01$
 Fehlernormquadrat $r'r = +7.1904324856637216e-04$

Schritt k = 5

Suchrichtung $p=r=b-Ax = [-3.4001345597654043e-03 \quad +4.0801614717184851e-03]$
 Suchschritt $\alpha = +2.9468599033816425e-01$
 Iterationsvektor $x = [+2.0285611303020294e-01 \quad +5.9768790849935953e-01]$
 Residuum $r = b-Ax = [-3.4001345597654043e-03 \quad +4.0801614717184851e-03]$
 Funktionswert $Q(x) = -6.9999042756235970e-01$
 Fehlernormquadrat $r'r = +2.8208632659807034e-05$

Schritt k = 6

Suchrichtung $p=r=b-Ax = [-1.7070063300046723e-03 \quad -1.4225052750038936e-03]$
 Suchschritt $\alpha = +6.2244897959183674e-01$
 Iterationsvektor $x = [+2.0073970274300202e-01 \quad +6.0022760084400062e-01]$
 Residuum $r = b-Ax = [-1.7070063300046723e-03 \quad -1.4225052750038936e-03]$
 Funktionswert $Q(x) = -6.9999920677966709e-01$
 Fehlernormquadrat $r'r = +4.9373918680899231e-06$

Schritt k = 7

Suchrichtung $p=r=b-Ax = [-2.8175225253700310e-04 \quad +3.3810270304440370e-04]$
 Suchschritt $\alpha = +2.9468599033816425e-01$
 Iterationsvektor $x = [+2.0023667189213108e-01 \quad +5.9980840846827484e-01]$
 Residuum $r = b-Ax = [-2.8175225253700310e-04 \quad +3.3810270304440370e-04]$
 Funktionswert $Q(x) = -6.9999993426977326e-01$
 Fehlernormquadrat $r'r = +1.9369776961560740e-07$

Schritt k = 8

Suchrichtung $p=r=b-Ax = [-1.4145113086551580e-04 \quad -1.1787594238792990e-04]$
 Suchschritt $\alpha = +6.2244897959183675e-01$
 Iterationsvektor $x = [+2.0006129549004172e-01 \quad +6.0001886015078207e-01]$
 Residuum $r = b-Ax = [-1.4145113086551580e-04 \quad -1.1787594238792990e-04]$
 Funktionswert $Q(x) = -6.9999999455326278e-01$
 Fehlernormquadrat $r'r = +3.3903160216975846e-08$

Schritt k = 9

Suchrichtung $p=r=b-Ax = [-2.3347408878768700e-05 \quad +2.8016890654522400e-05]$
 Suchschritt $\alpha = +2.9468599033816422e-01$
 Iterationsvektor $x = [+2.0001961182345817e-01 \quad +5.9998412376196244e-01]$
 Residuum $r = b-Ax = [-2.3347408878768700e-05 \quad +2.8016890654522400e-05]$
 Funktionswert $Q(x) = -6.9999999954865596e-01$
 Fehlernormquadrat $r'r = +1.3300476632998721e-09$

Schritt k = 10

Suchrichtung $p=r=b-Ax = [-1.1721352212606300e-05 \quad -9.7677935105053000e-06]$
 Suchschritt $\alpha = +6.2244897959183710e-01$
 Iterationsvektor $x = [+2.0000507925262546e-01 \quad +6.0000156284696168e-01]$
 Residuum $r = b-Ax = [-1.1721352212606300e-05 \quad -9.7677935105053000e-06]$
 Funktionswert $Q(x) = -6.999999996259936e-01$
 Fehlernormquadrat $r'r = +2.3279988775584006e-10$

Nun lassen wir die ausführliche Ausgabe beiseite und warten, bis die Prozedur von selbst stoppt. Dies geschieht entweder, wenn die Norm des Gradienten oder des Residuums kleiner als die vorgegebene Abbruchgenauigkeit $\varepsilon = \text{epsi}$ wird.

```
> Digits:=20:
  mmax:=100:
  x0:=vector(n, [1.5, 1.0]):
  epsi:=1E-16:
  erg:=gv(n,A,b,x0,mmax,epsi,nein);
  evalm(erg[1]);
  erg[2];
  evalm(xv);
  cd:=coldim(xv);
```

Abbruch, da Norm des Residuums < Epsilon

```
erg := [x, 16]
[ 0.20000000289010546117, 0.60000000088926321876]
16
[1.5, 0.61594202898550724637, 0.30772453909099871835, 0.23446704873926045950, 0.20892659717105239305,
0.20285611303020293959, 0.20073970274300202466, 0.20023667189213108258, 0.20006129549004172350,
0.20001961182345816571, 0.20000507925262546274, 0.20000162513434058842, 0.20000042089242154213,
0.20000013466680600068, 0.20000003487726316739, 0.20000001115916892874, 0.20000000289010546117]
[1.0, 0.26328502415458937198, 0.63314601202799960564, 0.57209810340155105661, 0.60274664528340073634,
0.59768790849935952512, 0.60022760084400062300, 0.59980840846827483793, 0.60001886015078206881,
0.59998412376196243730, 0.60000156284696168084, 0.59999868441505761889, 0.60000012950536047448,
0.59999989098401418993, 0.60000001073146558991, 0.59999999096638705769, 0.60000000088926321876]
cd := 17
```

Das GV konvergiert und bricht nach 16 Schritten ab.

Der Iterationsverlauf wird mit einem Maple-Plot grafisch veranschaulicht.

```
> evalm(A):
  z:=vector(n, [x1,x2]):
  Q:=evalm(0.5*transpose(z)*A&*z-transpose(z)*b):

> p2:=contourplot(Q,x1=0..2,x2=-0.5..1.5,scaling=constrained,contours=3,
  grid=[100,100],thickness=1,color=green):
p3:=contourplot(Q,x1=0..2,x2=-0.5..1.5,grid=[100,100],thickness=1,color=green,
  contours=[1.75,0.1,-0.497,-0.6832,-0.6986,-0.6999]):
p4:=plot([seq([xv[1,i],xv[2,i]],i=1..cd)],style=line,thickness=3,color=black):
p5:=pointplot([seq([xv[1,i],xv[2,i]],i=1..cd)],symbol=circle):
p7:=plot([[0,xv[2,1]], [xv[1,1],xv[2,1]], [xv[1,1],0]],style=line,
  linestyle=DASH,thickness=2,color=black):
p8:=textplot([[1.6,1.05,'x(0)'], [0.65,0.2,'x(1)'], [0.35,0.7,'x(2)'], [.1,.6,'x*']]):
plots[display]([p2,p3,p4,p5,p7,p8],title='GV - Iterationsverlauf, x(0)=[1.5,1]');

> # Speichern des Plots in einer Datei.
> pfad:='E:/Nico/Studium/Beleg/': dateiname:='gv_01a':
  file:=cat(pfad,dateiname,'.ps'):
  interface(plotdevice=ps,plotoutput=file,
    plotoptions='color,portrait,noborder,width=350,height=260');
plots[display]([p2,p3,p4,p5,p7,p8],title='GV - Iterationsverlauf, x(0)=[1.5,1]');
interface(plotdevice=win);
lprint('Creating ',file);
```

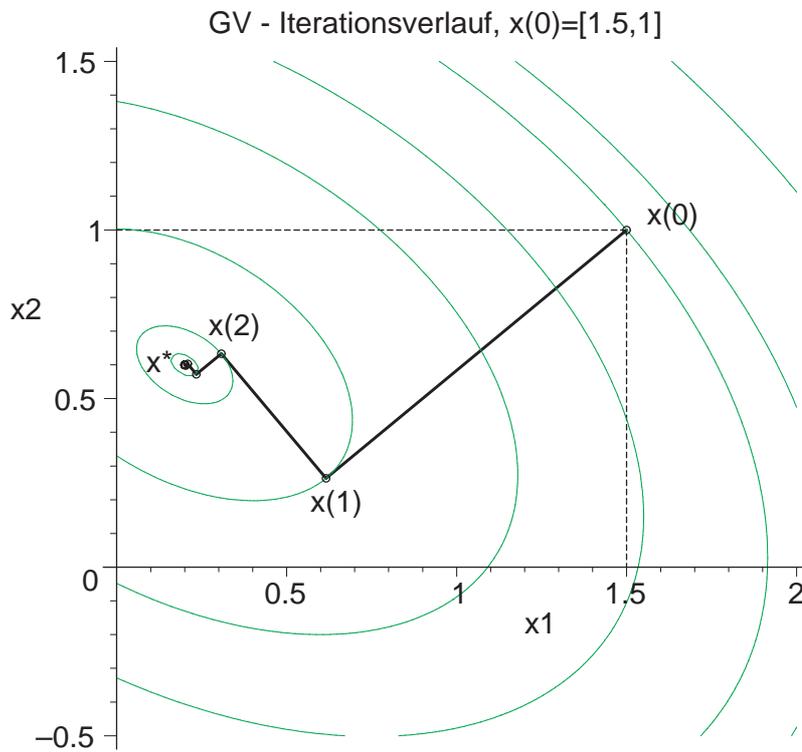


Abb. 2.1
Datei *gv_01a.ps*
Konvergenz
des GV

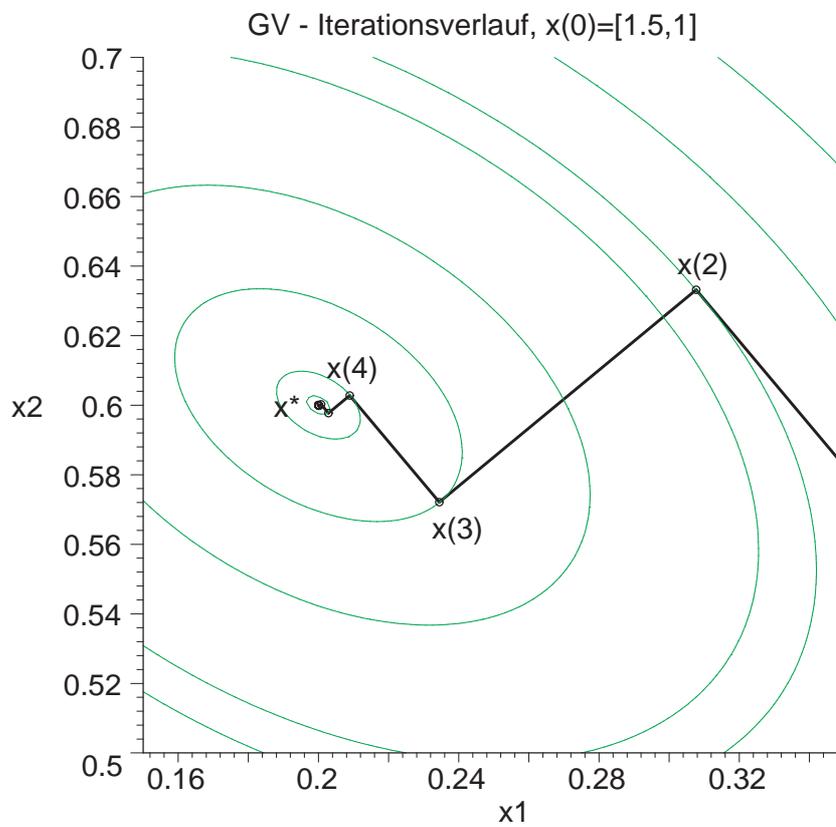


Abb. 2.2
Datei *gv_01b.ps*
Konvergenz
des GV
(Zoom) ◀

Beispiel 2.2 Sei

$$A = \begin{pmatrix} 2 & 1 \\ 1 & -3 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 3/2 \\ 1 \end{pmatrix}.$$

A ist symmetrisch und indefinit.

Die Lösung des Problems ist der Sattelpunkt $x^* = \frac{1}{7}(10, 1)^T$ mit $Q(x^*) = -\frac{31}{14}$.

Das GV konvergiert für den Startvektor. Aber es können auch Situationen auftreten, wo das GV divergiert bzw. wegen Nulldivision bei der Berechnung der Schrittzahl α abbricht.

Abbruch, da Norm des Residuums $<$ Epsilon

```

erg := [x, 22]
x := [1.42857142837454557514381, 0.142857140494546901725690]
xv := [1.5, 1.83333333333333333333333333333333, 1.41666666666666666666666667, 1.361111111111111111111112,
1.43055555555555555555555555555556, 1.43981481481481481481481482, 1.42824074074074074074074,
1.42669753086419753086420, 1.42862654320987654320988, 1.4288837448596707818930,
1.42856224279835390946502, 1.42851937585733882030178, 1.42857295953360768175583,
1.42858010402377686328304, 1.42857117341106538637403, 1.42856998266270385611949,
1.42857147109815576893766, 1.42857166955621602398008, 1.42857142148364070517705,
1.42857138840729732933665, 1.42857142975272654913716, 1.42857143526545044511056,
1.42857142837454557514381]
[1.0, 0.166666666666666666666666666666668, 0.3 · 10-23, 0.1388888888888888888888888890, 0.166666666666666666666667,
0.143518518518518518518519, 0.138888888888888888888888889, 0.142746913580246913580249,
0.143518518518518518518520, 0.142875514403292181069962, 0.142746913580246913580253,
0.142854080932784636488343, 0.142875514403292181069962, 0.142857653177869227251945,
0.142854080932784636488340, 0.142857057803688462124680, 0.142857653177869227251949,
0.142857157032718589645891, 0.142857057803688462124682, 0.142857140494546901725689,
0.142857157032718589645892, 0.142857143250908849712390, 0.142857140494546901725690]
    
```

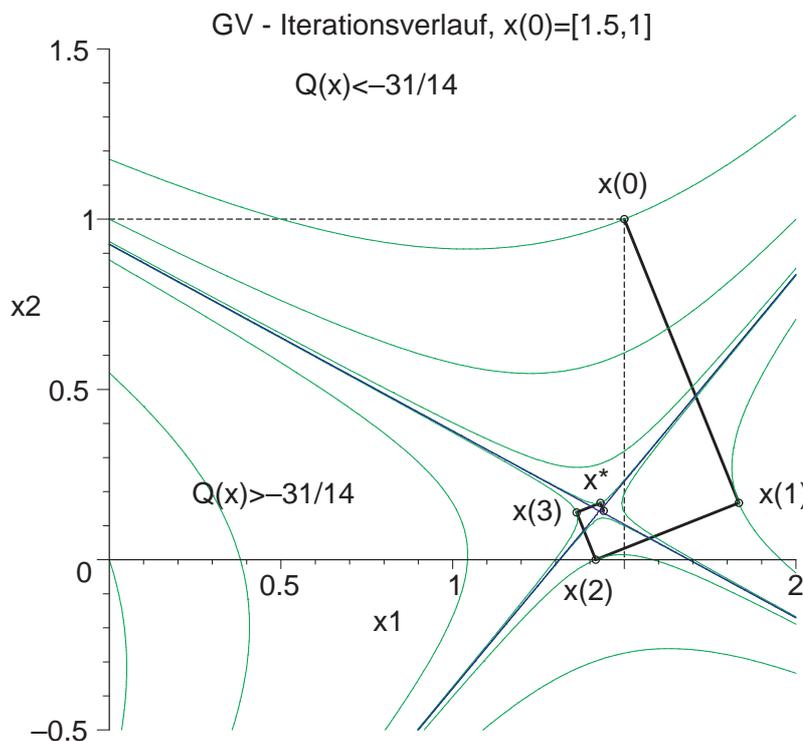


Abb. 2.3
 Datei *gv_02a.ps*
 Konvergenz
 des GV ◀

Beispiel 2.3 Sei

$$A = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

A ist positiv definit aber nicht symmetrisch.

Die Lösung des Problems ist $x^* = (1, 1)^T$ und keine Minimumstelle von $Q(x)$.

Es gilt $Q(x^*) = -3$. Das Funktional Q hat sein Minimum an der Stelle $z = \frac{1}{23}(30, 18)^T$ mit $Q(z) = -\frac{72}{23} = -3.130434$.

Das GV konvergiert jedoch gegen x^* .

Abbruch, da Norm des Residuums $<$ Epsilon

```

erg := [x, 13]
x := [1.00000000327048805351937, 0.999999999345902389296123]
xv := [1.0, 1.62500000000000000000, 1.00000000000000000000, 0.97395833333333333333,
1.00000000000000000000, 1.00108506944444444444, 0.99999999999999999999,
0.999954788773148148148142, 0.99999999999999999999, 1.0000188380111882716049,
0.99999999999999999999, 0.99999921508286715534973, 0.99999999999999999999,
1.00000000327048805351937]
[-1.0, 0.87500000000000000000, 1.08333333333333333333, 1.00520833333333333333,
0.99652777777777777777, 0.99978298611111111111, 1.00014467592592592592,
1.00000904224537037037036, 0.999993971836419753086410, 0.99999962323977623456789,
1.00000025117348251028806, 1.000000015698342656893000, 0.9999998953443822873799,
0.999999999345902389296123]

```

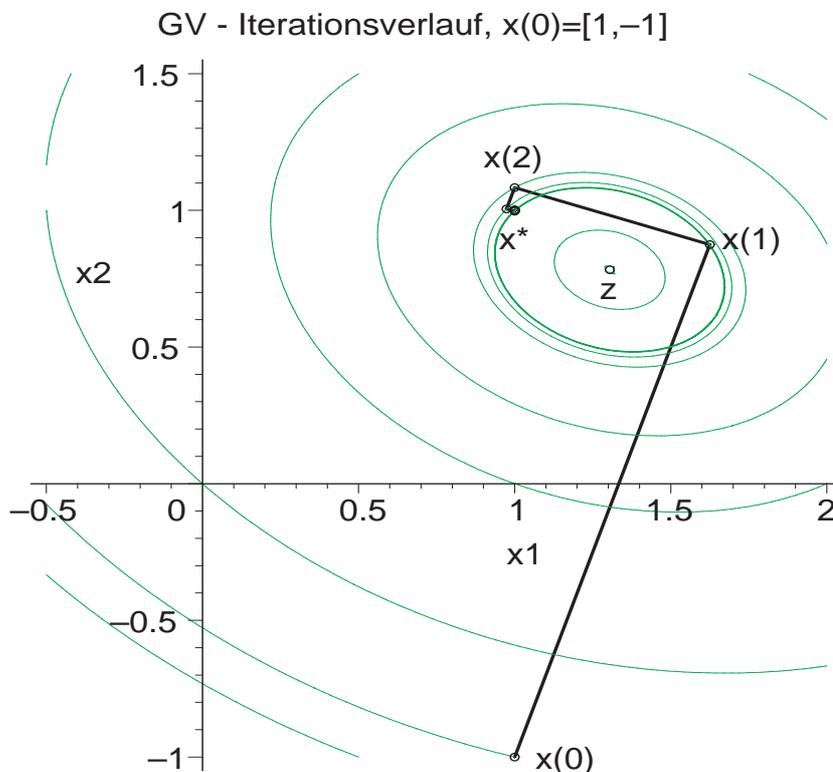


Abb. 2.4
Datei *gv_03a.ps*
Konvergenz
des GV

Die Situation ist etwas komplizierter und bedarf einiger Erläuterungen. Detaillierte Untersuchungen findet man in [4]. Hier sollen nur einige Hinweise gegeben werden.

Wenn $A > 0$ und nicht symmetrisch ist, kann man zwei Varianten einschlagen.

1. Der übliche Zugang für $A \neq A^T$ geht über Symmetrisierung des LGS und Betrachtung des Normalgleichungssystems (NGS)

$$A^T A x = A^T b, \quad B = A^T A = B^T > 0, \quad (2.6)$$

der Minimierung des Funktionals

$$R(x) = \frac{1}{2}(Ax)^T Ax - (Ax)^T b = \frac{1}{2}x^T A^T A x - x^T A^T b \quad (2.7)$$

und Anwendung des GV darauf. Aus der notwendigen Bedingung am Minimum $\nabla R(x) = 0$ leitet man die Lösung x^* des LGS und zugleich NGS ab. Das GV zu $R(x)$ konvergiert gegen x^* .

Der Gradient $\nabla R(x)$ ist senkrecht zur Höhenlinie $R(x) = \text{const}$ und liefert die Abstiegsrichtung

$$\hat{r}(x) = -\nabla R(x) = A^T(b - Ax) \perp R(x) = \text{const}. \quad (2.8)$$

2. Der zweite Weg benutzt

$$\tilde{A} = A + A^T.$$

Damit ist $\tilde{A} = \tilde{A}^T$ und positiv definit wegen $A > 0$.

Zum LGS $\frac{1}{2}\tilde{A}x = b$ mit seiner Lösung $z = \frac{1}{2}(\tilde{A})^{-1}b$, die im Allgemeinen verschieden von x^* ist, definiert man formal die Funktionale

$$\begin{aligned} T(x) &= \frac{1}{2}x^T \left(\frac{1}{2}\tilde{A} \right) x - x^T b, \\ S(x) &= \frac{1}{2}x^T \tilde{A}x - x^T b. \end{aligned}$$

Es gelten $T(x) = Q(x)$ und $S(x) = 2Q(x) + x^T b$.

Wenn man das GV auf $T(x)$ anwendet und dabei mit seiner Abstiegsrichtung $\tilde{r}(x) = b - \frac{1}{2}\tilde{A}x$ arbeitet, haben wir einen konvergenten Iterationsprozess zum Minimum an der Stelle z .

Das GV zu $Q(x)$, jedoch mit $r(x)$ als Suchrichtung liegt in der "Nachbarschaft" vom GV zu $R(x)$ mit $\hat{r}(x)$ als Abstiegsrichtung und kann damit zu x^* konvergieren.

Das Funktional $S(x)$ hat seine Minimumstelle bei $t = \tilde{A}^{-1}b = \frac{1}{2}z \neq x^*, z$. Es gelten jedoch $S(t) = \frac{1}{2}Q(z)$ und $S(0) = 0, S(x^*) = 0, S(z) = 0$, d. h. die Nullkontur von $S(x)$ geht durch die genannten drei Punkte.

Beispiel 2.4 Sei

$$A = \begin{pmatrix} 0.780 & 0.563 \\ 0.913 & 0.659 \end{pmatrix}, \quad b = \begin{pmatrix} 0.217 \\ 0.254 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 1.2 \\ -1.2 \end{pmatrix}.$$

A ist nicht symmetrisch und indefinit.

Die Lösung des Problems ist $x^* = (1, -1)^T$ und keine Minimumstelle von $Q(x)$. Es ist $Q(x^*) = \frac{37}{2000} = 0.0185$. Das Funktional Q ist eine Sattelfläche und hat seinen Sattelpunkt an der Stelle $z = \left(\frac{44449}{30624}, -\frac{6329}{5104}\right)^T$ mit $Q(z) = -\frac{37}{61248000} = -0.604 \cdot 10^{-6}$. Das GV divergiert (divergenter "Zick-Zack"-Iterationsverlauf).

```

erg := [x, 13]
x := [4606239.77364150165758084, -4606239.77364189438194648]
xv := [1.2, 1.169840174, 1.619228722, 1.525849569, 2.917221052, 2.628105783, 6.935991712, 6.040849318,
19.37868281, 16.60719341, 57.90304138, 49.32211217, 177.1799880, 150.6121989, 546.4785443,
464.2208536, 1689.879910, 1435.198287, 5230.014743, 4441.483862, 16190.78059, 13749.37574,
50126.88574, 42567.94571, 155197.9409, 131794.3770, 480513.0170, 408052.2222, 1487735.211,
1263386.185, 4606239.774 ]
[-1.2, -1.235302285, -1.619228722, -1.728529667, -2.917221052, -3.255632473, -6.935991712, -7.983762071,
-19.37868281, -22.62273032, -57.90304138, -67.94707833, -177.1799880, -208.2777689, -546.4785443,
-642.7617398, -1689.879910, -1987.986511, -5230.014743, -6152.995590, -16190.78059, -19048.46184,
-50126.88574, -58974.67729, -155197.9409, -182591.9742, -480513.0170, -565328.8781, -1487735.211,
-1750337.298, -4606239.774 ]

```

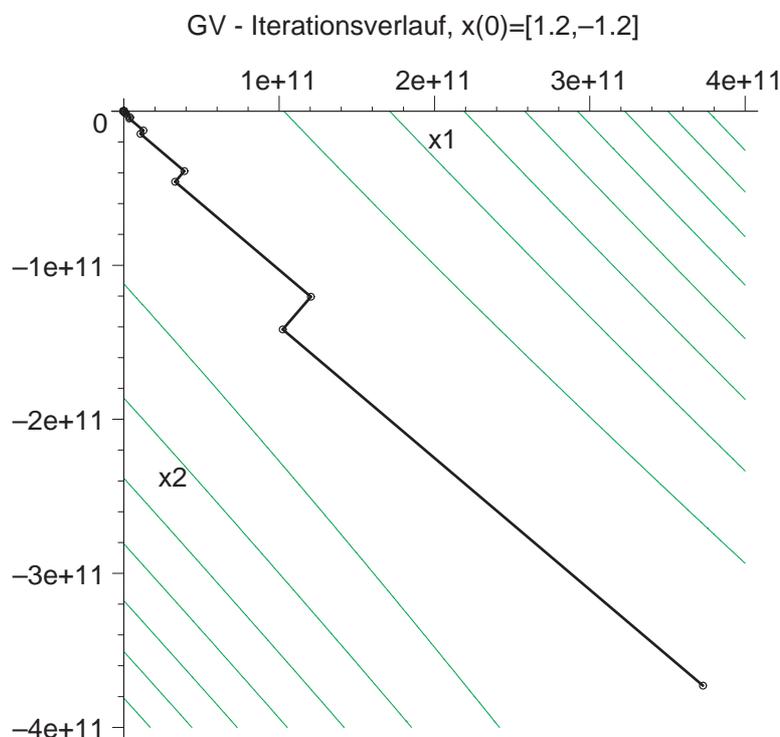


Abb. 2.6
Datei *gv_04.ps*
Divergenz
des GV

Wie in Beispiel 2.3. kann man die Funktionale $Q(x)$, $R(x)$ und $S(x)$ untersuchen.

Im Höhenlinienbild mit den drei Funktionalen treten einige Besonderheiten auf. Die Höhenlinie $S(x) = 0$ ist im Gegensatz zum Beispiel 2.3 keine Ellipse mehr. Die Niveaulinien $S(x) = S(t) = S(\frac{z}{2}) = -0.000\,000\,302\dots$ sind sich kreuzenden Kurven und $S(x) = 0$ zwei sich fast berührende Hyperbeläste. Sie sind lokal nur mit feiner Auflösung zu unterscheiden. Die Punkte 0 und x^* liegen auf dem linken unteren Ast, während der Punkt z auf dem rechten oberen sich befindet. Die Tallinie der sehr lang gestreckten Ellipsenschar zum Funktional $R(x)$ liegt nahe und fast parallel zu einem Zweig der Höhenlinien von $S(x) = S(t)$ und $R(t) \approx R(x^*)$. Mit dem anderen Zweig fällt ein Ast der Höhenlinie $Q(x) = Q(z) = -0.000\,000\,604\dots$ fast zusammen.

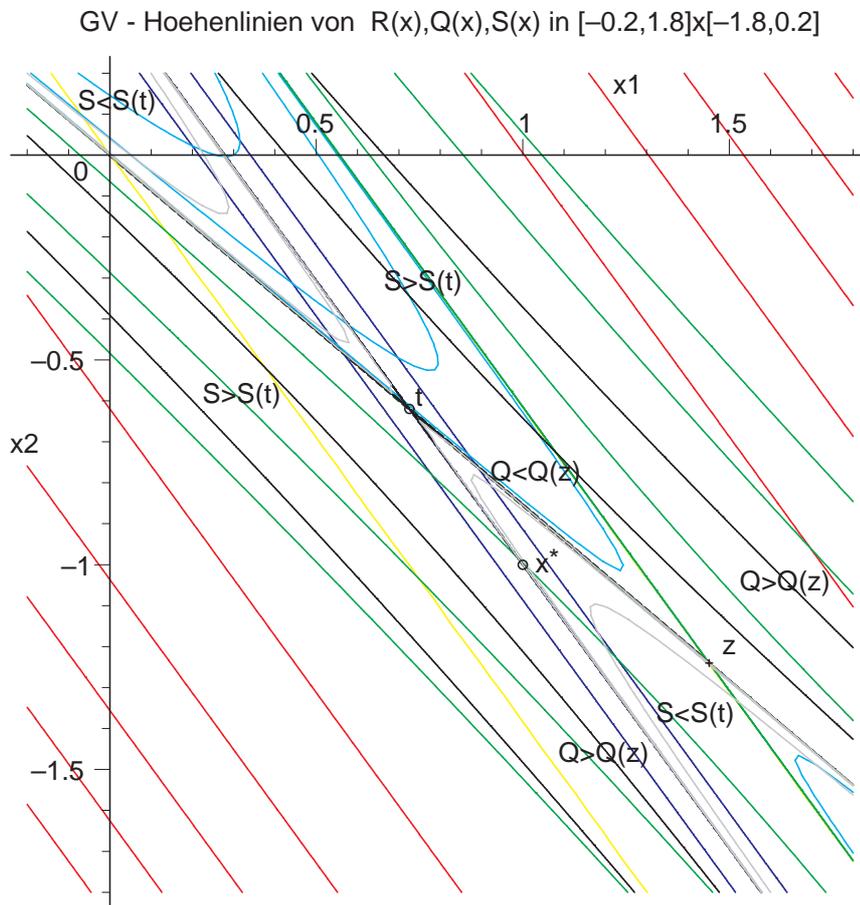


Abb. 2.7 Datei *gv_0040a.ps*

Höhenlinienbilder zu den Funktionalen

$Q(x)$ (Hyperbelschar), Sattel bei $z = (\frac{1}{2}\tilde{A})^{-1}b \neq x^*$,

contours=[0.2,0.1,0.01776,0,-0.00000060410,-0.001,-0.01,-0.03],

$R(x)$ (Ellipsenschar), Minimum bei $x^* = A^{-1}b$,

contours=[-0.0558025,-0.0558,-0.053,0] und contours=5,

$S(x)$ (Hyperbelschar), Sattel bei $t = \tilde{A}^{-1}b = \frac{z}{2}$,

contours=[0.2,0.05,0,-0.00000030205,-0.0000261,-0.001,-0.00888]

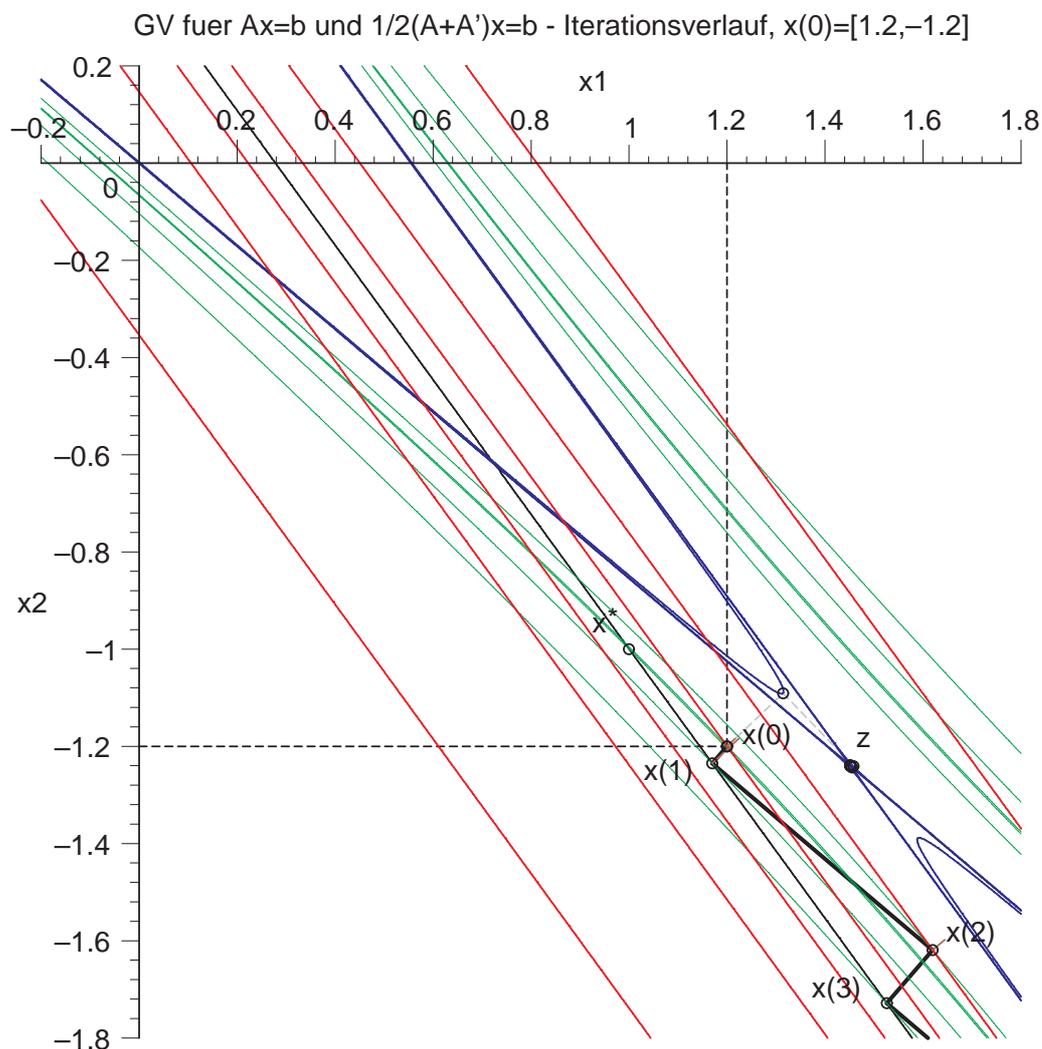


Abb. 2.8 Datei *gv_004a.ps*

Höhenlinienbild und GV zu $Q(x)$ mit den Iterierten $x^{(m)}$

(divergenter "Zick-Zack"- Iterationsverlauf fett gezeichnet),

$Q(x^*) = 0.0185$, $Q(z) = -0.000000604\dots$, Suchrichtung $r(x) = b - Ax$,

Vektor $\tilde{r}(x) \perp Q(x) = \text{const}$ in den Iterierten eingetragen,

Höhenlinien zu $Q(x)$ mit `contours`=[-0.0495,-0.000431,-0.000000604,
0.00000984,0.0114,0.01776,0.0185,0.02995,0.05397].

Zusätzlich eingezeichnet:

- Iterationsverlauf für $\frac{1}{2}\tilde{A}x = b$ (Funktional $Q(x) = T(x)$) mit Abstiegsrichtung $\tilde{r}(x)$, Konvergenz gegen z (dünn gestrichelt),
- Höhenlinien von $R(x)$ an den Iterierten, `contours`=[-0.0558025,-0.05357,-0.034405,0.14931],
- Funktional $S(x)$ mit Höhenlinie $S(x) = 0$.

Wichtiger Grund für die Divergenz des Verfahrens ist, dass die Matrix A nicht symmetrisch ist. Denn das GV ist grundsätzlich für spd Matrizen gedacht.

Abhilfe kann hier die Symmetrisierung des Systems $Ax = b$ zu NGS (2.6) schaffen.

In unserem Fall heißt das

$$B = A^T A = \begin{pmatrix} 1.441969 & 1.040807 \\ 1.040807 & 0.751250 \end{pmatrix} \quad \text{und} \quad c = A^T b = \begin{pmatrix} 0.401162 \\ 0.289557 \end{pmatrix}.$$

Bei einer Abbruchgenauigkeit von $\varepsilon = 10^{-32}$ und `Digits=24` braucht das GV dann nur drei Schritte bis es abbricht.

Abbruch, da Norm des Residuums < Epsilon

```

erg := [x, 3]
x := [1.00000000000175416320875, -1.00000000000243027666798]
xv := [1.2, 1.16341797148391625295701, 1.00000000000215594235871, 1.00000000000175416320875]
      [-1.2, -1.22640475027802150139476, -1.00000000000214027421180, -1.00000000000243027666798]

```

Die schlechte Kondition der Matrix $\text{cond}_2(A^T A) \approx 0.5 \cdot 10^{13}$ gestattet keine signifikante Verbesserung der nächsten Iterierten (nur Veränderungen in den letzten Dezimalstellen).

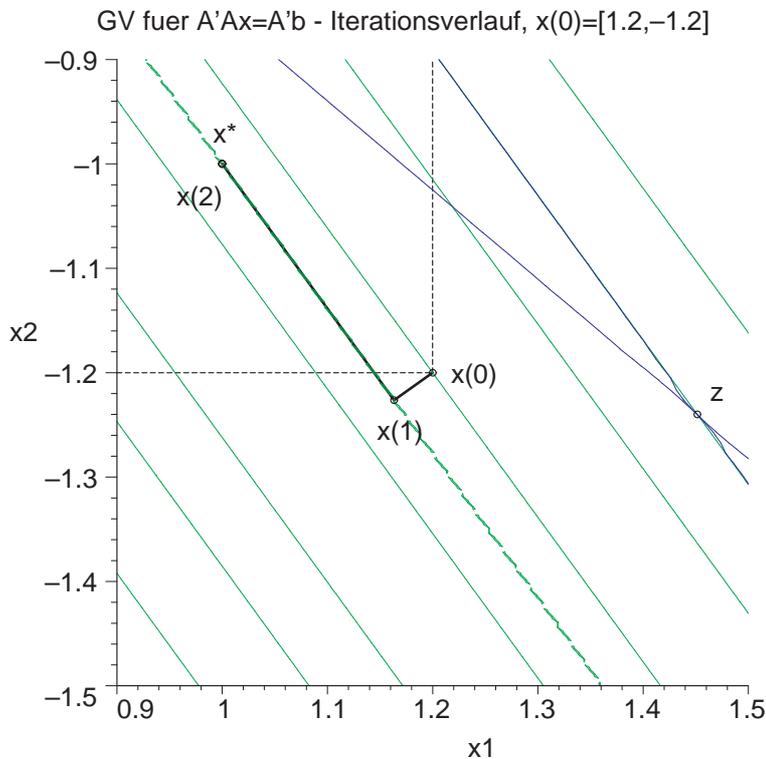


Abb. 2.9
 Datei `gv_04ca.ps`
 Konvergenz
 des GV
 zum NGS,
`Digits=24` ◀

Beispiel 2.5 Sei

$$A = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 6 \\ 2 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

A ist spd.

Die Lösung des Problems ist die Minimumstelle $x^* = (1, 2, 1)^T$ mit $Q(x^*) = -8$.

Das GV konvergiert.

Abbruch, da Norm des Residuums $<$ Epsilon

```

erg := [x, 14]
x := [0.999999998537093759409802, 1.99999999707418751881961, 0.999999998537093759409802]
xv := [0, 0.687500000000, 0.945312500000, 0.982910156250, 0.997009277344, 0.999065399170, 0.999836444855,
0.999948889017, 0.999991055578, 0.999997204868, 0.999999510852, 0.999999847141, 0.999999973250,
0.999999991641, 0.999999998537]
[0, 2.062500000000, 1.890625000000, 2.00341796875, 1.99401855469, 2.00018692017, 1.99967288971,
2.00001022220, 1.99998211116, 2.00000055903, 1.99999902170, 2.00000003057, 1.99999994650,
2.00000000167, 1.99999999707]
[0, 0.687500000000, 0.945312500000, 0.982910156250, 0.997009277344, 0.999065399170, 0.999836444855,
0.999948889017, 0.999991055578, 0.999997204868, 0.999999510852, 0.999999847141, 0.999999973250,
0.999999991641, 0.999999998537]

```

Plot des Iterationsverlaufes in der (x_1, x_2) -Ebene.

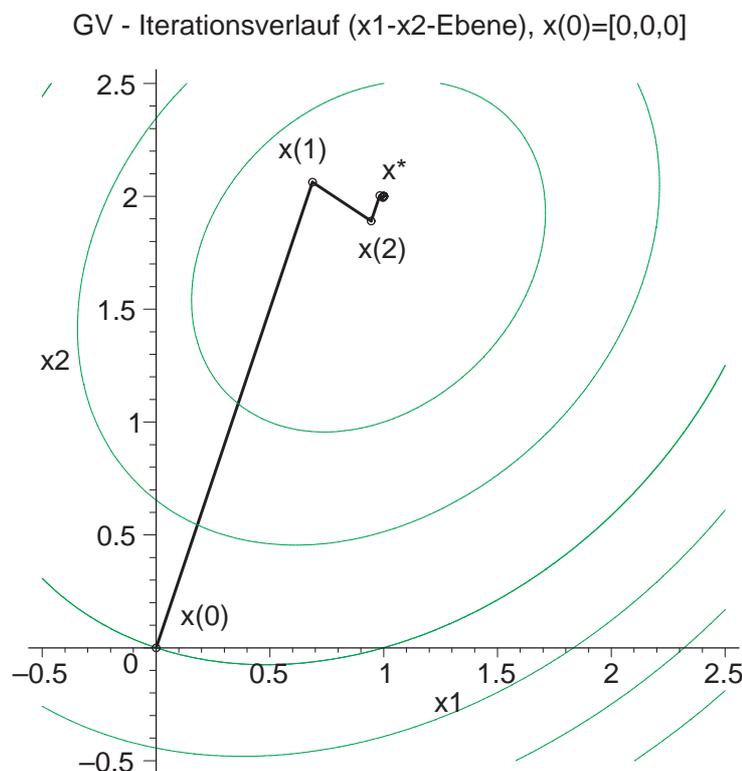


Abb. 2.10
Datei *gv_05a.ps*
Konvergenz
des GV ◀

Beispiel 2.6 Sei

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 9/2 \\ 3 \end{pmatrix}.$$

A ist diagonal und spd.

Die Lösung des Problems ist die Minimumstelle $x^* = (0, 0)^T$ mit $Q(x^*) = 0$.

Das GV konvergiert.

Abbruch, da Norm des Residuums $<$ Epsilon

```

erg := [x, 19]
x := [0.235204758972416374678303 · 10-8, -0.88201784614656085003810 · 10-9]
xv := [4.5, 1.75609756098, 0.464849354376, 0.181404626098, 0.480188716142 · 10-1,
0.187390718494 · 10-1, 0.496034254838 · 10-2, 0.193574343351 · 10-2,
0.512402673577 · 10-3, 0.199962018957 · 10-3, 0.529311226651 · 10-4,
0.206560478693 · 10-4, 0.546777737717 · 10-5, 0.213376678133 · 10-5,
0.564820618588 · 10-6, 0.220417802376 · 10-6, 0.583458888642 · 10-7,
0.227691273616 · 10-7, 0.602712194867 · 10-8, 0.235204758972 · 10-8]
[3.0, -0.658536585366, 0.309899569584, -0.680267347867 · 10-1, 0.320125810761 · 10-1,
-0.702715194353 · 10-2, 0.330689503225 · 10-2, -0.725903787567 · 10-3,
0.341601782385 · 10-3, -0.749857571088 · 10-4, 0.352874151100 · 10-4,
-0.774601795098 · 10-5, 0.364518491811 · 10-5, -0.800162543000 · 10-6,
0.376547079059 · 10-6, -0.826566758909 · 10-7, 0.388972592428 · 10-7,
-0.853842276061 · 10-8, 0.401808129911 · 10-8, -0.882017846147 · 10-9]

```

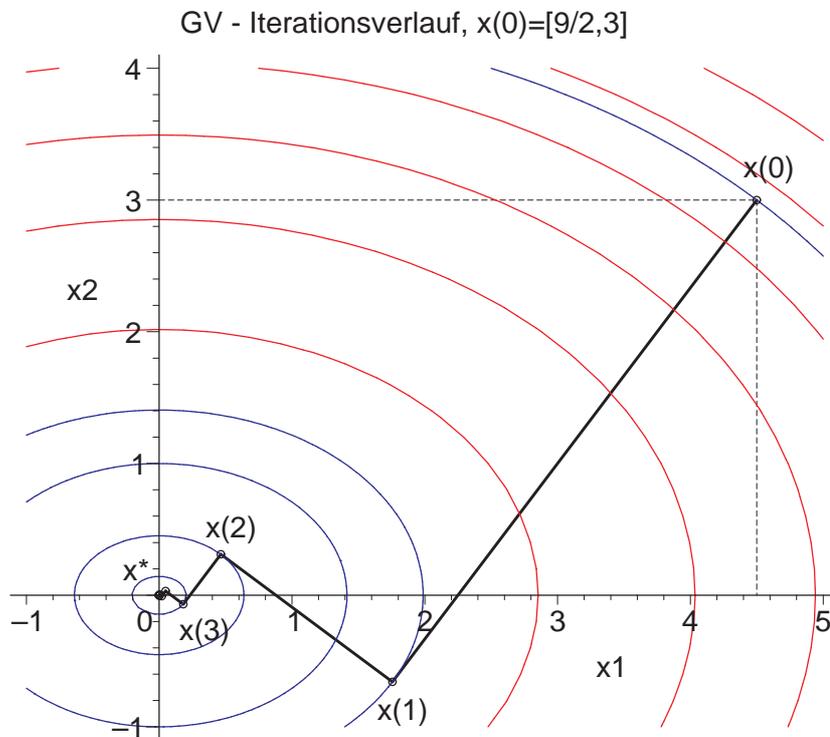


Abb. 2.11
Datei *gv_06a.ps*
Konvergenz
des GV ◀

Beispiel 2.7 Sei

$$A = \begin{pmatrix} 1 & 0 \\ 0 & -4 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

A ist diagonal, symmetrisch und indefinit.

Die Lösung des Problems ist $x^* = (0, 0)^T$ mit $Q(x^*) = 0$. x^* ist Sattelpunkt.

In Abhängigkeit vom Startvektor erhält man verschiedene Situationen.

Mit dem gegebenen Startvektor konvergiert das GV mit Abstiegs- und Anstiegsrichtungen gegen x^* .

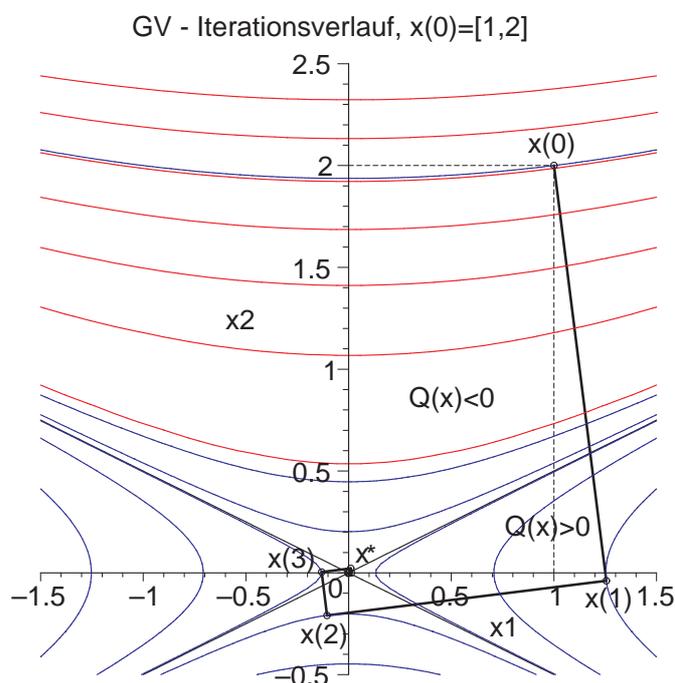


Abb. 2.12

Datei *gv_07aa.ps*

Höhenlinienbild

bei "einschachtelndem"

Iterationsverlauf des GV mit

$x^{(0)} = (1, 2)^T$

zu $Q(x) = \frac{1}{2}x_1^2 - 2x_2^2$, $Q(x^*) = 0$,

mit `contours=[-7.5,0.7843,`

`-0.0820,0.00858,0,-0.4,0.25]`

und `contours=7`

GV fuer $Ax=b$, A diagonal und indefinit - konvergenter Iterationsverlauf

m	[x(m) [1], x(m) [2]]	[r(m) [1], r(m) [2]]
0	[1.0000000000, 2.0000000000]	[-1.000e+00, 8.000e+00]
1	[1.2549019608, -0.0392156863]	[-1.255e+00, -1.569e-01]
2	[-0.1045751634, -0.2091503268]	[1.046e-01, -8.366e-01]
3	[-0.1312315776, 0.0041009868]	[1.312e-01, 1.640e-02]
4	[0.0109359648, 0.0218719296]	[-1.094e-02, 8.749e-02]
5	[0.0137235637, -0.0004288614]	[-1.372e-02, -1.715e-03]
6	[-0.0011436303, -0.0022872606]	[1.144e-03, -9.149e-03]
7	[-0.0014351439, 0.0000448482]	[1.435e-03, 1.794e-04]
8	[0.0001195953, 0.0002391907]	[-1.196e-04, 9.568e-04]
9	[0.0001500804, -0.0000046900]	[-1.501e-04, -1.876e-05]
10	[-0.0000125067, -0.0000250134]	[1.251e-05, -1.001e-04]

Mit den Startvektoren $x^{(0)} = (1, 0)^T$, $(0, 1)^T$ konvergiert das GV und braucht nur einen Schritt.

Mit dem Startvektor $x^{(0)} = (3, 1)^T$ divergiert das GV.

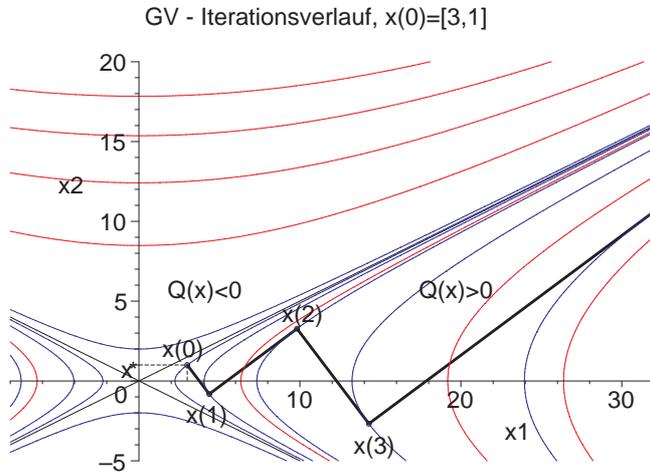


Abb. 2.13

Datei *gv_07ba.ps*

Höhenlinienbild

bei divergentem “Zick-Zack“-
Iterationsverlauf des GV mit

$x^{(0)} = (3, 1)^T$

zu $Q(x) = \frac{1}{2}x_1^2 - 2x_2^2$, $Q(x^*) = 0$,

mit `contours=[-8,0, 2.5,`

`8.18,26.78,87.63,286.8]`

und `contours=7`

Mit dem Startvektor $x^{(0)} = (2, 1)^T$ kommt es bei exakter Arithmetik nach dem ersten Schritt auf $x^{(1)} = \frac{1}{3}(8, -1)^T$ und dann zum Abbruch des GV, denn die weitere eindimensionale Minimumsuche schlägt im Zusammenhang mit der Nulldivision bei der Berechnung der Schrittzahl α fehl. Damit ist auch $x^{(0)} = (-8, 1)^T$ ein unbrauchbarer Startvektor und mit den “Nachbarn“ $(-8, 1.1)^T$, $(-8, 0.9)^T$ divergiert das GV.

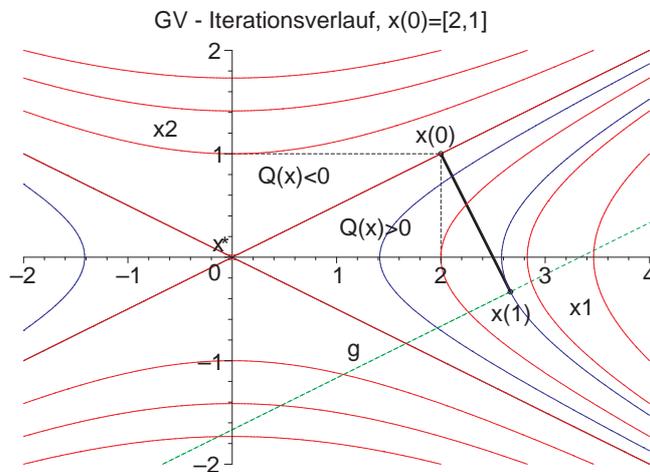


Abb. 2.14

Datei *gv_07ca.ps*

Höhenlinienbild

mit Abbruch bei $x^{(1)}$ des

Iterationsverlaufs des GV mit

$g(\alpha)$ und $x^{(0)} = (2, 1)^T$

zu $Q(x) = \frac{1}{2}x_1^2 - 2x_2^2$, $Q(x^*) = 0$,

mit `contours=[3.333,1,0]`

und `contours=7`

Rechnet man numerisch, wird die Nulldivision knapp umgangen, aber das GV divergiert.

```

xv := [2.0, 2.666666667, -0.4740740741 · 1024, -0.6320987654 · 1024, 0.2428407061 · 1047, 0.3237876081 · 1047,
-0.2121586231 · 1070, -0.2828781641 · 1070, 0.1010532431 · 1093, 0.1347376575 · 1093,
-0.3397304553 · 10115, -0.4529739404 · 10115, 0.6834090805 · 10137, 0.9112121074 · 10137,
-0.1167571375 · 10160, -0.1556761834 · 10160, 0.1684296354 · 10182, 0.2245728472 · 10182,
-0.2247197273 · 10204, -0.2996263031 · 10204, 0.2949476696 · 10226]

[1.0, -0.3333333333, -0.2370370370 · 1024, 0.7901234568 · 1023, 0.1214203531 · 1047, -0.4047345102 · 1046,
-0.1060793115 · 1070, 0.3535977051 · 1069, 0.5052662156 · 1092, -0.1684220719 · 1092,
-0.1698652277 · 10115, 0.5662174256 · 10114, 0.3417045403 · 10137, -0.1139015134 · 10137,
-0.5837856876 · 10159, 0.1945952292 · 10159, 0.8421481771 · 10181, -0.2807160590 · 10181,
-0.1123598637 · 10204, 0.3745328789 · 10203, 0.1474738348 · 10226] ◀

```

Beispiel 2.8 Sei

$$A = \begin{pmatrix} -1.0 & 0 & 2.1 & 0.4 \\ 0.2 & 2.3 & 0 & 1.0 \\ 2.0 & -0.4 & -5.0 & 0 \\ 0 & 1.0 & -1.0 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 1.0 \\ -1.0 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 1.5 \\ -1.0 \\ 0 \\ 1.0 \end{pmatrix}.$$

A ist nicht symmetrisch und indefinit.

Die exakte Lösung des Problems ist $x^* = \begin{pmatrix} 1.383\ 640\ 552\ 995 \\ -0.691\ 244\ 239\ 631 \\ 0.408\ 755\ 760\ 368 \\ 1.313\ 133\ 640\ 553 \end{pmatrix}$.

Das GV konvergiert nicht.

```

erg := [x, 50]
x := [-61.790643507, 133.354977557, 142.935006878, 83.272812560]
xv := [1.5, 1.268010994, 1.137166504, 1.385093154, 1.575298507, 1.272430270, 1.106294094, 1.437231129, 1.713257279,
1.285229248, 1.049642808, 1.531387627, 1.951526718, 1.305073736, 0.9369413119, 1.687938094, 2.347406969,
1.324219527, 0.7106801336, 1.934699106, 2.989576416, 1.322349573, 0.2658929062, 2.310685523, 4.019370400,
1.256623565, -0.5839833247, 2.873469819, 5.668652672, 1.045468085, -2.164652330, 3.712928554, 8.326249748,
0.5421182981, -5.037611486, 4.976354276, 12.65358742, -0.5085487463, -10.16238166, 6.913714739, 19.78496849,
-2.540690718, -19.16940777, 9.958226055, 31.67310053, -6.284072525, -34.82075584, 14.86815264, 51.68330157,
-12.95804890, -61.79064351]
[-1.0, -1.210899096, -0.2390863586, -0.0094012961, -1.084880319, -1.359883955, -0.0732232881, 0.2416283413,
-1.250421881, -1.639598079, 0.2070579120, 0.6743960962, -1.560894059, -2.151182620, 0.6795688529,
1.415279266, -2.124738933, -3.065051234, 1.474596576, 2.675870784, -3.123990193, -4.667646960,
2.812629637, 4.813511206, -4.864976141, -7.442303886, 5.070244617, 8.436710938, -7.865432561,
-12.20840632, 8.893896466, 14.58690756, -13.00330283, -20.35934336, 15.39499663, 25.04974181,
-21.77021768, -34.26811960, 26.48309814, 42.88644724, -36.70320942, -57.97834075, 45.43410784,
73.34000600, -62.12058785, -98.38197579, 77.85894521, 125.3801726, -105.3774251,
-167.2283032, 133.3549776]
[0, 0.5061578308, 0.8345665167, 0.2469940081, -0.09823827617, 0.5539736434, 0.9997110464, 0.2073898046,
-0.275555157, 0.6413503273, 1.292382323, 0.1373526507, -0.5947575413, 0.7942157958, 1.800417404,
0.01086499092, -1.163577960, 1.053800649, 2.668519351, -0.2183764160, -2.165809491, 1.486471203,
4.137378760, -0.6317468633, -3.915082627, 2.201157754, 6.612019049, -1.370952590, -6.947389801,
3.379816801, 10.78001011, -2.680974836, -12.17919312, 5.330136144, 17.81447370, -4.982789573,
-21.17653291, 8.576289525, 29.72145881, -8.996367398, -36.61259049, 14.01483585, 49.93350366,
-15.94893571, -63.04609167, 23.18210217, 84.32336091, -27.92829887, -108.2468816,
38.71218650, 142.9350069]
[1.0, 1.021089910, 1.418119428, 1.423642650, 0.8832862123, 0.9073238091, 1.439978667, 1.445687089,
0.7063115118, 0.7329665945, 1.517129104, 1.520241931, 0.4368590323, 0.4654588050, 1.702975808,
1.698534209, 0.02300415019, 0.05271278595, 2.088422807, 2.068354436, -0.6223104312, -0.5919452026,
2.825390557, 2.777282428, -1.649412530, -1.617266777, 4.166376803, 4.071716677, -3.321715452,
-3.282746283, 6.531236212, 6.362700075, -6.102472254, -6.043381532, 10.61993326, 10.33697104,
-10.80435923, -10.69590328, 17.60341976, 17.14479884, -18.84576747, -18.62951990, 29.44799825,
28.71891268, -32.68828830, -32.25428051, 49.46837825, 48.31789605,
-56.58202316, -55.73135463, 83.27281256]

```

Startvektor $x^{(0)} = (0, 0, 0, 0)^T$

$erg := [x, 50]$

$x := [-0.832447599 \cdot 10^{41}, 0.681237312 \cdot 10^{42}, 0.177324956 \cdot 10^{42}, 0.302138767 \cdot 10^{42}]$

Startvektor $x^{(0)} = (1, 1, 1, 1)^T$

$erg := [x, 50]$

$x := [-0.115393116 \cdot 10^{23}, 0.549528303 \cdot 10^{23}, 0.251671480 \cdot 10^{23}, 0.262810246 \cdot 10^{23}]$

Da hilft auch kein noch so guter Startvektor wie z. B. $x^{(0)} = (1.4, -0.7, 0.4, 1.3)^T$.

$$\text{erg} := [x, 50]$$

$$x := [-332.377065860, 758.766205212, 739.529758156, 433.315368356]$$

Grund für die Divergenz des Verfahrens ist, wie in Beispiel 2.4, dass die Matrix A nicht symmetrisch ist. Abhilfe kann hier die Symmetrisierung des Systems $Ax = b$ zu dem NGS schaffen.

In unserem Fall heißt das

$$B = A^T A = \begin{pmatrix} 5.04 & -0.34 & -12.1 & -0.2 \\ -0.34 & 6.45 & 1.0 & 2.30 \\ -12.1 & 1.0 & 30.41 & 0.84 \\ -0.2 & 2.30 & 0.84 & 1.61 \end{pmatrix} \quad \text{und} \quad c = A^T b = \begin{pmatrix} 2.0 \\ -1.5 \\ -3.9 \\ 0 \end{pmatrix}.$$

Nun konvergiert das GV sehr langsam. Nach 500 Iterationen ist der absolute Fehler von der Größenordnung $\mathcal{O}(10^{-5})$, nach 1000 Iterationen $\mathcal{O}(10^{-8})$. Wesentlich kleiner wird der Fehler nicht werden, denn in der dritten Komponente erkennt man schon die Stagnation.

$$\begin{aligned} x^{(1)} &= [1.334677048, -0.908347276, 0.417948024, 1.041765798] \\ x^{(5)} &= [1.354535863, -0.635061984, 0.400022951, 1.161631876] \\ x^{(10)} &= [1.366236489, -0.643877357, 0.400518509, 1.186051344] \\ x^{(20)} &= [1.379015961, -0.656385697, 0.406070581, 1.219270978] \\ x^{(50)} &= [1.394555264, -0.676525339, 0.412916429, 1.272969972] \\ x^{(100)} &= [1.394455385, -0.686831608, 0.413073676, 1.300778315] \\ x^{(250)} &= [1.385462192, -0.690842715, 0.409481960, 1.311973623] \\ x^{(500)} &= [1.383707721, -0.691229782, 0.408782529, 1.313091803] \\ x^{(600)} &= [1.383658450, -0.691240388, 0.408762893, 1.313122494] \\ x^{(700)} &= [1.383645322, -0.691243213, 0.408757661, 1.313130670] \\ x^{(800)} &= [1.383641824, -0.691243966, 0.408756267, 1.313132849] \\ x^{(900)} &= [1.383640892, -0.691244167, 0.408755895, 1.313133430] \\ x^{(1000)} &= [1.383640643, -0.691244220, 0.408755796, 1.313133584] \\ x^{(1010)} &= [1.383640632, -0.691244223, 0.408755792, 1.313133591] \\ x^{(1020)} &= [1.383640622, -0.691244225, 0.408755788, 1.313133597] \\ x^{(1030)} &= [1.383640614, -0.691244227, 0.408755785, 1.313133603] \\ x^{(1036)} &= [1.383640609, -0.691244228, 0.408755783, 1.313133606] \\ x^{(1037)} &= [1.383640609, -0.691244228, 0.408755783, 1.313133606] \\ x^{(1038)} &= [1.383640608, -0.691244228, 0.408755782, 1.313133607] \\ x^{(1039)} &= [1.383640607, -0.691244228, 0.408755783, 1.313133607] \\ x^* &= [1.383640553, -0.691244239, 0.408755760, 1.313133640] \quad \blacktriangleleft \end{aligned}$$

Kapitel 3

Modifiziertes Gradientenverfahren

3.1 Linearer Fall

3.1.1 Theorie

Die Idee zur Wahl von quadratischen Formen und ihre Einbindung in ein Optimierungsproblem für die Lösung von $Ax = b$ kann man ganz einfach im eindimensionalen (skalaren) Fall nachvollziehen.

Für die Nullstellenaufgabe

$$f(x) = ax + b = 0, \quad a \neq 0, \quad (3.1)$$

mit der Nullstelle $x^* = -\frac{b}{a}$ ergibt sich sofort die daraus abgeleitete und zu minimierende Funktion

$$h(x) = [f(x)]^2 \geq 0. \quad (3.2)$$

Sie hat mindestens dieselben Glattheitseigenschaften wie $f(x)$ und an der Stelle x^* ihr eindeutiges absolutes Minimum.

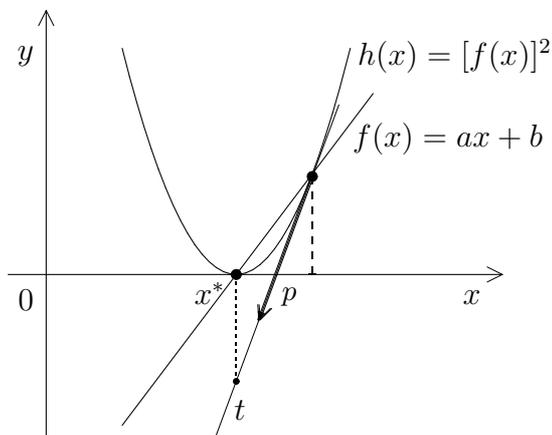


Abb. 3.1

Datei *abst1.pic*

Nullstellensituation $f(x) = 0$

und zugehörige Minimierungsaufgabe

$\min_x [f(x)]^2$ im \mathbb{R}^1

Man schreibt

$$\begin{aligned} 0 = h(x^*) &= \min_{x \in \mathbb{R}} h(x) \quad \text{und} \\ x^* &= \arg \min_{x \in \mathbb{R}} h(x). \end{aligned} \quad (3.3)$$

Bezüglich der Minimumstelle ändert sich nichts, wenn wir die Funktion $h(x)$ strecken und vertikal verschieben, also mit

$$H(x) = c_1 h(x) + c_2, \quad c_1 > 0,$$

arbeiten.

Wenn man entlang der Tangente t an die Kurve $h(x)$ in Richtung des steilsten Abstiegs (Vektor p) geht, "fährt" man sozusagen unter dem Minimum entlang und kann mittels einer Strahlenminimierung genau die Stelle darunter treffen.

Im mehrdimensionalen Fall werden im Allgemeinen mehrere aufeinander folgende Strahlenminimierungen in Richtung der Minimumstelle erforderlich sein.

Wir übertragen die Vorgehensweise auf den mehrdimensionalen Fall des LGS (2.1) mit der regulären Matrix

$$A = \begin{pmatrix} -a_1 & - \\ \vdots & \\ -a_n & - \end{pmatrix} \in \mathbb{R}^{n \times n} \quad \text{und} \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^n, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n. \quad (3.4)$$

a_i sind die Zeilen von A als Vektoren.

Die Lösung von (2.1) bezeichnen wir meistens als $x^* = A^{-1}b$.

Des weiteren definieren wir

$$\begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = \begin{pmatrix} a_1 x - b_1 \\ \vdots \\ a_n x - b_n \end{pmatrix} = Ax - b = -r(x) \quad (3.5)$$

und

$$h(x) = f_1^2(x) + f_2^2(x) + \dots + f_n^2(x) \geq 0. \quad (3.6)$$

Wie man leicht sieht, ist $h(x) = 0$ genau dann, wenn $Ax = b$.

Die Suche nach der Lösung des LGS bedeutet also die Suche nach der Minimumstelle von $h(x)$. Damit gilt $h(x^*) = 0$ und $h(x) > 0$ für $x \neq x^*$.

$h(x)$ hat mindestens dieselben Glattheitseigenschaften wie $f_i(x)$.

Das Funktional $h(x)$ lässt sich nun wie folgt umformen.

$$\begin{aligned}
 h(x) &= \sum_{i=1}^n f_i^2(x), \quad f_i(x) = a_i x - b_i \\
 &= \sum_{i=1}^n (a_i x - b_i)^2 \\
 &= \sum_{i=1}^n [(a_i x)^2 - 2(a_i x)b_i + b_i^2] \\
 &= \sum_{i=1}^n (a_i x)^T (a_i x) - 2 \sum_{i=1}^n (a_i x)^T b_i + \sum_{i=1}^n b_i^2 \\
 &= (Ax)^T (Ax) - 2(Ax)^T b + b^T b \\
 &= x^T A^T A x - 2x^T A^T b + b^T b, \\
 h(x) &= x^T B x - 2x^T c + b^T b = 2\left(\frac{1}{2}x^T B x - x^T c\right) + b^T b = 2R(x) + b^T b. \quad (3.7)
 \end{aligned}$$

mit

$$R(x) = \frac{1}{2}x^T B x - x^T c, \text{ wobei } B = A^T A = B^T > 0, \quad c = A^T b. \quad (3.8)$$

$h(x)$ ist ein zu minimierendes Funktional zum NGS $Bx = c$, dessen eindeutige Lösung (Minimumstelle) x^* ist. Da es nur um diese Stelle geht, könnten wir genauso das Funktional $R(x)$ nehmen. Die Minimumstelle existiert, sie ist global und eindeutig. Es gibt keine lokalen Minima.

Um die Minimumstelle von $h(x)$ zu finden, verwenden wir als Abstiegsverfahren das MGV. Dazu benötigen wir den Gradienten des Funktionals h an einer Stelle x , das ist

$$h'(x) = \nabla h(x) = \begin{pmatrix} h_{x_1}(x) \\ \vdots \\ h_{x_n}(x) \end{pmatrix} = 2Bx - 2c = 2(Bx - c). \quad (3.9)$$

An der Lösung x^* wird nicht nur $h(x^*) = 0$, sondern verschwindet auch der Gradient, also $h'(x^*) = 0$. Das ist ein Hinweis darauf, dass eine Nullstellensuche zur Funktion $h(x)$ in Verbindung mit dem Newton-Verfahren wegen des Terms $h(x)/h'(x)$ nicht unbedingt zu empfehlen ist.

Wir starten nun das MGV in einem Startpunkt $x^{(0)} \in \mathbb{R}^n$.

Als Abstiegsrichtung nehmen wir den negativen Gradienten und erhalten für die Abstiegsgerade im Punkt $x^{(0)}$

$$g = g(\alpha) = \{x^{(0)} - \alpha h'(x^{(0)}) \mid \alpha \in \mathbb{R}\}. \quad (3.10)$$

Anders als beim GV suchen wir jetzt nicht den Punkt auf g , an welchem das Funktional $h(x)$ minimal wird, sondern folgen einer $(n + 1)$ -dimensionalen Abstiegsgeraden durch den Punkt

$$\tilde{x}^{(0)} = \begin{pmatrix} x^{(0)} \\ x_{n+1}^{(0)} \end{pmatrix} \in \mathbb{R}^{n+1} \quad \text{mit} \quad x_{n+1}^{(0)} = z^{(0)} = h(x^{(0)})$$

bis diese Gerade die (x_1, \dots, x_n) -Ebene durchstößt, also ihre $(n + 1)$ -te Koordinate Null wird. Dazu legen wir eine Tangentialebene im Punkt $\tilde{x}^{(0)}$ an $h(x)$ an. Sie hat die Gleichung

$$z - z^{(0)} = \sum_{i=1}^n h_{x_i}(x^{(0)}) (x_i - x_i^{(0)}) = h'(x^{(0)})^T (x - x^{(0)}) \quad (3.11)$$

Nun folgen wir der Abstiegsgeraden g so lange, bis diese die Tangentialebene durchstößt. Dazu gehört ein entsprechender Wert α_0 . Den Durchstoßpunkt bezeichnen wir mit

$$x^{(1)} = x^{(0)} - \alpha_0 h'(x^{(0)}) \quad (3.12)$$

Setzt man in (3.10) $z = 0$, so reduziert sich die Tangentialebene auf eine Gerade in der (x_1, \dots, x_n) -Ebene, welche als Spur bezeichnet wird. Sie liegt also in der gleichen Ebene wie die Abstiegsgerade g . Damit ist $x^{(1)}$ auch der Schnittpunkt von g mit der Spur.

Auf der Geraden

$$l = l(\beta) = \tilde{x}^{(0)} + \beta (\tilde{x}^{(1)} - \tilde{x}^{(0)}), \quad \beta \in \mathbb{R},$$

welche die Punkte $\tilde{x}^{(0)} = (x^{(0)}, h(x^{(0)}))^T$ und $\tilde{x}^{(1)} = (x^{(1)}, 0)^T$ verbindet, befindet sich der in der Tangentialebene liegende Gradientenvektor zu $h(x)$. Es gilt $l(1) = \tilde{x}^{(1)} = (x^{(1)}, 0)^T$.

Mittels der folgenden Umformungen lässt sich nun $x^{(1)}$ bestimmen. Einerseits gilt

$$-z^{(0)} = -h(x^{(0)}) = \sum_{i=1}^n h_{x_i}(x^{(0)}) (x_i^{(1)} - x_i^{(0)}). \quad (3.13)$$

Andererseits hat man

$$\begin{aligned} x^{(1)} &= x^{(0)} - \alpha_0 h'(x^{(0)}) \quad \text{bzw. komponentenweise} \\ x_1^{(1)} &= x_1^{(0)} - \alpha_0 h_{x_1}(x^{(0)}), \\ &\vdots \\ x_n^{(1)} &= x_n^{(0)} - \alpha_0 h_{x_n}(x^{(0)}), \end{aligned}$$

so dass

$$\alpha_0 = \frac{x_j^{(1)} - x_j^{(0)}}{h_{x_j}(x^{(0)})} \quad \forall j = 1, 2, \dots, n. \quad (3.14)$$

Daraus folgen

$$x_i^{(1)} - x_i^{(0)} = \frac{x_j^{(1)} - x_j^{(0)}}{h_{x_j}(x^{(0)})} h_{x_i}(x^{(0)}) = \frac{h_{x_i}(x^{(0)})}{h_{x_j}(x^{(0)})} (x_j^{(1)} - x_j^{(0)}), \quad i, j = 1, 2, \dots, n, \quad (3.15)$$

und mit $k \in \{1, 2, \dots, n\}$ beliebig aber fest

$$x_i^{(1)} - x_i^{(0)} = \frac{h_{x_i}(x^{(0)})}{h_{x_k}(x^{(0)})} (x_k^{(1)} - x_k^{(0)}), \quad i = 1, 2, \dots, n. \quad (3.16)$$

Nun setzt man (3.15) in (3.12) ein und bestimmt daraus $x_k^{(1)}$.

$$\begin{aligned} -h(x^{(0)}) &= \sum_{i=1}^n \frac{h_{x_i}^2(x^{(0)})}{h_{x_k}(x^{(0)})} (x_k^{(1)} - x_k^{(0)}) \\ &= \frac{x_k^{(1)} - x_k^{(0)}}{h_{x_k}(x^{(0)})} \sum_{i=1}^n h_{x_i}^2(x^{(0)}), \\ x_k^{(1)} &= x_k^{(0)} - \frac{h(x^{(0)})}{\sum_{i=1}^n h_{x_i}^2(x^{(0)})} h_{x_k}(x^{(0)}), \quad k = 1, 2, \dots, n, \\ x^{(1)} &= x^{(0)} - \frac{h(x^{(0)})}{\sum_{i=1}^n h_{x_i}^2(x^{(0)})} h'(x^{(0)}) \quad \text{mit} \quad \sum_{i=1}^n h_{x_i}^2(x^{(0)}) > 0 \quad \text{bei} \quad x^{(0)} \neq x^*. \end{aligned}$$

Analog erhält man dies für weitere Iterierte, so dass

$$x^{(k+1)} = x^{(k)} - \frac{h(x^{(k)})}{\sum_{i=1}^n h_{x_i}^2(x^{(k)})} h'(x^{(k)}), \quad k = 0, 1, 2, \dots \quad (3.17)$$

Da das MGW auf das NGS mit der spd-Matrix B angewandt wird, ist die Konvergenz für jede reguläre Matrix A gesichert. Sie wird aber i. Allg. langsamer sein als beim GV zu (2.1) mit A spd bzw. beim GV zum NGS mit dem Funktional $R(x)$, wo die eindimensionale Minimumsuche entlang der Geraden des steilsten Abstiegs immer bis zur "lokalen" Minimumstelle führt. Die genannten Durchstoßpunkte zum MGW können entweder "davor" oder "danach" liegen.

Diese Herleitung der Iterationsvorschrift beschränkt sich natürlich nicht nur auf lineare Funktionen $f_i(x)$, sondern lässt sich auch auf nichtlineare Gleichungssysteme anwenden, wie wir in Kapitel 3.2 sehen werden.

Das somit erhaltene Iterationsverfahren MGW lässt sich in Form einer Maple-Prozedur implementieren.

```
> mgv:=proc(n::posint,A::matrix,b::vector,x0::vector,
            maxiter::posint,etol::numeric,aus::name)

    local k,i,m,d,dd,Dh,r,rr,v,x,x_iter,B,c,h,fh,fh1,xx,xv1,z;
    global xv;

    B:=evalm(transpose(A)&*A);
    c:=evalm(transpose(A)&*b);
    h:=x->evalm(transpose(x)&*(B&*x-2*c)+transpose(b)&*b);
    Dh:=x->evalm(2*(B&*x-c));

    fh1:='%+.16e';
    fh :=fh1;
    for m from 2 to n do, fh:=cat(fh,' ',fh1); end do;
    x:=evalm(x0);
    if aus=ja then
        printf('\n'):
        printf('Startvektor          x = [||fh||]\n',seq(x[i],i=1..n));
        printf('Funktionswert      h(x) = '||fh1||'\n\n',h(x0));
    end if;

    xx:=matrix(n,0,[]):
    xv1:=concat(xx,x);
    xv:=evalm(xv1);
    for k from 1 by 1 to maxiter do
        d:=evalm(Dh(x));
        dd:=evalm(transpose(d)&*d);
        if dd<etol then break; end if;
        x_iter:=evalm(x-(h(x)/dd)*d );
        z:=evalm(transpose(d)&*(x_iter-x)+h(x));

        if aus=ja then
            printf('Schritt k = %g\n',k);
            printf('Iterationsvektor    x = [||fh||]\n',seq(x_iter[i],i=1..n));
            printf('Funktionswert      h(x) = '||fh1||'\n',h(x_iter));
            printf('z-Koordinate        z = '||fh1||'\n\n',z);
        end if;

        x:=evalm(x_iter);
        xv1:=concat(xv,x);
        xv:=evalm(xv1);
        v:=evalm(A&*x);
        r:=evalm(b-v);
        rr:=evalm(transpose(r)&*r);
        if evalm(rr)<etol then printf('Abbruch, da Residuum < Epsilon'); break; end if;
    end do;
    if k>maxiter then k:=k-1; end if;
    [x,k];
end;
```

3.1.2 Beispiele

Im Folgenden betrachten wir die Beispiele aus Abschnitt 2.2 nun zum MGK sowie zum GV für das NGS.

Beispiel 3.1 Sei

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 3/2 \\ 1 \end{pmatrix}.$$

A ist spd.

Die Lösung des Problems ist $x^* = \frac{1}{5}(1, 3)^T$. Das MGK konvergiert.

Es gelten natürlich $h(x^*) = 0$ sowie $Q(x^*) = -\frac{7}{10}$ und $R(x^*) = -\frac{5}{2}$

```
> n:=2:
  A:=matrix(n,n,[[2,1],
                 [1,3]]):
  b:=vector(n,[1,2]):
  xs:=linsolve(A,b);
```

$$xs := \left[\frac{1}{5}, \frac{3}{5} \right]$$

Wir betrachten das Funktional $h(x)$ sowie das NGS mit dem Funktional $R(x)$.

```
> z:=vector(n,[x1,x2]);
  B:=evalm(transpose(A)&*A);
  c:=evalm(transpose(A)&*b);
  R:=expand(evalm(transpose(z)&*(B&*z-2*c)/2));
  h:=evalm(2*R+transpose(b)&*b);
```

$$\begin{aligned} z &:= [x_1, x_2] \\ B &:= \begin{bmatrix} 5 & 5 \\ 5 & 10 \end{bmatrix} \\ c &:= [4, 7] \\ R &:= \frac{5}{2}x_1^2 + 5x_1x_2 - 4x_1 + 5x_2^2 - 7x_2 \\ h &:= 5 + 5x_1^2 + 10x_1x_2 - 8x_1 + 10x_2^2 - 14x_2 \end{aligned}$$

Wählt man in der Prozedur die ausführlicher Ausgabe, so lassen sich neben den Iterierten selbst auch noch der Wert des Funktionals beobachten – welcher, je näher man der Lösung kommt, immer weiter gegen Null gehen sollte – sowie eine Kontrollrechnung der z -Koordinate, welche im Rahmen der eingestellten Genauigkeit Null sein sollte. Im weiteren verzichten wir allerdings auf die ausführliche Ausgabe.

```
> Digits:=20:
  mmax:=12:
  x0:=vector(n,[1.5,1.0]):
  epsi:=1E-16:
  erg:=mgv(n,A,b,x0,mmax,epsi,ja);
```

Startvektor $x = [+1.5000000000000000e+00, +1.0000000000000000e+00]$
 Funktionswert $h(x) = +1.5250000000000000e+01$

Schritt $k = 1$
 Iterationsvektor $x = [+1.1448630136986301e+00, +5.6130136986301370e-01]$
 Funktionswert $h(x) = +4.1131573700506662e+00$
 z-Koordinate $z = +0.0000000000000000e-01$

Schritt $k = 2$
 Iterationsvektor $x = [+9.0800968735594418e-01, +3.3456309487207510e-01]$
 Funktionswert $h(x) = +1.3316370908647039e+00$
 z-Koordinate $z = -2.0000000000000000e-19$

Schritt $k = 3$
 Iterationsvektor $x = [+6.4866884129802131e-01, +2.3076419426394288e-01]$
 Funktionswert $h(x) = +7.1322343688071791e-01$
 z-Koordinate $z = +0.0000000000000000e-01$

Schritt $k = 4$
 Iterationsvektor $x = [+5.8592633722748516e-01, +4.5967337995301941e-01]$
 Funktionswert $h(x) = +4.0005390686476742e-01$
 z-Koordinate $z = +1.0000000000000000e-20$

Schritt $k = 5$
 Iterationsvektor $x = [+4.4832008890452481e-01, +4.0069026664904071e-01]$
 Funktionswert $h(x) = +2.1063192379958703e-01$
 z-Koordinate $z = -1.0000000000000000e-20$

Schritt $k = 6$
 Iterationsvektor $x = [+4.0701415348046134e-01, +5.2736260606495858e-01]$
 Funktionswert $h(x) = +1.1666652251800998e-01$
 z-Koordinate $z = +0.0000000000000000e-01$

Schritt $k = 7$
 Iterationsvektor $x = [+3.3532652432972442e-01, +4.9442574435724204e-01]$
 Funktionswert $h(x) = +6.0155604732719280e-02$
 z-Koordinate $z = +0.0000000000000000e-01$

Schritt $k = 8$
 Iterationsvektor $x = [+3.0834853926372199e-01, +5.6317762520672007e-01]$
 Funktionswert $h(x) = +3.2359397446347528e-02$
 z-Koordinate $z = +3.0000000000000000e-21$

Schritt $k = 9$
 Iterationsvektor $x = [+2.7172801455052868e-01, +5.4540970773311974e-01]$
 Funktionswert $h(x) = +1.6369007674274333e-02$
 z-Koordinate $z = -1.0000000000000000e-21$

Schritt $k = 10$
 Iterationsvektor $x = [+2.5519134380712176e-01, +5.8154874698561367e-01]$
 Funktionswert $h(x) = +8.4514150462967176e-03$
 z-Koordinate $z = -2.0000000000000000e-21$

Schritt $k = 11$
 Iterationsvektor $x = [+2.3675621749060706e-01, +5.7237193148648656e-01]$
 Funktionswert $h(x) = +4.2331663676770404e-03$
 z-Koordinate $z = +2.0000000000000000e-21$

Schritt $k = 12$
 Iterationsvektor $x = [+2.2767640412177211e-01, +5.9077388977514703e-01]$
 Funktionswert $h(x) = +2.1276722738191402e-03$
 z-Koordinate $z = -1.4000000000000000e-21$

Verzichtet man in der Prozedur auf die ausführliche Ausgabe, so lassen sich das Ergebnis mittels `evalm(erg[1])` die Anzahl der Iterationen mittels `erg[2]` und die Iterierten mittels `evalm(xv)` ausgeben.

```
> Digits:=20:
  mmax:=100:
  x0:=vector(n,[1.5,1.0]):
  epsi:=1E-16:

> erg:=mgv(n,A,b,x0,mmax,epsi,nein);
  xm:=evalm(erg[1]);
  iter:=erg[2];
  evalm(xv);
  cd:=coldim(xv);
```

Abbruch, da Norm des Residuums < Epsilon

```
erg := [x, 57]
xm := [0.20000000439828813876, 0.59999999669997817228]
iter := 57

[1.5, 1.1448630136986301370, 0.90800968735594417785, 0.64866884129802130961, 0.58592633722748515752,
0.44832008890452480680, 0.40701415348046133981, 0.33532652432972441507, 0.30834853926372199226,
0.27172801455052868189, 0.25519134380712176119, 0.23675621749060706470, 0.22767640412177210880,
0.21845048652001129885, 0.21383916520449836173, 0.20922611000961502860, 0.20691958287378603303,
0.20461305524919066819, 0.20345979143689306007, 0.20230652762459537196, 0.20172989571844653285,
0.20115326381229768980, 0.20086494785922326667, 0.20057663190614884325, 0.20043247392961164425,
0.20028831595307444379, 0.20021623696480578079, 0.20014415797653723535, 0.20010811848240279478,
0.20007207898826858471, 0.20005405924120127987, 0.20003603949413413579, 0.20002702962060052044,
0.20001801974706752613, 0.20001351481029943352, 0.20000900987353237431, 0.20000675740515042751,
0.20000450493676492851, 0.20000337870257861997, 0.20000225246838583385, 0.20000168935129019834,
0.20000112623419345238, 0.20000084467564624922, 0.20000056311709760853, 0.20000042233783844172,
0.20000028155857078350, 0.20000021116890645183, 0.20000014077923677152, 0.20000010558453275040,
0.20000007038965829029, 0.20000005279241378985, 0.20000003519474350675, 0.20000002639632051564,
0.20000001759781231185, 0.20000001319767679048, 0.20000000879854330877, 0.20000000659892373587,
0.20000000439828813876]

[1.0, 0.56130136986301369863, 0.33456309487207509905, 0.23076419426394287565, 0.45967337995301940624,
0.40069026664904070596, 0.52736260606495857503, 0.49442574435724203803, 0.56317762520672007106,
0.54540970773311973818, 0.58154874698561367166, 0.57237193148648656160, 0.59077388977514702530,
0.58616141265269857117, 0.59538694475080929694, 0.59308041728913607363, 0.59769347237540462686,
0.59654020856310696601, 0.59884673618770231365, 0.59827010428155346933, 0.59942336809385115438,
0.59913505214077673386, 0.59971168404692557529, 0.59956752607038836542, 0.59985584202346277201,
0.59978376303519418169, 0.59992792101173144952, 0.59989188151759714448, 0.59996396050586574164,
0.59994594075879863166, 0.59998198025293296058, 0.59997297037939934816, 0.59999099012646677884,
0.59998648518970006745, 0.59999549506323340454, 0.59999324259484996315, 0.59999774753161688120,
0.59999662129742386393, 0.59999887376580623974, 0.59999831064871051191, 0.59999943688290223371,
0.59999915532435463585, 0.59999971844145014163, 0.59999957766217665275, 0.59999985922069554980,
0.59999978883108044792, 0.59999992961037496911, 0.59999989441553228914, 0.59999996480510085997,
0.59999994720770479742, 0.59999998240239973642, 0.59999997360366134653, 0.59999999120163142471,
0.59999998680207364150, 0.59999999560061308455, 0.59999999340116733234, 0.59999999780010723492,
0.59999999669997817228]

cd := 58
```

Die Konvergenz des MGv ist, bei gleichen Abbruchkriterien für Norm des Gradienten bzw. des Residuums, deutlich langsamer als die des GV (vergl. Beispiel 2.1).

Es braucht 57 Iterationsschritte gegenüber 16 des GV.

Plot der Höhenlinien des Funktionals $h(x)$ und des Iterationsverlaufes des MGV.

```
> p2:=contourplot(h,x1=0..2,x2=-0.5..1.5,scaling=constrained,contours=3,
  grid=[100,100],thickness=1,color=green):
p3:=contourplot(h,x1=0..2,x2=-0.5..1.5,grid=[100,100],thickness=1,
  contours=[seq(hh(xv[1,i],xv[2,i]),i=1..10)],color=red):
p4:=plot([seq([xv[1,i],xv[2,i]],i=1..cd)],
  style=line,thickness=3,color=black):
p5:=pointplot([seq([xv[1,i],xv[2,i]],i=1..cd)],symbol=circle):
p6:=plot([[0,xv[2,1]],[xv[1,1],xv[2,1]],[xv[1,1],0]],style=line,
  linestyle=DASH,thickness=2,color=black):
p7:=textplot([[1.5,1.15,'x(0)'],[1.2,0.45,'x(1)'],[0.95,0.25,'x(2)'],
  [0.6,0.15,'x(3)'],[0.2,0.7,'x*']]):

> plots[display]([p2,p3,p4,p5,p6,p7],
  title='MGV - Iterationsverlauf, x(0)=[1.5,1]');

> # Dateiausgabe des Plots
> pfad:='E:/Nico/Studium/Beleg/':
dateiname:='mgv_01a':
file:=cat(pfad,dateiname,'.ps'):
interface(plotdevice=ps,plotoutput=file,
  plotoptions='color,portrait,noborder,width=350,height=260');
plots[display]([p2,p3,p4,p5,p6,p7],
  title='MGV - Iterationsverlauf, x(0)=[1.5,1]');
interface(plotdevice=win);
lprint('Creating ',file);
```

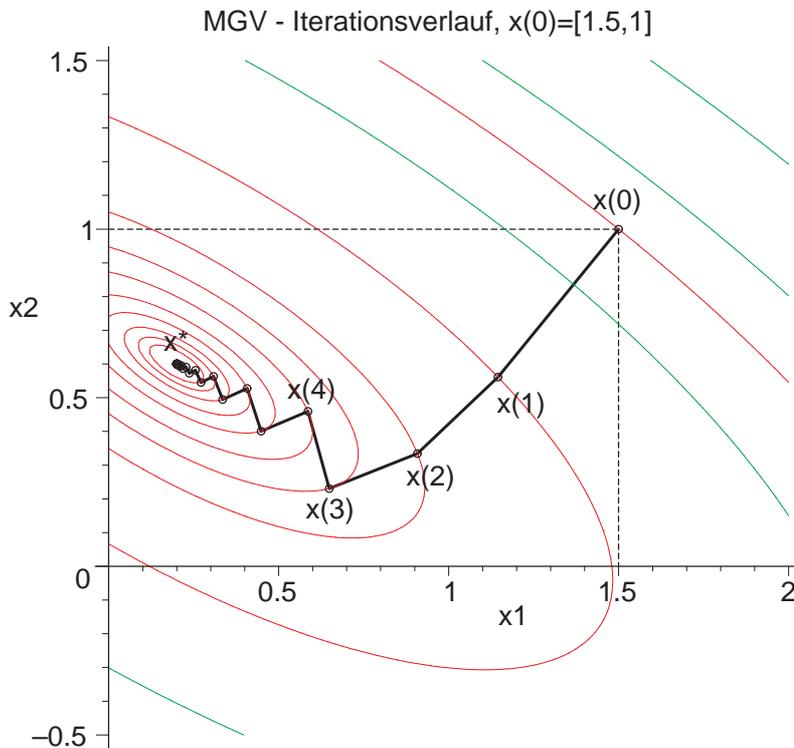


Abb. 3.2
Datei *mgv_01a.ps*
Konvergenz
des MGV

Das GV für das NGS $Bx = c$ ist mit dem Funktional $R(x)$ verknüpft und schneller als das MGK. Es bringt jedoch meistens eine Verschlechterung der Kondition des Systems durch die Multiplikation mit A^T , aber für beliebiges reguläres A ist die Matrix $B = A^T A$ spd. Im Allgemeinen ergeben sich dadurch mehr Iterationsschritte im Vergleich zum GV für das LGS, aber nicht in jedem Fall.

Mehr Informationen zu den Eigenschaften von Abstiegsverfahren mit dem Funktional $R(x)$ findet man in [3].

Abbruch, da Norm des Residuums $<$ Epsilon

```

erg := [x, 17]
xm := [0.2000000005229578, 0.5999999996631795]
xv := [[1.5, 0.8416445623342175, 0.2950254381006201,
0.2469019658774534, 0.2069460260663190, 0.2034283691194503, 0.2005077301307770, 0.2002506017519587,
0.2000371132908568, 0.2000183181086681, 0.2000027128513254, 0.2000013389894626, 0.2000001982999120,
0.2000000978754309, 0.2000000144950277, 0.2000000071543505, 0.2000000010595357, 0.2000000005229578]
[1.0, 0.1867374005305040, 0.6292385963386528, 0.5697919541806231, 0.6021372387896364, 0.5977918978552691,
0.6001562246556235, 0.5998385954817891, 0.6000114194741095, 0.5999882018961117, 0.6000008347234844,
0.5999991376000068, 0.6000000610153574, 0.5999999369615866, 0.6000000044600083, 0.5999999953921130,
0.6000000003260110, 0.599999996631795]

```

Ähnlich wie zum MGK erhält man das Höhenlinienbild zum Funktional $R(x)$ und darin den Iterationsverlauf des GV.

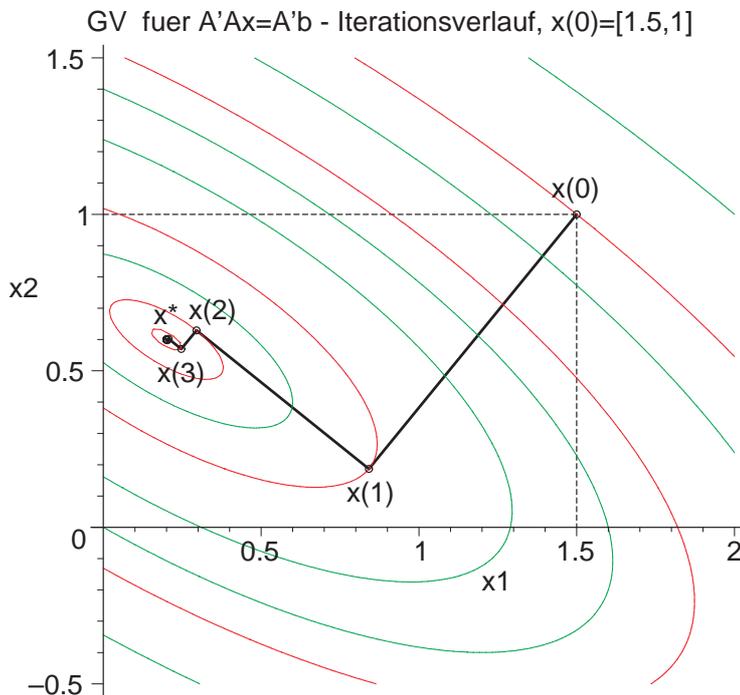


Abb. 3.3

Datei *mgv_01b.ps*

Konvergenz
des GV
für NGS

Im Iterationsverlauf und Konvergenzverhalten des GV erkennt man hier bezüglich des LGS $Ax = b$ (vergl. Beispiel 2.1) und des NGS $Bx = c$ keine wesentlichen Unterschiede. Das liegt mit an der kleinen Dimension des Problems und der Ausgeglichenheit der Elemente der Matrix $B = A^T A$. ◀

Beispiel 3.2 Sei

$$A = \begin{pmatrix} 2 & 1 \\ 1 & -3 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 3/2 \\ 1 \end{pmatrix}.$$

A ist symmetrisch und indefinit.

Die Lösung des Problems ist $x^* = \frac{1}{7}(10, 1)^T$. Das MGV konvergiert.

Es gelten $h(x^*) = 0$ sowie $Q(x^*) = -\frac{31}{14}$ und $R(x^*) = -5$

Abbruch, da Norm des Residuums $<$ Epsilon

```

erg := [x, 29]
xm := [1.42857143127686832366870, 0.142857143378162947335652]
xv := [1.5, 1.52500000000, 1.52243654822, 1.50866636865, 1.49048063406, 1.47020521019, 1.45126714396,
1.43991881105, 1.43424514843, 1.43140828678, 1.42998985778, 1.42928064317, 1.42892603587,
1.42874873222, 1.42866008040, 1.42861575448, 1.42859359153, 1.42858251005, 1.42857696931,
1.42857419894, 1.42857281376, 1.42857212116, 1.42857177487, 1.42857160172, 1.42857151515,
1.42857147186, 1.42857145021, 1.42857143939, 1.42857143398, 1.42857143128]
[1.0, 0.575000000000, 0.358388324873, 0.246539956980, 0.187916736877, 0.158120167737, 0.147223359346,
0.145042724192, 0.143949784932, 0.143403472824, 0.143130307304, 0.142993725113, 0.142925433983,
0.142891288420, 0.142874215639, 0.142865679248, 0.142861411053, 0.142859276955, 0.142858209906,
0.142857676382, 0.142857409619, 0.142857276238, 0.142857209548, 0.142857176202, 0.142857159530,
0.142857151193, 0.142857147025, 0.142857144941, 0.142857143899, 0.142857143378 ]

```

Die Konvergenz des MGV ist bei diesem Beispiel nicht signifikant langsamer als die des GV zum Sattelpunkt (vergl. Beispiel 2.2). Es braucht 29 Iterationsschritte gegenüber 22 des GV .

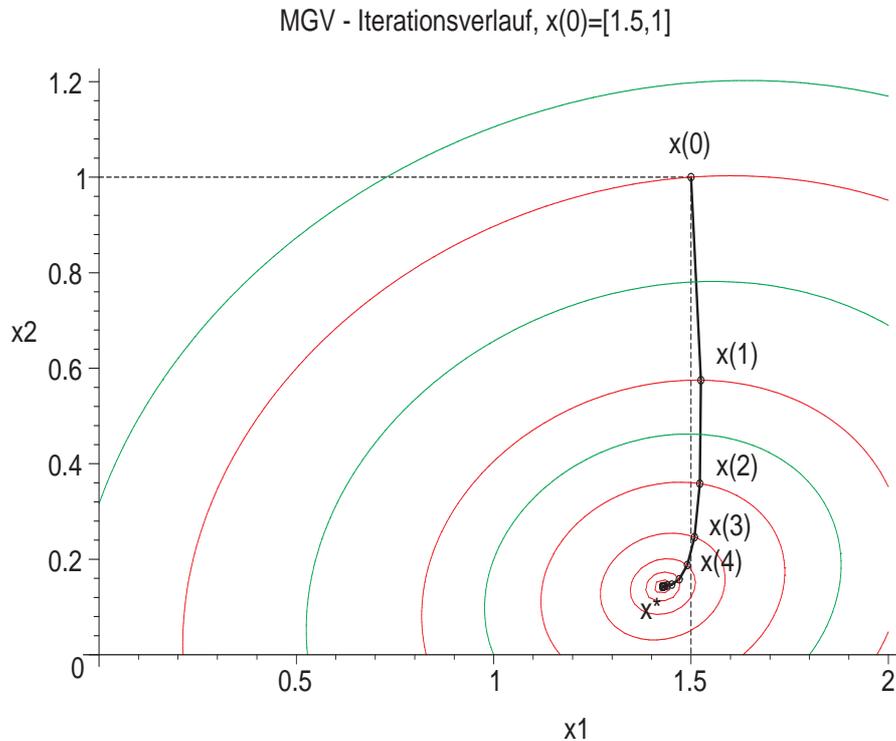


Abb. 3.4

Datei
mgv_02a.ps
 Konvergenz
 des MGV

Ähnlich wie zum MGW erhält man das Höhenlinienbild zum Funktional $R(x)$ und darin den Iterationsverlauf des GV.

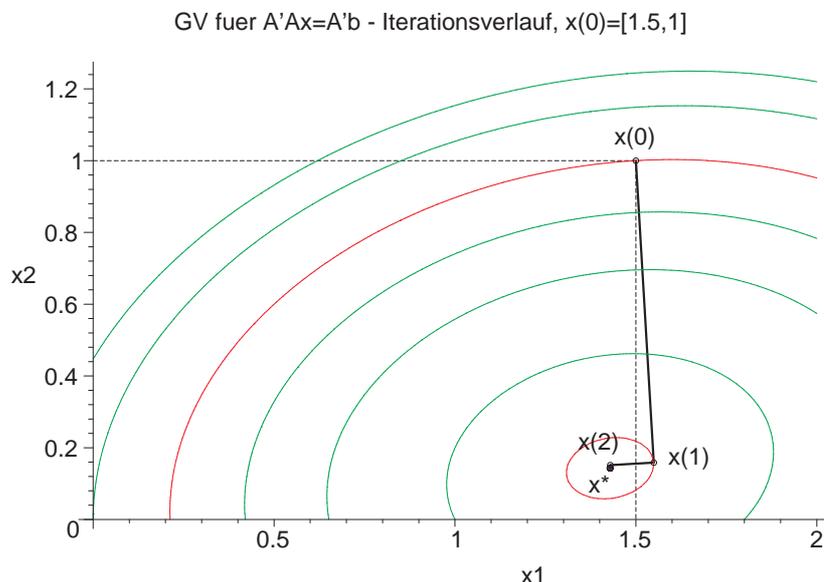


Abb. 3.5
 Datei *mgv_02b.ps*
 Konvergenz
 des GV
 für NGS ◀

Beispiel 3.3 Sei

$$A = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

A ist positiv definit aber nicht symmetrisch .

Die Lösung des Problems ist $x^* = (1, 1)^T$. Das MGW konvergiert.

Abbruch, da Norm des Residuums $<$ Epsilon

```

erg := [x, 31]
xm := [1.00000000490466127168882, 0.99999999876714545032320]
xv := [1.0, 1.19230769230769, 1.25744886040715, 1.26024498705185, 1.22458337102334, 1.14189056926171,
1.07088059149265, 1.03897828355116, 1.01933340243467, 1.01050278867293, 1.00519841523216,
1.00280904639897, 1.00138909818906, 1.00074859527726, 1.00037002121845, 1.00019911361847,
1.00009839576512, 1.00005290409643, 1.00002614006195, 1.00001404789549, 1.00000694057301,
1.00000372888221, 1.00000184222824, 1.00000098959078, 1.00000048888772, 1.00000026259078,
1.00000012972579, 1.00000006967420, 1.00000003442032, 1.00000001848613, 1.00000000913244,
1.00000000490466]
[-1.0, -0.038461538461538, 0.459676805828414, 0.728470258976837, 0.885664801806494, 0.971385701586938,
0.971714338732714, 0.991547998798251, 0.992475550056233, 0.997661469582865, 0.997999577689026,
0.999366567247891, 0.999468581849053, 0.999830056301951, 0.999858899937927, 0.999954628556994,
0.999962547586631, 0.999987919111433, 0.999990060851572, 0.999996788122324, 0.999997362643239,
0.999999146819134, 0.999999300223949, 0.999999773481842, 0.999999814334263, 0.999999939877665,
0.999999950740037, 0.999999984045152, 0.999999986930762, 0.999999995766450, 0.999999996532609,
0.99999999876715]
    
```

Die Konvergenz des MGW ist deutlich langsamer als die des GV (vergl. Beispiel 2.3).
 Es braucht 31 Iterationsschritte gegenüber 13 des GV.

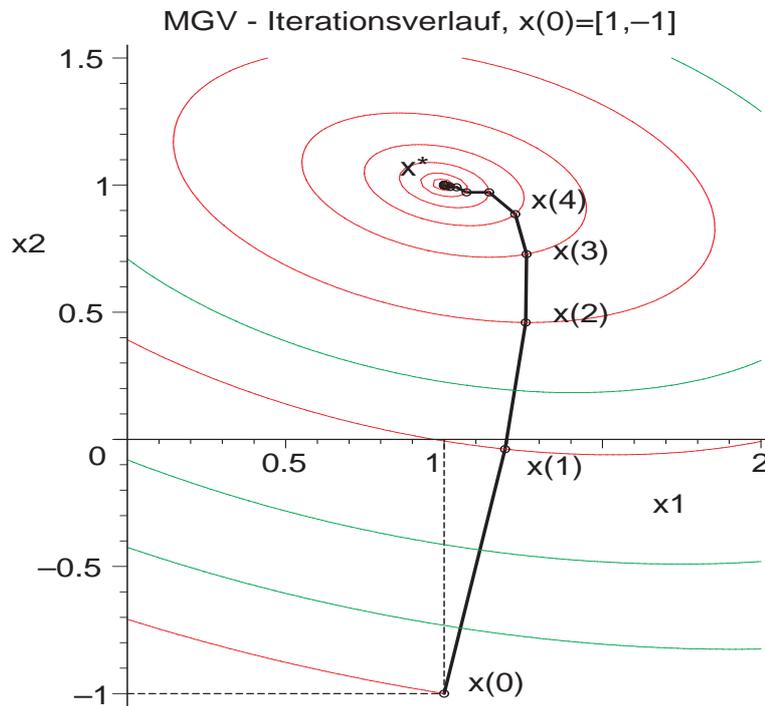


Abb. 3.6
 Datei *mgv_03a.ps*
 Konvergenz
 des MGV

Ähnlich wie zum MGV erhält man das Höhenlinienbild zum Funktional $R(x)$ und darin den Iterationsverlauf des GV.

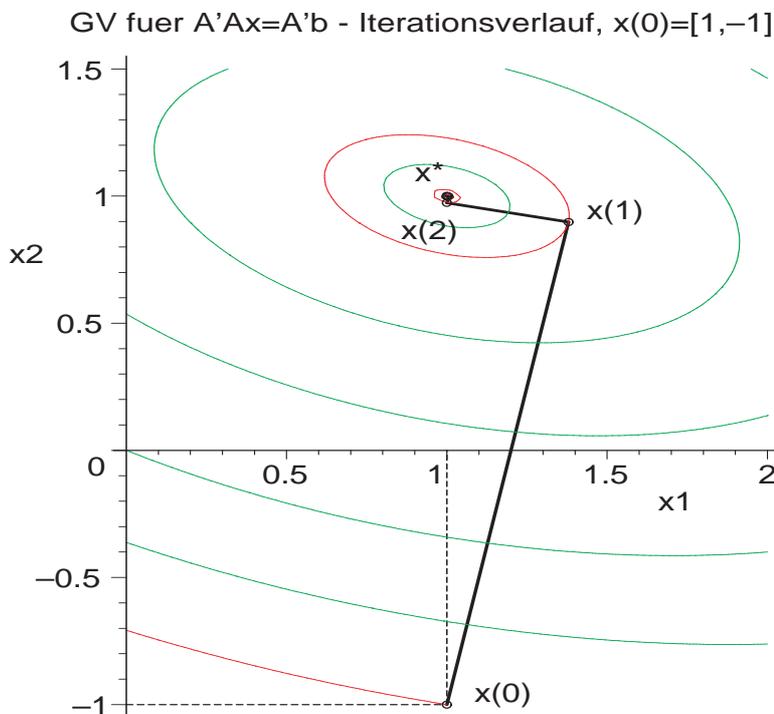


Abb. 3.7
 Datei *mgv_03b.ps*
 Konvergenz
 des GV
 für NGS ◀

Beispiel 3.4 Sei

$$A = \begin{pmatrix} 0.780 & 0.563 \\ 0.913 & 0.659 \end{pmatrix}, \quad b = \begin{pmatrix} 0.217 \\ 0.254 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 1.2 \\ -1.2 \end{pmatrix}.$$

A ist nicht symmetrisch und indefinit.

Die Lösung des Problems ist $x^* = (1, -1)^T$.

Bei diesem Beispiel (vergl. Beispiel 2.4) divergiert das GV, konvergiert beim NGS aber schon in drei Schritten (`Digits=25`).

Das MGV hingegen läuft bis zur maximalen Iterationsanzahl und sich an einer Stelle fest bzw. liefert **Abbruch wegen Division durch 0**, an der der Gradient entsprechend der gegebenen Genauigkeit `Digits=10,16,25,30,50,100,...` "nahezu" 0 wird, und konvergiert nicht (folgende Rechnung mit `Digits=25`).

Abbruch wegen Division durch 0

```

erg := [x, 152]
xm := [1.163417972853129796396177, -1.226404749069646386249264]
xv := [1.2, 1.18170898574, 1.17256347861, 1.16799072505, 1.16570434826, 1.16456115987, 1.16398956567, 1.16370376857,
1.16356087001, 1.16348942071, 1.16345369602, 1.16343583360, 1.16342690225, 1.16342243627, 1.16342020268, 1.16341908470,
1.16341852330, 1.16341823774, 1.16341808460, 1.16341798094, 1.16341741262, 1.16341770159, 1.16341785628, 1.16341796013,
1.16341843508, 1.16341819179, 1.16341805745, 1.16341795248, 1.16341824248, 1.16341808732, 1.16341798340, 1.16341753061,
1.16341776313, 1.16341789288, 1.16341799997, 1.16341779868, 1.16341791592, 1.16341803959, 1.16341792731, 1.16341807003,
1.16341796669, 1.16341908168, 1.16341852177, 1.16341823694, 1.16341808413, 1.16341798051, 1.16341738657, ...]
[-1.2, -1.21320237514, -1.21980356271, -1.22310415649, -1.22475445339, -1.22557960183, -1.22599217606, -1.22619846318,
-1.22630160674, -1.22635317854, -1.22637896446, -1.22639185748, -1.22639830409, -1.22640152762, -1.22640313981, -1.22640394677,
-1.22640435198, -1.22640455810, -1.22640466863, -1.22640474345, -1.22640515366, -1.22640494508, -1.22640483343, -1.22640475847,
-1.22640441565, -1.22640459126, -1.22640468822, -1.22640476398, -1.22640455466, -1.22640466666, -1.22640474166, -1.22640506849,
-1.22640490066, -1.22640480700, -1.22640472971, -1.22640487499, -1.22640479037, -1.22640470111, -1.22640478215, -1.22640467914,
-1.22640475373, -1.22640394892, -1.22640435305, -1.22640455864, -1.22640466894, -1.22640474373, -1.22640517243, ...]

```

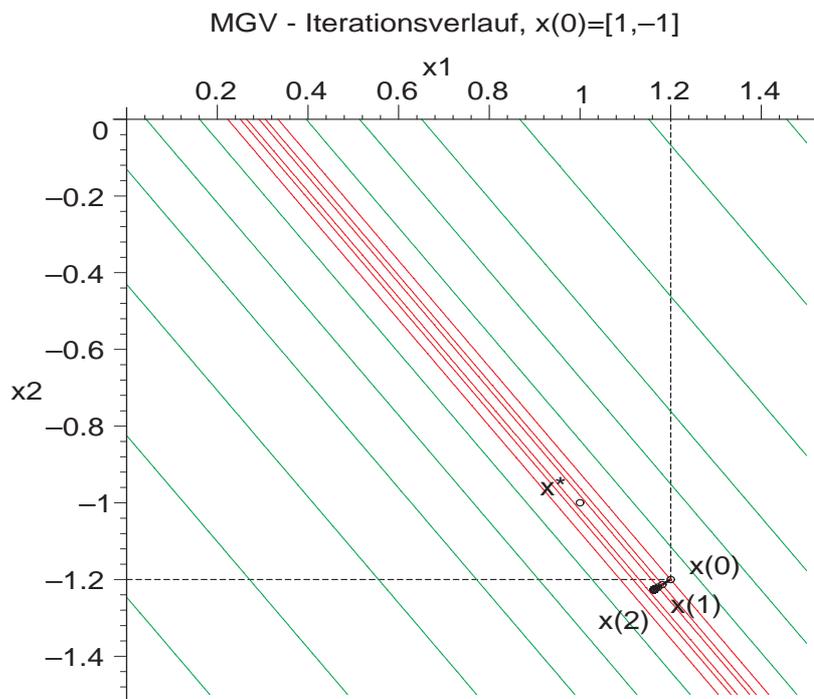


Abb. 3.8
 Datei `mgv_04a.ps`
 MGV
 konvergiert nicht

Beispiel 3.5 Sei

$$A = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 6 \\ 2 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

A ist spd. Die Lösung des Problems ist $x^* = (1, 2, 1)^T$. Das MGW konvergiert.

Abbruch, da Norm des Residuums < Epsilon

```

erg := [x, 44]
xm := [0.999999986301170492898918, 1.99999997318170342701592, 0.999999986301170492898917]
xv := [ 0, 0.107843137255, 0.597905022125, 0.648087101709, 0.840468051608, 0.861010727692, 0.936914322748,
0.945093802231, 0.975071958445, 0.978308932600, 0.990151423651, 0.991430712803, 0.996109164360,
0.996614604617, 0.998462876271, 0.998662560025, 0.999392740941, 0.999471628667, 0.999760095160,
0.999791260709, 0.999905222777, 0.999917535095, 0.999962557146, 0.999967421272, 0.999985207761,
0.999987129391, 0.999994156153, 0.999994915315, 0.999997691320, 0.999997991236, 0.999999087929,
0.999999206414, 0.999999639676, 0.999999686485, 0.999999857650, 0.999999876142, 0.999999943763,
0.999999951068, 0.999999977783, 0.999999980669, 0.999999991223, 0.999999992363, 0.999999996532,
0.999999996983, 0.99999998630]
[0, 1.07843137255, 1.20896704391, 1.63964114985, 1.68734414050, 1.85793545773, 1.87646719331, 1.94390135091,
1.95119563238, 1.97783978721, 1.98071915019, 1.99124554030, 1.99238286442, 1.99654146471, 1.99699076041,
1.99863366652, 1.99881116453, 1.99946021406, 1.99953033660, 1.99978675124, 1.99981445396, 1.99991575358,
1.99992669786, 1.99996671746, 1.99997104113, 1.99998685134, 1.99998855946, 1.99999480547, 1.99999548028,
1.99999794784, 1.99999821443, 1.99999918927, 1.99999929459, 1.99999967971, 1.99999972132, 1.99999987347,
1.99999988990, 1.99999995001, 1.99999995651, 1.99999998025, 1.99999998282, 1.99999999220, 1.99999999321,
1.99999999692, 1.99999999732]
[0, 0.107843137255, 0.597905022125, 0.648087101709, 0.840468051608, 0.861010727692, 0.936914322748,
0.945093802231, 0.975071958445, 0.978308932600, 0.990151423651, 0.991430712803, 0.996109164360,
0.996614604617, 0.998462876271, 0.998662560025, 0.999392740941, 0.999471628667, 0.999760095160,
0.999791260709, 0.999905222777, 0.999917535095, 0.999962557146, 0.999967421272, 0.999985207761,
0.999987129391, 0.999994156153, 0.999994915315, 0.999997691320, 0.999997991236, 0.999999087929,
0.999999206414, 0.999999639676, 0.999999686485, 0.999999857650, 0.999999876142, 0.999999943763,
0.999999951068, 0.999999977783, 0.999999980669, 0.999999991223, 0.999999992363, 0.999999996532,
0.999999996983, 0.99999998630]

```

Die Konvergenz des MGW ist, bei gleichen Abbruchkriterien für Norm des Gradienten bzw. des Residuums, deutlich langsamer als die des GV (vergl. Beispiel 2.5). Es braucht 44 Iterationsschritte gegenüber 14 des GV.

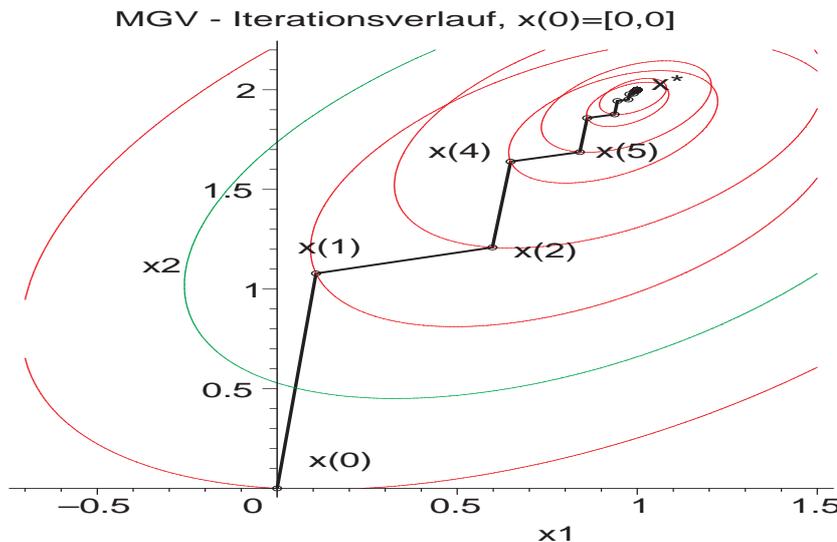


Abb. 3.10
Datei *mgv_05b.ps*
Konvergenz
des MGW ◀

Beispiel 3.6 Sei

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 9/2 \\ 3 \end{pmatrix}.$$

A ist diagonal und spd. Die Lösung des Problems ist $x^* = (0, 0)^T$.
Das MGV konvergiert.

Abbruch, da Norm des Residuums $<$ Epsilon

$$\begin{aligned} \text{erg} &:= [x, 40] \\ \text{xm} &:= [0.8433464170063693664910444, -0.1271392560682148826434761 \cdot 10^{-8}] \\ \text{xv} &:= \\ &[4.5, 3.72945205479, 2.57356380814, 1.39328627020, 0.815572804873, 0.490877684756, 0.294310940940, 0.176614953570, \\ &0.105965176344, 0.0635796115972, 0.0381476995102, 0.0228886286989, 0.0137331760203, 0.823990577206 \cdot 10^{-2}, \\ &0.494394344192 \cdot 10^{-2}, 0.296636606800 \cdot 10^{-2}, 0.177981964042 \cdot 10^{-2}, 0.106789178430 \cdot 10^{-2}, \\ &0.640735070574 \cdot 10^{-3}, 0.384441042345 \cdot 10^{-3}, 0.230664625407 \cdot 10^{-3}, 0.138398775244 \cdot 10^{-3}, \\ &0.830392651466 \cdot 10^{-4}, 0.498235590879 \cdot 10^{-4}, 0.298941354528 \cdot 10^{-4}, 0.179364812717 \cdot 10^{-4}, \\ &0.107618887630 \cdot 10^{-4}, 0.645713325780 \cdot 10^{-5}, 0.387427995468 \cdot 10^{-5}, 0.232456797281 \cdot 10^{-5}, \\ &0.139474078368 \cdot 10^{-5}, 0.836844470211 \cdot 10^{-6}, 0.502106682126 \cdot 10^{-6}, 0.301264009276 \cdot 10^{-6}, \\ &0.180758405565 \cdot 10^{-6}, 0.108455043339 \cdot 10^{-6}, 0.650730260036 \cdot 10^{-7}, 0.390438156021 \cdot 10^{-7}, \\ &0.234262893613 \cdot 10^{-7}, 0.140557736168 \cdot 10^{-7}, 0.843346417006 \cdot 10^{-8}] \\ &[3.0, 0.945205479452, -0.226604092911, 0.189092957262, -0.124529734485, 0.0737809343122, -0.0443982248860, \\ &0.0266218044577, -0.0159753712914, 0.958491776170 \cdot 10^{-2}, -0.575099132946 \cdot 10^{-2}, 0.345058937480 \cdot 10^{-2}, \\ &-0.207035434793 \cdot 10^{-2}, 0.124221251235 \cdot 10^{-2}, -0.745327520267 \cdot 10^{-3}, 0.447196510446 \cdot 10^{-3}, \\ &-0.268317906496 \cdot 10^{-3}, 0.160990743867 \cdot 10^{-3}, -0.965944463244 \cdot 10^{-4}, 0.579566677941 \cdot 10^{-4}, \\ &-0.347740006765 \cdot 10^{-4}, 0.208644004059 \cdot 10^{-4}, -0.125186402435 \cdot 10^{-4}, 0.751118414613 \cdot 10^{-5}, \\ &-0.450671048768 \cdot 10^{-5}, 0.270402629261 \cdot 10^{-5}, -0.162241577556 \cdot 10^{-5}, 0.973449465338 \cdot 10^{-6}, \\ &-0.584069679203 \cdot 10^{-6}, 0.350441807522 \cdot 10^{-6}, -0.210265084513 \cdot 10^{-6}, 0.126159050708 \cdot 10^{-6}, \\ &-0.756954304247 \cdot 10^{-7}, 0.454172582548 \cdot 10^{-7}, -0.272503549529 \cdot 10^{-7}, 0.163502129717 \cdot 10^{-7}, \\ &-0.981012778304 \cdot 10^{-8}, 0.588607666982 \cdot 10^{-8}, -0.353164600189 \cdot 10^{-8}, 0.211898760114 \cdot 10^{-8}, \\ &-0.127139256068 \cdot 10^{-8}] \end{aligned}$$

Die Konvergenz des MGV ist, bei gleichen Abbruchkriterien für Norm des Gradienten bzw. des Residuums, deutlich langsamer als die des GV (vergl. Beispiel 2.6).
Es braucht 40 Iterationsschritte gegenüber 19 des GV.

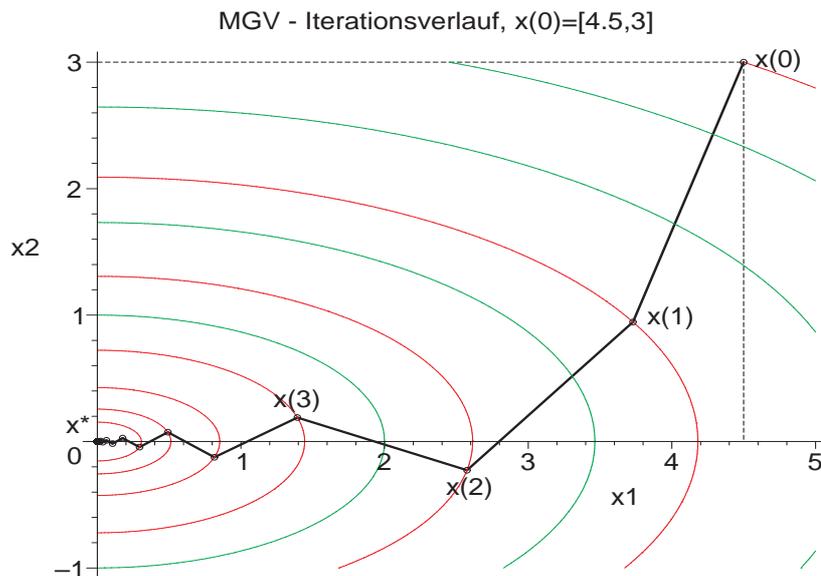


Abb. 3.11
Datei *mgv_06a.ps*
Konvergenz
des MGV ◀

Beispiel 3.7 Sei

$$A = \begin{pmatrix} 1 & 0 \\ 0 & -4 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

A ist diagonal und indefinit. Die Lösung des Problems ist $x^* = (0, 0)^T$. Das MGW konvergiert, wohingegen das GV in Abhängigkeit vom Startvektor ganz unterschiedliches Verhalten (Konvergenz, Divergenz, Abbruch) zeigt (vergl. Beispiel 2.7).

Abbruch, da Norm des Residuums $<$ Epsilon

```

erg := [x, 90]
xm := [0.9238354322357389962480822 · 10-8, -0.373566113795012335919753 · 10-9]
xv := [2.0, 1.923076923076923, 1.782781275663838, 1.100310849644827, 1.044421678944452,
       .7534188495418641, .7134967700251554, .4727625416360105, .4489075876415532,
       .3277088399712075, .3101266806287703, .1998948084118636, .1898358171991258,
       .1392467038569377, .1317348991255022, .8386438942890926e-1, .7964050705027700e-1,
       .5832768526236089e-1, .5518675292520083e-1, .3527580103836335e-1, .3349988902599397e-1,
       .2455363322152196e-1, .2323025432037990e-1, .1481908363079949e-1, .1407288939893203e-1,
       .1031110982276340e-1, .9755590425937225e-2, .6228999191724935e-2, .5915376254840534e-2,...]
[1.0, .3846153846153846, -.643306871064877e-1, .3296944648062150, .617505436356294e-1,
 -.2135341736755185, -.324985488579568e-1, .1429419220568327, .275394466324452e-1,
 -.914247779973002e-1, -.1294315926590623e-1, .6066540226203547e-1, .1182109234769733e-1,
 -.3858192132417249e-1, -.528046701343466e-2, .2542102807066985e-1, .493549264938085e-2,
 -.1619732614704757e-1, -.224176998411661e-2, .1069924123445631e-1, .2081022298094929e-2,
 -.6810879546294558e-2, -.937451881415563e-3, .4493443659620007e-2, .873266516616521e-3,
 -.2861615273341124e-2, -.394865807307031e-3, .1889002594608341e-2, .367256906048023e-3,...]
    
```

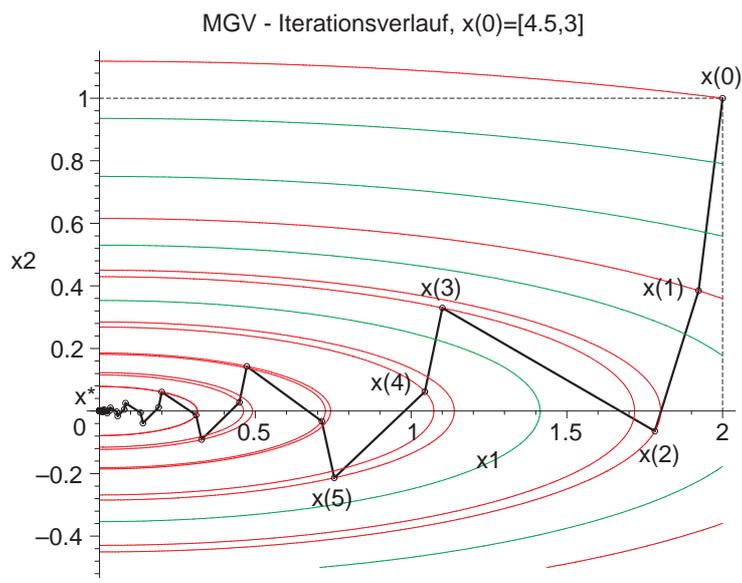


Abb. 3.12
 Datei *mgv_07a.ps*
 Konvergenz
 des MGW ◀

Beispiel 3.8 Sei

$$A = \begin{pmatrix} -1.0 & 0 & 2.1 & 0.4 \\ 0.2 & 2.3 & 0 & 1.0 \\ 2.0 & -0.4 & -5.0 & 0 \\ 0 & 1.0 & -1.0 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 1.0 \\ -1.0 \end{pmatrix} \quad \text{und} \quad x^{(0)} = \begin{pmatrix} 1.5 \\ -1.0 \\ 0 \\ 1.0 \end{pmatrix}.$$

A ist nicht symmetrisch und indefinit.

Die Lösung des Problems ist $x^* = \begin{pmatrix} 1.383\,640\,552\,995 \\ -0.691\,244\,239\,631 \\ 0.408\,755\,760\,368 \\ 1.313\,133\,640\,553 \end{pmatrix}.$

Das MGV konvergiert, wohingegen das GV erst nach der Symmetrisierung als NGS wesentlich langsamer konvergiert (vergl. Beispiel 2.8). Unabhängig von der Genauigkeit $\text{Digits}=k$, $k>15$, stagniert das MGV nach ca 360 Iterationen und der absolute Fehler ist von der Größenordnung $\mathcal{O}(10^{-8})$ (Rechnung mit $\text{Digits}=35$). Bei geringerer Genauigkeit ist die Stagnation entsprechend "früher", aber der Fehler größer. Z. B. hat man mit $\text{Digits}=10$ ca 180 Iterationen und die Fehlerordnung $\mathcal{O}(10^{-5})$.

Abbruch, da Norm des Residuums $<$ Epsilon

$$\text{erg} := [x, 377]$$

$$xm := [1.3836405729803364367351197024147815, -0.69124423527862037468264106710669989, 0.4087557687938460560111351295537757, 1.3131336280188310626514504708108073]$$

$$\begin{aligned} x^{(0)} &= [1.500000000, -1.000000000, 0.000000000, 1.000000000] \\ x^{(20)} &= [1.384952216, -0.657499566, 0.397530356, 1.224338652] \\ x^{(40)} &= [1.398280560, -0.681255549, 0.413014348, 1.285398592] \\ x^{(60)} &= [1.395636735, -0.687487739, 0.413480451, 1.302489777] \\ x^{(80)} &= [1.390483627, -0.689390153, 0.412969987, 1.308004425] \\ x^{(100)} &= [1.387368094, -0.690415242, 0.410340719, 1.310751085] \\ x^{(200)} &= [1.383665643, -0.691241704, 0.408771077, 1.313115692] \\ x^{(300)} &= [1.383641893, -0.691243981, 0.408756023, 1.313132857] \\ x^{(360)} &= [1.383640580, -0.691244234, 0.408755773, 1.313133623] \\ x^{(370)} &= [1.383640576, -0.691244235, 0.408755769, 1.313133626] \\ x^{(371)} &= [1.383640575, -0.691244235, 0.408755771, 1.313133626] \\ x^{(372)} &= [1.383640576, -0.691244235, 0.408755769, 1.313133626] \\ x^{(373)} &= [1.383640571, -0.691244235, 0.408755774, 1.313133628] \\ x^{(374)} &= [1.383640573, -0.691244235, 0.408755771, 1.313133628] \\ x^{(375)} &= [1.383640573, -0.691244235, 0.408755769, 1.313133628] \\ x^{(376)} &= [1.383640574, -0.691244235, 0.408755767, 1.313133628] \\ x^{(377)} &= [1.383640573, -0.691244235, 0.408755769, 1.313133628] \\ x^* &= [1.383640553, -0.691244239, 0.408755760, 1.313133640] \quad \blacktriangleleft \end{aligned}$$

3.2 Nichtlinearer Fall

3.2.1 Theorie

Mit Hilfe des MGV ist es auch möglich, aufgrund der ursprünglichen Definition des Funktionals

$$h(x) = f_1^2(x) + f_2^2(x) + \dots + f_n^2(x) \geq 0 \quad (3.18)$$

nichtlineare Gleichungssysteme (NLGS)

$$f(x) = (f_1(x), f_2(x), \dots, f_n(x))^T = 0, \quad x \in \mathbb{R}^n, \quad (3.19)$$

zu lösen.

Die Iterationsvorschrift

$$x^{(k+1)} = x^{(k)} - \frac{h(x^{(k)})}{\sum_{i=1}^n h_{x_i}^2(x^{(k)})} h'(x^{(k)}), \quad k = 0, 1, 2, \dots, \quad x^{(0)} \text{ gegeben}, \quad (3.20)$$

erhält man wie im Abschnitt 3.1.1.

Die Durchführbarkeit verlangt einen nicht verschwindenden Nenner. Die Konvergenz gegen eine Lösung verlangt eine entsprechende Glattheit der Funktion $f(x)$ und einen hinreichend guten Startvektor dazu. Die Situation ist ähnlich wie beim Newton-Verfahren, nur dass bei $h(x)$ die Nullstelle gleichzeitig auch Minimumstelle ist. Dadurch geht die Konvergenzordnung mindestens um Eins im Vergleich zur quadratischen Konvergenz des Newton-Verfahrens (bei $f'(x^*) \neq 0$) zurück.

An der Abb. 3.1 erkennt man schon, dass im linearen skalaren Fall das MGV für $h(x)$ identisch ist mit dem Newton-Verfahren zu $h(x) = 0$. Beide stellen einen unendlichen Iterationsprozess dar. Aber das GV für $h(x)$ und das Newton-Verfahren für $f(x)$ wären jeweils nach einer Iteration an der exakten Lösung x^* .

Das MGV zu $h(x)$ und Newton-Verfahren zu $f^2(x) = 0$ liefern auch für den nichtlinearen skalaren Fall die gleiche Iterationsvorschrift.

Newton-Verfahren zu $f(x) = 0$

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad k = 0, 1, \dots \quad (3.21)$$

Newton-Verfahren zu $h(x) = f^2(x) = 0$

$$x^{(k+1)} = x^{(k)} - \frac{h(x^{(k)})}{h'(x^{(k)})} = x^{(k)} - \frac{f^2(x^{(k)})}{2f(x^{(k)})f'(x^{(k)})} = x^{(k)} - \frac{1}{2} \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad k = 0, 1, \dots \quad (3.22)$$

MGV zu $h(x) = f^2(x) \geq 0$

$$\begin{aligned}
 x^{(k+1)} &= x^{(k)} - \alpha_k h'(x^{(k)}), \quad \alpha_k = \frac{h(x^{(k)})}{[h'(x^{(k)})]^2} \\
 &= x^{(k)} - \frac{h(x^{(k)})}{h'(x^{(k)})} \\
 &= x^{(k)} - \frac{1}{2} \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad k = 0, 1, \dots
 \end{aligned} \tag{3.23}$$

Die Situation ändert sich im mehrdimensionalen Fall, auch bei LGS. Betrachten wir dazu das einfache LGS aus Beispiel 2.6 bzw. 3.6 mit diagonaler spd Matrix.

Das GV (2.5) liefert im ersten Schritt

$$\begin{aligned}
 x^{(0)} &= (9/2, 3)^T, \quad r^{(0)} = b - Ax^{(0)} = -(9/2, 6)^T, \\
 \alpha_0 &= \frac{r(x^{(0)})^T r(x^{(0)})}{r(x^{(0)})^T A r(x^{(0)})} = \frac{225}{369}, \\
 x^{(1)} &= x^{(0)} + \alpha_0 r(x^{(0)}) = \frac{1}{41} (72, -29)^T = (1.756\dots, -0.658\dots)^T.
 \end{aligned}$$

Beim MGV benötigen wir

$$h(x) = h(x_1, x_2) = x_1^2 + 4x_2^2, \quad h_{x_1}(x) = 2x_1, \quad h_{x_2}(x) = 8x_2.$$

Dann folgt (3.16)

$$\begin{aligned}
 x^{(1)} &= x^{(0)} - \frac{h(x^{(0)})}{\sum_{i=1}^2 h_{x_i}^2(x^{(0)})} h'(x^{(0)}) \\
 &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \frac{x_1^{(0)2} + 4x_2^{(0)2}}{4x_1^{(0)2} + 64x_2^{(0)2}} \begin{pmatrix} 2x_1^{(0)} \\ 8x_2^{(0)} \end{pmatrix} \\
 &= \begin{pmatrix} 9/2 \\ 3 \end{pmatrix} - \frac{25}{292} \begin{pmatrix} 9 \\ 24 \end{pmatrix} \\
 &= \frac{1}{292} \begin{pmatrix} 1089 \\ 276 \end{pmatrix} \\
 &= (3.729\dots, 1.057\dots)^T.
 \end{aligned}$$

Beim entsprechenden Newton-Verfahren benötigen wir $f_1(x) = (Ax - b)_1 = x_1$, $f_2(x) = (Ax - b)_2 = 2x_2$ und

$$F(x) = \begin{pmatrix} f_1^2(x) \\ f_2^2(x) \end{pmatrix} = \begin{pmatrix} x_1^2 \\ 4x_2^2 \end{pmatrix}$$

und die Jacobi-Matrix mit den partiellen Ableitungen

$$DF(x) = \begin{pmatrix} \frac{\partial f_1^2(x)}{\partial x_1} & \frac{\partial f_1^2(x)}{\partial x_2} \\ \frac{\partial f_2^2(x)}{\partial x_1} & \frac{\partial f_2^2(x)}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 & 0 \\ 0 & 8x_2 \end{pmatrix}, \quad \frac{\partial f_i(x)}{\partial x_j} = f_{i,x_j}(x).$$

Daraus ergibt sich

$$\begin{aligned} x^{(1)} &= x^{(0)} - [DF(x^{(0)})]^{-1} F(x^{(0)}) \\ &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} \begin{pmatrix} 1/(2x_1^{(0)}) & 0 \\ 0 & 1/(8x_2^{(0)}) \end{pmatrix} \begin{pmatrix} x_1^{(0)2} \\ 4x_2^{(0)2} \end{pmatrix} \\ &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \frac{1}{2} \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 9/2 \\ 3 \end{pmatrix} = (2.25, 1.5)^T. \end{aligned}$$

Die drei Verfahren liefern als unterschiedliche Ergebnisse.

Gleichzeitig können wir im \mathbb{R}^2 zeigen, dass MGW sowie die Newton-Verfahren für $f(x) = 0$ und $F(x) = 0$ immer die Gestalt

$$\begin{aligned} x^{(1)} &= x^{(0)} - \gamma C f(x^{(0)}) \text{ bzw.} \\ \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \gamma C \begin{pmatrix} f_1(x^{(0)}) \\ f_2(x^{(0)}) \end{pmatrix} \end{aligned} \quad (3.24)$$

annehmen, wobei $\gamma \in \mathbb{R}$ und $C \in \mathbb{R}^{2 \times 2}$.

(1) Newton-Verfahren für $f(x) = 0$

$$\begin{aligned} x^{(1)} &= x^{(0)} - [Df(x^{(0)})]^{-1} f(x^{(0)}), \\ \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \begin{pmatrix} f_{1,x_1}(x^{(0)}) & f_{1,x_2}(x^{(0)}) \\ f_{2,x_1}(x^{(0)}) & f_{2,x_2}(x^{(0)}) \end{pmatrix}^{-1} \begin{pmatrix} f_1(x^{(0)}) \\ f_2(x^{(0)}) \end{pmatrix} \\ &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \frac{\begin{pmatrix} f_{2,x_2}(x^{(0)}) & -f_{1,x_2}(x^{(0)}) \\ -f_{2,x_1}(x^{(0)}) & f_{1,x_1}(x^{(0)}) \end{pmatrix}}{f_{1,x_1}(x^{(0)}) f_{2,x_2}(x^{(0)}) - f_{2,x_1}(x^{(0)}) f_{1,x_2}(x^{(0)})} \begin{pmatrix} f_1(x^{(0)}) \\ f_2(x^{(0)}) \end{pmatrix} \\ x^{(1)} &= x^{(0)} - \gamma_1 C_1 f(x^{(0)}) \end{aligned}$$

(2) Newton-Verfahren für $F(x) = 0$

$$\begin{aligned}
 x^{(1)} &= x^{(0)} - [DF(x^{(0)})]^{-1} F(x^{(0)}), \\
 \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \begin{pmatrix} 2f_1(x^{(0)})f_{1,x_1}(x^{(0)}) & 2f_1(x^{(0)})f_{1,x_2}(x^{(0)}) \\ 2f_2(x^{(0)})f_{2,x_1}(x^{(0)}) & 2f_2(x^{(0)})f_{2,x_2}(x^{(0)}) \end{pmatrix}^{-1} \begin{pmatrix} f_1^2(x^{(0)}) \\ f_2^2(x^{(0)}) \end{pmatrix} \\
 &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \left[2 \begin{pmatrix} f_1(x^{(0)}) & 0 \\ 0 & f_2(x^{(0)}) \end{pmatrix} \begin{pmatrix} f_{1,x_1}(x^{(0)}) & f_{1,x_2}(x^{(0)}) \\ f_{2,x_1}(x^{(0)}) & f_{2,x_2}(x^{(0)}) \end{pmatrix} \right]^{-1} \begin{pmatrix} f_1^2(x^{(0)}) \\ f_2^2(x^{(0)}) \end{pmatrix} \\
 &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \frac{1}{2} \begin{pmatrix} f_{1,x_1}(x^{(0)}) & f_{1,x_2}(x^{(0)}) \\ f_{2,x_1}(x^{(0)}) & f_{2,x_2}(x^{(0)}) \end{pmatrix}^{-1} \begin{pmatrix} f_1^{-1}(x^{(0)}) & 0 \\ 0 & f_2^{-1}(x^{(0)}) \end{pmatrix} \begin{pmatrix} f_1^2(x^{(0)}) \\ f_2^2(x^{(0)}) \end{pmatrix} \\
 &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \frac{1}{2} \frac{\begin{pmatrix} f_{2,x_2}(x^{(0)}) & -f_{1,x_2}(x^{(0)}) \\ -f_{2,x_1}(x^{(0)}) & f_{1,x_1}(x^{(0)}) \end{pmatrix}}{f_{1,x_1}(x^{(0)})f_{2,x_2}(x^{(0)}) - f_{2,x_1}(x^{(0)})f_{1,x_2}(x^{(0)})} \begin{pmatrix} f_1(x^{(0)}) \\ f_2(x^{(0)}) \end{pmatrix} \\
 x^{(1)} &= x^{(0)} - \frac{1}{2} \gamma_1 C_1 f(x^{(0)})
 \end{aligned}$$

(3) MGV für $h(x) = f_1^2(x) + f_2^2(x)$, $h_{x_i}(x) = 2[f_1(x)f_{1,x_i}(x) + f_2(x)f_{2,x_i}(x)]$, $i = 1, 2$

$$\begin{aligned}
 x^{(1)} &= x^{(0)} - \frac{h(x^{(0)})}{h_{x_1}^2(x^{(0)}) + h_{x_2}^2(x^{(0)})} h'(x^{(0)}) \\
 \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} &= \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \frac{2h(x^{(0)})}{h_{x_1}^2(x^{(0)}) + h_{x_2}^2(x^{(0)})} \begin{pmatrix} f_{1,x_1}(x^{(0)}) & f_{2,x_1}(x^{(0)}) \\ f_{1,x_2}(x^{(0)}) & f_{2,x_2}(x^{(0)}) \end{pmatrix} \begin{pmatrix} f_1(x^{(0)}) \\ f_2(x^{(0)}) \end{pmatrix} \\
 x^{(1)} &= x^{(0)} - \gamma_2 C_2 f(x^{(0)})
 \end{aligned}$$

Die Korrektur zum Punkt $x^{(0)}$ ist damit ein Vektor, der aus $f(x^{(0)})$ entsteht und einer Matrixtransformation γC unterworfen wird. Die Transformation ist in jeder der drei Varianten eine andere.

Das somit erhaltene Iterationsverfahren MGV zum nichtlinearen Fall lässt sich in Form einer Maple-Prozedur implementieren. Sie ist für ein System der Dimension 2 ausgelegt.

```

> mgvnl:=proc(n::posint,x0::vector,maxiter::posint,etol::numeric,aus::name)
  local k,i,m,d,dd,x,x_iter,B,c,fh,fh1,xx,xv1;
  global xv,h,Dh;

  fh1:='%+.16e';
  fh :=fh1;
  for m from 2 to n do, fh:=cat(fh,' ',fh1); end do;

```

```

x:=evalm(x0):
if aus=ja then
  printf('\n'):
  printf('Startvektor          x = [||fh||']\n',seq(x[i],i=1..n));
  printf('Funktionswert      h(x) = '||fh1||'\n\n',h(x[1],x[2]));
end if;
xx:=matrix(n,0,[]):
xv1:=concat(xx,x);
xv:=evalm(xv1);

for k from 1 by 1 to maxiter do
  d:=evalf(Dh(x[1],x[2]));          # Dimension = 2
  dd:=evalm(transpose(d(x))&*d(x));
  x_iter:=evalm(x-(h(x[1],x[2])/dd)*d );
  x:=evalm(x_iter);
  if aus=ja then
    printf('Schritt k = %g\n',k);
    printf('Iterationsvektor    x = [||fh||']\n',seq(x[i],i=1..n));
    printf('Funktionswert      h(x) = '||fh1||'\n\n',h(x[1],x[2]));
  end if;

  x:=evalm(x_iter);
  xv1:=concat(xv,x);
  xv:=evalm(xv1);
end do;
if k>maxiter then k:=k-1; end if;
[x,k];
end:

```

Die Daten zum System werden wie folgt bereitgestellt (Beispiel).

```

> n:=2:
f1:=(x1*sin(x2)+x2)/4-x1;
f2:=arctan(4/(x1+x2))-x2;

> hh:=f1^2+f2^2;
h:=unapply(hh,[x1,x2]);          # unapply(hh,x1,x2), unapply(hh,(x1,x2));
D[1](h)(x1,x2), diff(h(x1,x2),x1); # Test der partiellen Ableitungen
D[2](h)(x1,x2), diff(h(x1,x2),x2);

> # Fehler bei Verwendung von
Dh:=(x1,x2)->vector([D[1](h)(x1,x2),D[2](h)(x1,x2)]);
Dh:=(x1,x2)->vector([diff(h(x1,x2),x1),diff(h(x1,x2),x2)]);

> # Moeglich sind
Dh:=(x1,x2)->vector([2*(1/4*x1*sin(x1)+1/4*x2-x1)*(1/4*sin(x1)+1/4*x1*cos(x1)-1)
-8*(arctan(4*1/(x1+x2))-x2)/((x1+x2)^2*(1+16*1/((x1+x2)^2))),
1/8*x1*sin(x1)+1/8*x2-1/2*x1+2*(arctan(4*1/(x1+x2))-x2)
*(-4*1/((x1+x2)^2*(1+16*1/((x1+x2)^2)))-1]]);
# Dh:=(x1,x2)->[ , ];
Dh:=unapply([D[1](h)(x1,x2),D[2](h)(x1,x2)], [x1,x2]);
Dh:=unapply([diff(h(x1,x2),x1),diff(h(x1,x2),x2)], [x1,x2]);

```

3.2.2 Beispiele im skalaren Fall

Die Darstellung und Erklärung zum MGV und Newton-Verfahren für den linearen skalaren Fall ist einfach (vergl. Abb. 3.1).

Bei nichtlinearen Funktionen können der Iterationsprozess und das Konvergenzverhalten schon viel komplizierter sein sowie sehr unterschiedliche Situationen auftreten. Sind die Funktionen hinreichend glatt in der Umgebung einer Lösung x^* , so ist die lokale Konvergenz von MGV und Newton-Verfahren zu x^* gesichert.

Wählt man weiter entfernte oder beliebige Startpunkte, können beispielsweise folgende Situationen eintreten.

- Konvergenz gegen eine andere noch vorhandene Lösung.
- Abbruch der Iterationsprozesses, weil eine Iterierte zufällig an eine Extremwertstelle (nicht verschwindendes lokales Minimum oder Maximum) gelangt und damit eine Nulldivision durch $f'(x)$ bzw. $h'(x)$ verursacht.
- Anfänglicher Iterationsverlauf in der Umgebung von Extremwertstellen, um danach diesen Bereich zu verlassen und entweder zu konvergieren oder divergieren.
- Divergenz der Iterationsprozesses, weil die Iterierten in die Nähe einer Extremwertstelle kommen und sich nicht von dieser mehr entfernen können.
- Divergenz der Iterationsprozesses in “Unendliche“ ohne Lösung.
- Divergenz der Iterationsprozesses in “Unendliche“ gegen eine “Pseudo-Lösung“.
- Eventuell sonstige divergente Situationen.

Dazu kommt noch bei numerischen Rechnungen wegen möglicher betragskleiner Nenner in der Iterationsformel die Wahl oder sogar die Notwendigkeit einer hinreichend starken Gleitpunktarithmetik.

Beispiel 3.9 Nichtlineare Fälle

$$(1) \quad f(x) = \frac{2x}{x^2 + 1}, \quad h(x) = f^2(x). \quad (3.25)$$

Die einzige Lösung (Nullstelle von $f(x)$ und absolutes Minimum von $h(x)$) ist $x^* = 0$. $f(x)$ hat noch zwei “Pseudo-Nullstellen“ im Unendlichen und 2 Extremwertstellen. $h(x)$ hat dazu zwei lokale Maxima bei ± 1 und “Pseudo-Minima“ im Unendlichen.

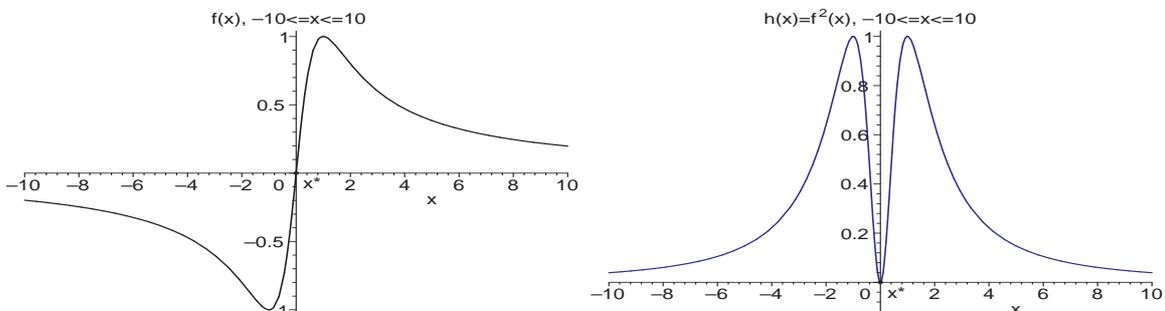


Abb. 3.13 Dateien *mgv_nl1_01a.ps*, *mgv_nl1_01b.ps*
 $f(x)$ bzw. $h(x) = f^2(x)$, $-10 \leq x \leq 10$

(2)

$$f(x) = \frac{2x}{x^2 + 1} + \frac{1}{2}, \quad h(x) = f^2(x). \quad (3.26)$$

Die Funktion $f(x)$ entsteht durch eine vertikale Verschiebung von (3.25).

Es gibt zwei Lösungen x^* (Nullstellen von $f(x)$ und absolute Minima von $h(x)$).

Zusätzlich hat $f(x)$ zwei Extremwertstellen sowie "Pseudo-Extremwertstellen" im Unendlichen.

$h(x)$ hat noch zwei lokale Maxima sowie je ein "Pseudo-Maximum" und "Pseudo-Minimum" im Unendlichen.

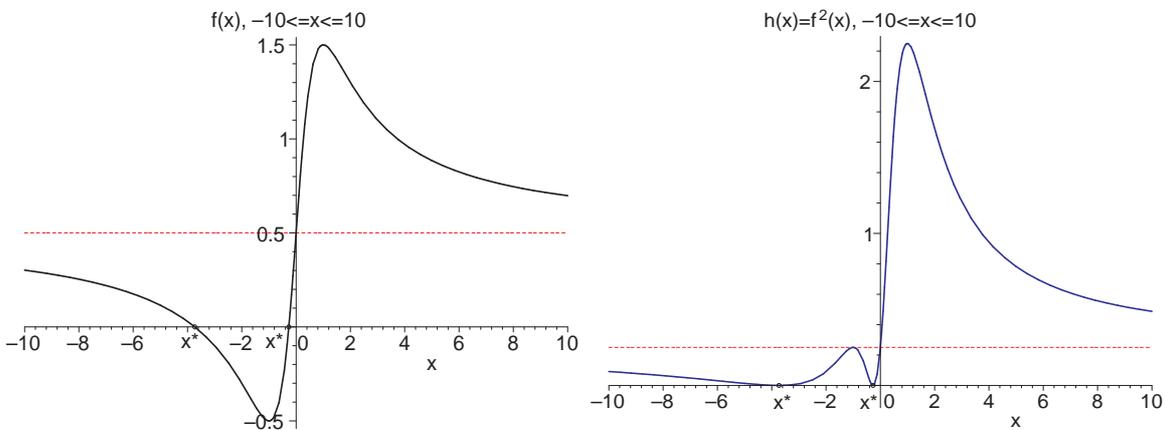


Abb. 3.14 Dateien *mgv_nl1_02a.ps*, *mgv_nl1_02b.ps*
 $f(x)$ bzw. $h(x) = f^2(x)$, $-10 \leq x \leq 10$

(3)

$$f(x) = \begin{cases} \sqrt{\frac{19}{128}}(x+5)\sqrt{x^2 + \frac{14}{19}x + \frac{11}{19}} & \text{für } x \leq -1, \\ \sqrt{x^2 + 1} & \text{für } -1 < x \leq 2, \\ \sqrt{5}e^{2/3 - 3/50(x-16/3)^2} & \text{für } 2 < x, \end{cases} \quad h(x) = f^2(x). \quad (3.27)$$

$f(x)$ ist eine stückweise zusammengesetzte Funktion, die zweimal stetig differenzierbar ist, genauso $h(x)$. An den beiden Intervallgrenzen -1 und 2 gibt es keine dritte Ableitung.

Die einzige Lösung x^* (Nullstelle von $f(x)$ und absolutes Minimum von $h(x)$) ist $x^* = -5$.

Zusätzlich hat $f(x)$ drei Extremwertstellen, das sind ein Minimum bei 0 , Maxima bei $-(29 + 2\sqrt{101})/19 = -2.584197433$ und $16/3$ sowie eine "Pseudo-Nullstelle" und zugleich "Extremwertstelle" im Unendlichen.

$h(x)$ hat entsprechend an den genannten Stellen ein lokales Minimum, zwei lokale Maxima sowie ein "Pseudo-Minimum" im Unendlichen.

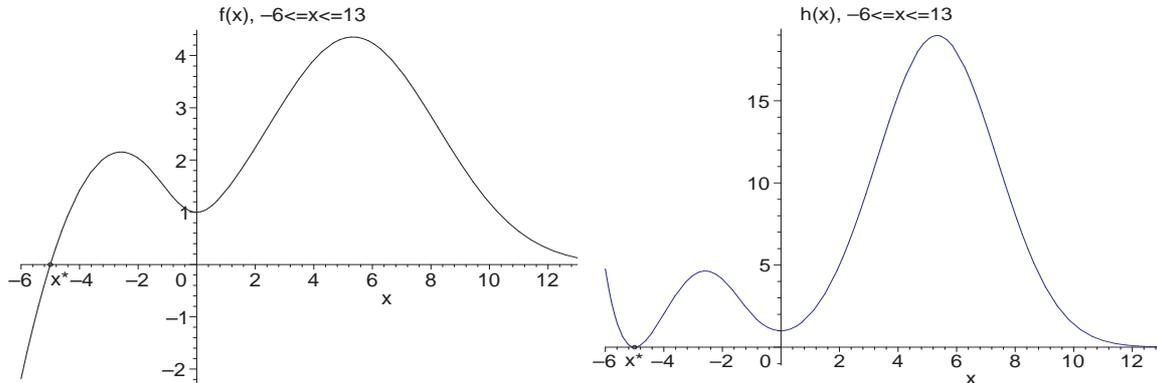


Abb. 3.15 Dateien *mgv_nl1_03a.ps*, *mgv_nl1_03b.ps*

$f(x)$ bzw. $h(x) = f^2(x)$, $-6 \leq x \leq 13$

$$(4) \quad f(x) = \begin{cases} \sqrt{\frac{19}{128}}(x+5)\sqrt{x^2 + \frac{14}{19}x + \frac{11}{19}} - \frac{9}{10} & \text{für } x \leq -1, \\ \sqrt{x^2 + 1} - \frac{9}{10} & \text{für } -1 < x \leq 2, \\ \sqrt{5}e^{2/3 - 3/50(x-16/3)^2} - \frac{9}{10} & \text{für } 2 < x, \end{cases} \quad h(x) = f^2(x). \quad (3.28)$$

Die Funktion $f(x)$ entsteht durch eine vertikale Verschiebung von (3.27).

$f(x)$ bleibt damit eine stückweise zusammengesetzte Funktion, die zweimal stetig differenzierbar ist, genauso $h(x)$.

Es gibt zwei Lösungen x^* (Nullstellen von $f(x)$ und absolute Minima von $h(x)$).

Zusätzlich hat $f(x)$ drei Extremwertstellen, ein Minimum bei 0 und zwei Maxima bei $-(29 + 2\sqrt{101})/19 = -2.584197433$ und $16/3$ sowie ein "Pseudo-Minimum" im Unendlichen.

$h(x)$ hat entsprechend an den genannten Stellen ein lokales Minimum > 0 (mit der Verschiebung kann man beliebig nahe der Null kommen), zwei lokale Maxima sowie "Pseudo-Maximum" im Unendlichen.

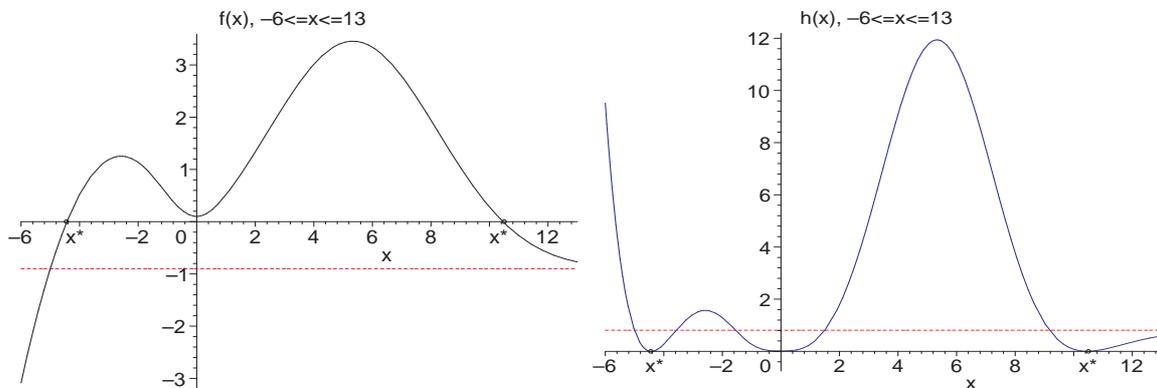


Abb. 3.16 Dateien *mgv_nl1_04a.ps*, *mgv_nl1_04b.ps*

$f(x)$ bzw. $h(x) = f^2(x)$, $-6 \leq x \leq 13$ ◀

Wird man nun zu einer der im Beispiel 3.9 genannten Funktionen die Iteration des Newton-Verfahrens für $f(x) = 0$ bzw. des MGV für $h(x)$ (entspricht dem Newton-Verfahren zu $h(x) = 0$) durchführen, so lassen en sich ganz unterschiedliche Iterationsverläufe und Situationen von Konvergenz, Divergenz und Abbruch beobachten. An der Funktion (3.27) soll dies beispielhaft, aber nicht in aller Vollständigkeit, demonstriert werden.

Beispiel 3.10 Betrachten wir also die Funktionen

$$f(x) = \begin{cases} \sqrt{\frac{19}{128}}(x+5)\sqrt{x^2 + \frac{14}{19}x + \frac{11}{19}} & \text{für } x \leq -1, \\ \sqrt{x^2 + 1} & \text{für } -1 < x \leq 2, \\ \sqrt{5}e^{2/3 - 3/50(x-16/3)^2} & \text{für } 2 < x, \end{cases} \quad (3.29)$$

und

$$h(x) = f^2(x) = \begin{cases} p(x) = \frac{19}{128}(x+5)^2(x^2 + \frac{14}{19}x + \frac{11}{19}) & \text{für } x \leq -1, \\ q(x) = x^2 + 1 & \text{für } -1 < x \leq 2, \\ g(x) = 5e^{4/3 - 3/25(x-16/3)^2} & \text{für } 2 < x. \end{cases} \quad (3.30)$$

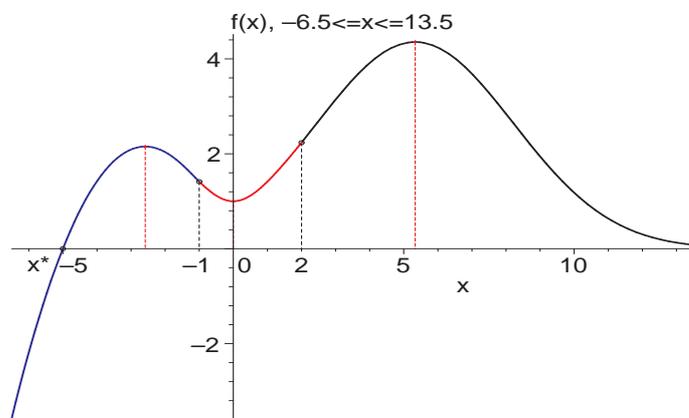


Abb. 3.17

Datei *mgv_nl1_031a.ps*

$f(x)$, $-6.5 \leq x \leq 13.5$

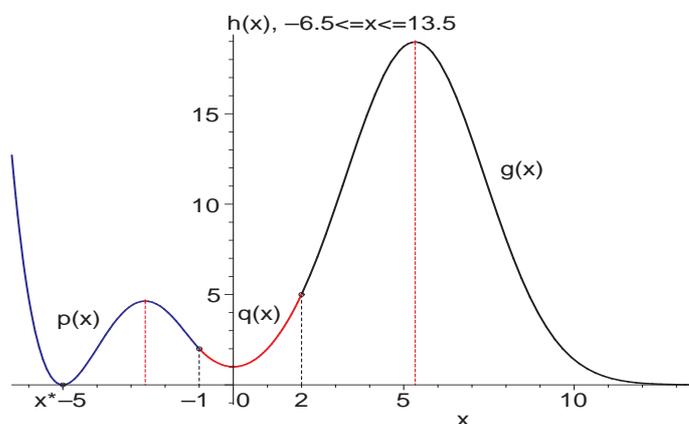


Abb. 3.18

Datei *mgv_nl1_031b.ps*

$h(x) = f^2(x)$, $-6.5 \leq x \leq 13.5$

Die Nullstelle von $f(x)$ und das absolute Minimum von $h(x)$ ist bei $x^* = -5$.

An den Teilintervallgrenzen gelten

$$\begin{aligned} f(-1) &= \sqrt{2}, & f'(-1) &= -\sqrt{2}/2, & f''(-1) &= \sqrt{2}/4, & f'''(-1-) &= 15\sqrt{2}/8, & f'''(-1+) &= 3\sqrt{2}/8, \\ f(2) &= \sqrt{5}, & f'(2) &= 2\sqrt{5}/5, & f''(2) &= \sqrt{5}/25, & f'''(2-) &= -6\sqrt{5}/125, & f'''(2+) &= -2\sqrt{5}/25, \\ h(-1) &= 2, & h'(-1) &= -2, & h''(-1) &= 2, & h'''(-1-) &= 6, & h'''(-1+) &= 0, \\ h(2) &= 5, & h'(2) &= 4, & h''(2) &= 2, & h'''(2-) &= 0, & h'''(2+) &= -\frac{8}{25}. \end{aligned}$$

Das lokale Minimum ist bei $x_{min} = 0$ mit $f(x_{min}) = h(x_{min}) = 1$.

Für die beiden lokalen Maxima haben wir

$$\begin{aligned} x_{max1} &= -(29 + 2\sqrt{101})/19 = -2.584\,197 \quad \text{mit} \\ f(x_{max1}) &= \sqrt{38} (33 - \sqrt{101}) \sqrt{262 + 22\sqrt{101}} / 1444 = 2.153\,411, \\ h(x_{max1}) &= (20641 + 1111\sqrt{101})/6859 = 4.637\,179, \end{aligned}$$

und $x_{max2} = 16/3 = 5.333\,333$ mit

$$\begin{aligned} x_{max2} &= 16/3 = 5.333\,333 \quad \text{mit} \\ f(x_{max2}) &= \sqrt{5} e^{2/3} = 4.355\,266, \\ h(x_{max2}) &= 5 e^{4/3} = 18.968\,339. \end{aligned}$$

Bevor das Newton-Verfahren für verschiedene Startwerte gerechnet wird, muss auf eine Besonderheit hingewiesen werden.

Liegt der Iterationsverlauf im Bereich der mittleren Funktionen von f und h , erhalten wir bei $\sqrt{x^2 + 1}$ die Newton-Iteration

$$x^{(k+1)} = x^{(k)} - \frac{\sqrt{x^{(k)2} + 1}}{x^{(k)}(x^{(k)2} + 1)^{-1/2}} = -\frac{1}{x^{(k)}},$$

so dass $x^{(k+1)}x^{(k)} = -1$, $x^{(0)} = x^{(2)} = \dots$, $x^{(1)} = x^{(3)} = \dots$ und $|x^{(k+1)} - x^{(k)}| = \text{const}$ gelten und so die Konvergenz gegen $x^* = -5$ verhindert wird. Ähnliche Eigenschaften sind noch in der unmittelbaren Nachbarschaft davon zu erkennen.

Mit der mittleren Funktion $x^2 + 1$ lautet die Newton-Iteration

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)2} + 1}{2x^{(k)}} = \frac{1}{2} \left(x^{(k)} - \frac{1}{x^{(k)}} \right).$$

Das ist, abgesehen vom Startwert $x^{(0)} = \pm 1$ mit Abbruch der Iteration dann bei $x^{(1)} = 0$, ein sich aufschaukelnder Prozess, wo irgendwann mal die Iterierte "nach außen driftet". Sie kann dann entweder in der gewünschten Umgebung von x^* landen oder auch im ungünstigen positiven Bereich $> x_{max2}$.

Für speziell ausgewählte Startwerte gelingt es, die Iteration zu einer Extremwertstelle zu führen, wo dann der Abbruch wegen Nulldivision eintritt.

Starten wir z. B. das Newton-Verfahren zu $f(x) = 0$ mit $x^{(0)} = x_w = 19/(29+2\sqrt{101}) = 0.386\,967$, so ist $x^{(1)} = -1/x^{(0)} = x_{max1} = -2.584\,197$ und $f'(x^{(1)}) = 0$.

Es ist also eine große Vielfalt im Verhalten der Iterationen zu erwarten.

Newton-Verfahren für $f(x) = 0$

In den meisten Fällen beschränken wir uns nur auf das Zeigen des grafischen Verlaufs der Iteration. Das Newton-Verfahren ist als Maple-Prozedur mit grafischer und Datei-Ausgabe sowie zusätzlichen Tests zur Durchführbarkeit implementiert.

Normalfall mit Konvergenz

```
> # Parameter: Funktion, Startwert, maximale Iterationsanzahl
  qf1:=Newton2(f,-3.5,6):
  display(qf1,p12,p13,p14,p15,p16,p17,view=[-6.5..13.5,-3.8..4.4],
         title='f(x), -6.5<=x<=13.5, Newton-Verfahren mit x(0)=-3.5');
```

Iterationsverlauf

```
x = -3.500000000000000000000000e+00, f(x)= +1.85022697087411413618e+00
x = -6.26895734597156398104e+00, f(x)= -2.90306775701951220300e+00
x = -5.22226914285739857782e+00, f(x)= -4.19550503577436796405e-01
x = -5.00956057106158108505e+00, f(x)= -1.72704230163696018080e-02
x = -5.00001925847661963693e+00, f(x)= -3.47188539105770154528e-05
x = -5.00000000007845663925e+00, f(x)= -1.41439717865983537045e-10
x = -5.0000000000000000000000e+00, f(x)= -1.80277563773199464656e-21
delta = 7.84566392520000000000e-11 # |x(k+1)-x(k)|
```

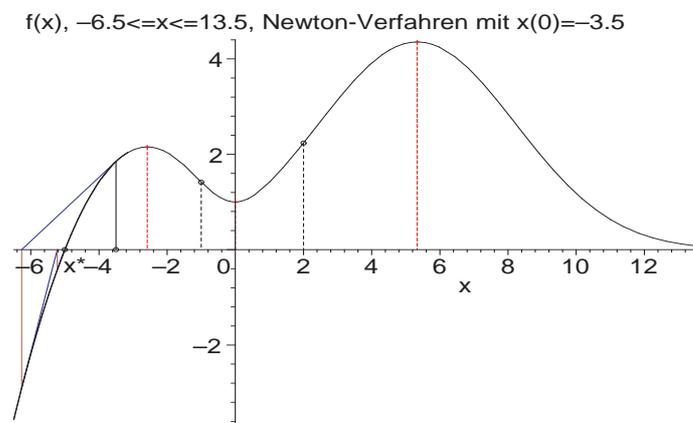


Abb. 3.19

Datei *mgv_nll1_032a.ps*
 Newton-Verfahren für
 $f(x)$, $-6.5 \leq x \leq 13.5$,
 $x^{(0)} = -3.5$

Weitere Newton-Iteration mit $x^{(0)} = -20$

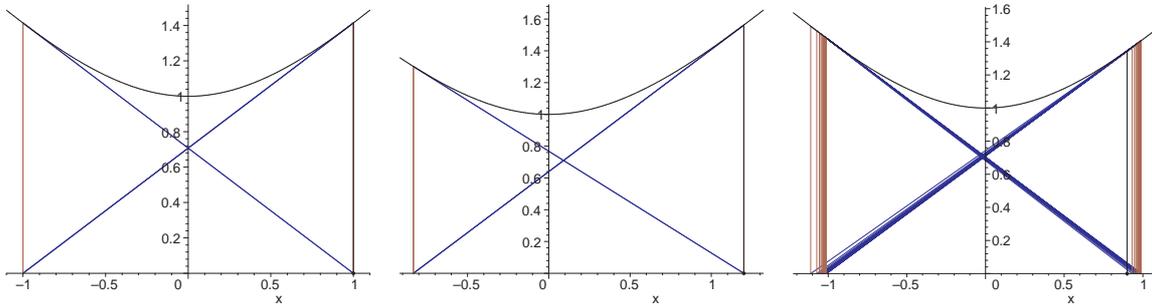
Iterationsverlauf

```
x = -2.0000000000000000000000e+01, f(x)= -1.13518824154851075519e+02
x = -1.14927278353705709423e+01, f(x)= -2.78771452593991749475e+01
x = -7.38748426136449782699e+00, f(x)= -6.48539199323218762493e+00
x = -5.60192889934146186520e+00, f(x)= -1.22347640369233848060e+00
x = -5.06120114676313644582e+00, f(x)= -1.11760525971157854010e-01
x = -5.00077253516360708900e+00, f(x)= -1.39293517015121943287e-03
x = -5.00000012620756786923e+00, f(x)= -2.27523934726438564978e-07
x = -5.0000000000000000336946e+00, f(x)= -6.07437859753681327944e-15
x = -5.0000000000000000000000e+00, f(x)= +0.00000000000000000000e+00
delta = 3.36945900000000000000e-15
```

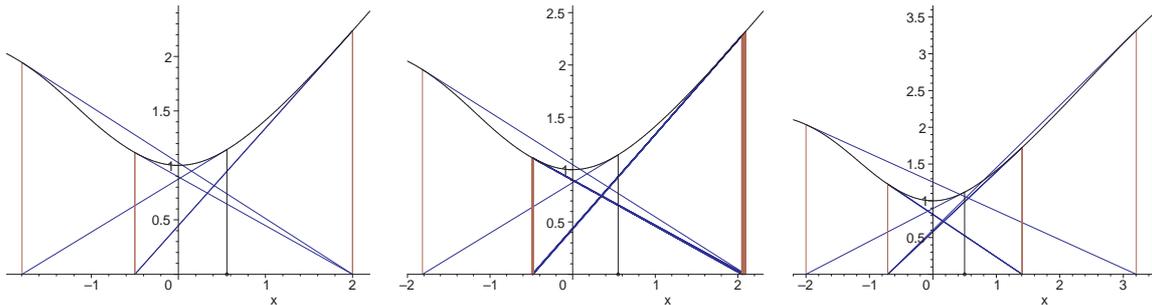
Die Startwerte $x^{(0)} = x_{min}, x_{max1}, x_{max2}$ würden gleich zur Nulldivision und zum Abbruch führen, der Startwert $x^{(0)} = x_w = 19/(29+2\sqrt{101})$ nach der ersten Iteration. Man kann leicht weitere $x^{(0)}$ finden, so dass man nach der ersten oder wenigen Anfangsiterationen an einer Extremwertstelle "landet", was den Abbruch bedeutet.

Wir testen einige Intervalle von Startwerten und geben typische Merkmale zum Iterationsverlauf an.

$$\begin{array}{lll} x^{(k)} = (-1)^k & x^{(0)} \in [1, 2], x^{(1)} \in [-1, 0] & x^{(0)} \in [\frac{3}{4}, 1], x^{(1)} \in (-2, -1) \\ x^{(k)} = (-1)^{k+1} & x^{(0)} = x^{(2)} = \dots, x^{(1)} = x^{(3)} = \dots & x^{(0)} < x^{(2)} < 1, x^{(1)} < x^{(3)} < -1 \\ & & |x^{(k+1)} - x^{(k)}| \rightarrow 2 \end{array}$$



$$\begin{array}{lll} x^{(0)} \in [0.557, \frac{3}{4}), x^{(1)} < -1 & x^{(0)} \in [0.513, 0.556], x^{(1)} < -1 & x^{(0)} \in [0.493, 0.512], x^{(1)} < -1 \\ x^{(2)} \in (1, 2), x^{(3)} \in (-1, 0) & 2 < x^{(2)}, -\frac{1}{2} < x^{(3)} < 0 & 2 < x^{(2)}, -1 < x^{(3)} < -\frac{1}{2} \\ x^{(2)} = x^{(4)} = x^{(6)} = \dots & 2 < x^{(6)} < x^{(4)} < x^{(2)} & 1 < x^{(8)} = x^{(6)} = x^{(4)} < 2 < x^{(2)} \\ x^{(3)} = x^{(5)} = x^{(7)} = \dots & -\frac{1}{2} < x^{(7)} < x^{(5)} < x^{(3)} < 0 & -1 < x^{(7)} = x^{(5)} = x^{(3)} < -\frac{1}{2} \\ |x^{(k+1)} - x^{(k)}| = c > 2, k \geq 2 & |x^{(k+1)} - x^{(k)}| \rightarrow c > 2 & |x^{(k+1)} - x^{(k)}| = c > 2, k \geq 3 \end{array}$$



$$\begin{array}{lll} x^{(0)} \in [0.488, 0.492], x^{(1)} \ll -1 & x^{(0)} \in [0.484, 0.487], x^{(1)} < -2 & x^{(0)} = 0.483, x^{(1)} < -2 \\ & \text{und } x^{(0)} = 0.482 & \\ 2 \ll x^{(2)}, x^{(3)} \in (-2, -1) & 2 \ll x^{(2)}, -2 < x^{(3)} < -1 & 2 \ll x^{(2)}, -2 < x^{(3)} < -1 \\ 0 < x^{(4)} < x^{(6)} < x^{(6)} < 1 & 1 < x^{(4)} = x^{(6)} = x^{(8)} < 2 & 2 < x^{(8)} = x^{(6)} = x^{(4)} < x^{(2)} \\ x^{(3)} < x^{(5)} < x^{(7)} < -1 & x^{(3)} < -1 < x^{(5)} = x^{(7)} < -\frac{1}{2} & x^{(3)} < -\frac{1}{2} < x^{(7)} = x^{(5)} < 0 \\ |x^{(k+1)} - x^{(k)}| \rightarrow 2 & |x^{(k+1)} - x^{(k)}| = c > 2, k \geq 4 & |x^{(k+1)} - x^{(k)}| \rightarrow c > 2 \end{array}$$

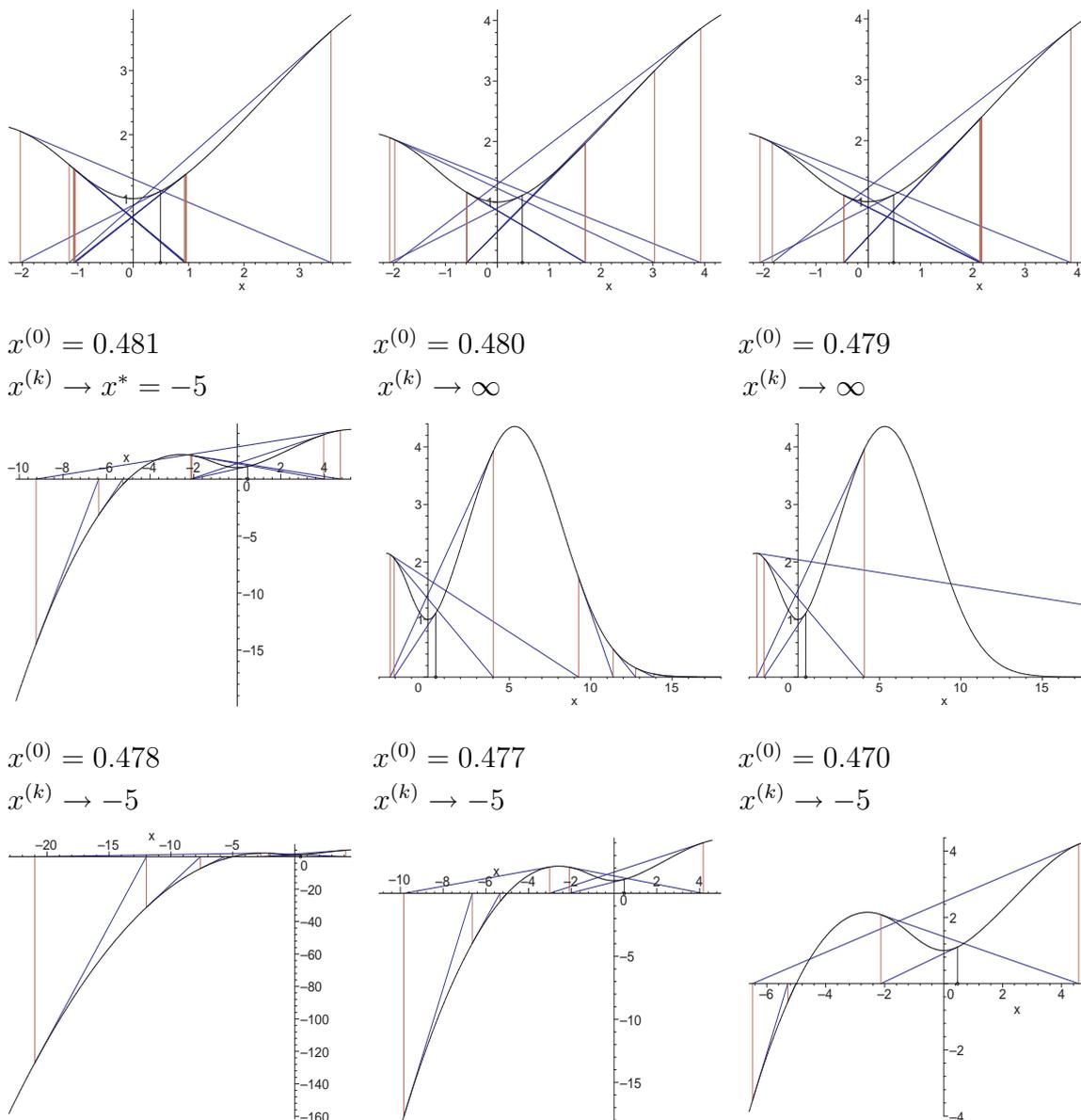


Abb. 3.20 Dateien *mgv_nl1_032b.p.ps*, Newton-Verfahren für $f(x) = 0$ mit verschiedenen Startwerten $x^{(0)}$ und Konvergenzverhalten

Betrachten wir noch einmal den Startwert

$$x^{(0)} = x_w = 19 / (29 + 2\sqrt{101}) = 0.386\ 967\ 337\ 293\dots$$

sowie solche aus seiner Umgebung.

Zunächst ist bei exakter Rechnung $x^{(1)} = -1/x^{(0)} = -1/x_w = x_{max1} = -2.584\ 197$ und $f'(x^{(1)}) = 0$. Das führt entweder zum Abbruch wegen Nulldivision oder zur Fehlermeldung wegen zu vieler Rekursionen.

```
in Newton3 numeric exception: division by zero
Error, (in simpl/Im) too many levels of recursion
```

Konvertiert man als Startwert $x^{(0)}$ die Dezimalzahl 0.3869673372 in den Bruch $17108627/44212070$ und rechnet dann exakt, werden die Iterierten als Brüche sowie die Rechenzeiten immer länger. Die numerische Rechnung bei `Digits=22` mit dem dezimalen Startwert $x^{(0)} = 0.3869673372 < x_w$ liefert Konvergenz gegen $x^* = -5$.

Iterationsverlauf

$x(0) = +3.86967337200000000000e-01,$	$f(x(0)) = +1.07226103168009351278e+00$
$x(1) = -2.58419743442884047114e+00,$	$f(x(1)) = +2.15341105079272706131e+00$
$x(2) = -4.87677346935637192771e+09,$	$f(x(2)) = -9.16298510373161811674e+18$
$x(3) = -2.43838673602029122754e+09,$	$f(x(3)) = -2.29074627593290452869e+18$
$x(4) = -1.21919336935225087798e+09,$	$f(x(4)) = -5.72686568983226131677e+17$
$x(5) = -6.09596686018230704257e+08,$	$f(x(5)) = -1.43171642245806532424e+17$
$x(6) = -3.04798344351220619503e+08,$	$f(x(6)) = -3.57929105614516326108e+16$
$x(7) = -1.52399173517715581343e+08,$	$f(x(7)) = -8.94822764036290765748e+15$
$x(8) = -7.61995881009630706974e+07,$	$f(x(8)) = -2.23705691009072641917e+15$
$x(9) = -3.80997953925868322422e+07,$	$f(x(9)) = -5.59264227522681109590e+14$
$x(10) = -1.90498990383987467501e+07,$	$f(x(10)) = -1.39816056880669782195e+14$
$x(15) = -5.95311945281784163078e+05,$	$f(x(15)) = -1.36539118046869458947e+11$
$x(20) = -1.86060987110344178576e+04,$	$f(x(20)) = -1.33338981808008964378e+08$
$x(25) = -5.84043859294435580259e+02,$	$f(x(25)) = -1.30213189412327053961e+05$
$x(30) = -2.09464564207603128692e+01,$	$f(x(30)) = -1.26493169719447277946e+02$
$x(33) = -5.68522439075646241293e+00,$	$f(x(33)) = -1.41459920593372656521e+00$
$x(34) = -5.07715735139397519666e+00,$	$f(x(34)) = -1.41368082541133279265e-01$
$x(35) = -5.00121992483966717019e+00,$	$f(x(35)) = -2.19981832323674918405e-03$
$x(36) = -5.00000031465433026133e+00,$	$f(x(36)) = -5.67251198659087019975e-07$
$x(37) = -5.00000000000002094386e+00,$	$f(x(37)) = -3.77570787652941429464e-14$
$x(38) = -5.00000000000000000000e+00,$	$f(x(38)) = +0.00000000000000000000e+00$
$\text{delta} = 0.00000000000000000000e+00$	

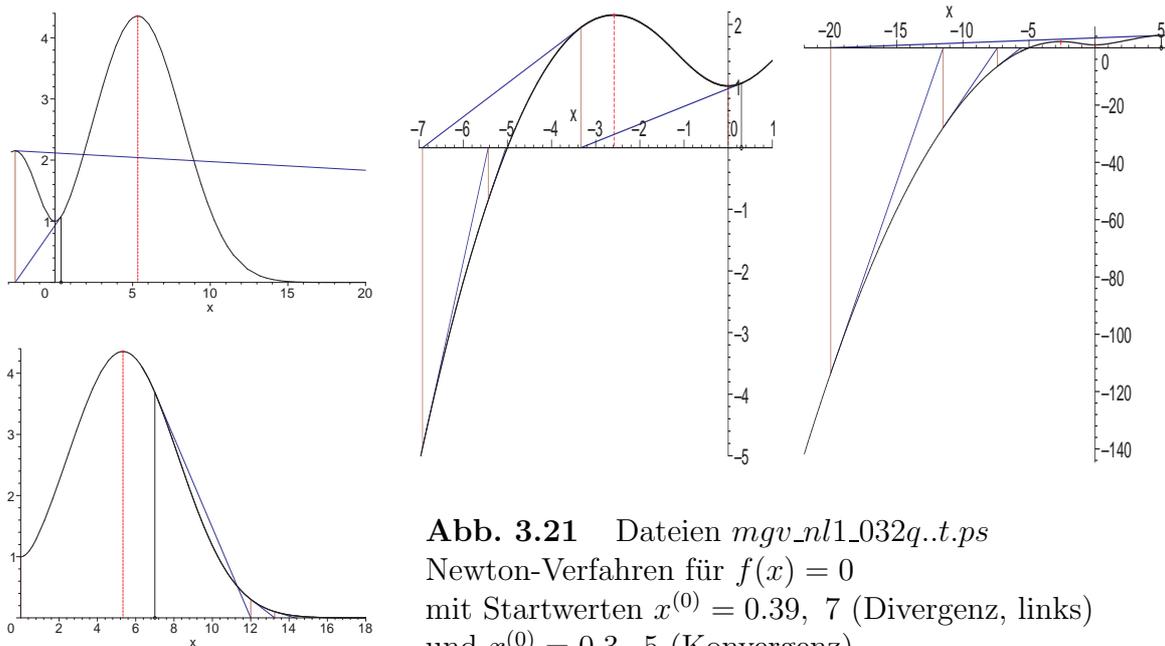


Abb. 3.21 Dateien *mgv_n11_032q..t.ps*
 Newton-Verfahren für $f(x) = 0$
 mit Startwerten $x^{(0)} = 0.39, 7$ (Divergenz, links)
 und $x^{(0)} = 0.3, 5$ (Konvergenz)

Eine ähnliche Vielfalt des Lösungsverhaltens bekommt man zum MGW für die Bestimmung der absoluten Minimumstelle $x^* = -5$ von $h(x)$.

Das MGW ist das Newton-Verfahren zu $h(x)$ und ist als Maple-Prozedur mit grafischer und Datei-Ausgabe sowie zusätzlichen Tests zur Durchführbarkeit implementiert.

Normalfall mit Konvergenz

```
> # Parameter: Funktion, Startwert, maximale Iterationsanzahl
> qh1:=Newton2(h,-3.5,6):
  display(qh1,ps2,ps3,ps4,ps5,ps6,ps7,view=[-6.5..13.5,-1..19],
    title='h(x), -6.5<=x<=13.5, MGW mit x(0)=-3.5');
```

Iterationsverlauf

```
x = -3.500000000000000000000000e+00, h(x)= 3.4233398437500000000000e+00
x = -4.88447867298578199052e+00, h(x)= 4.12784887609207181079e-02
x = -4.94372256970007116131e+00, h(x)= 1.00496450784449981531e-02
x = -4.97220435474353750265e+00, h(x)= 2.48150383452869193337e-03
x = -4.98618485663495809667e+00, h(x)= 6.16669004092825471158e-04
x = -4.99311273268772661414e+00, h(x)= 1.53713095940128319327e-04
x = -4.99656139795891573608e+00, h(x)= 3.83720640263053622768e-05
delta = 3.44866527118912194300e-03 # |x(k+1)-x(k)|
```

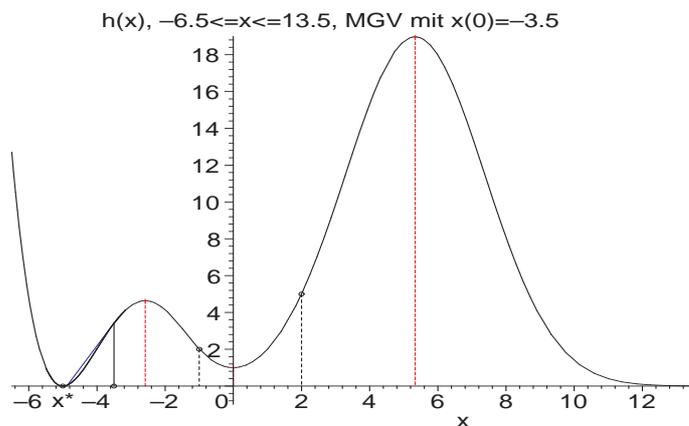


Abb. 3.22

Datei *mgv_nl1_033a.ps*

MGW für

$h(x)$, $-6.5 \leq x \leq 13.5$,

$x^{(0)} = -3.5$

Weitere MGW-Iteration mit $x^{(0)} = -20$

Iterationsverlauf

```
x(0) = -2.0000000000000000000000e+01, h(x(0)) = 1.28865234375000000000e+04
x(10) = -5.13696201422073054996e+00, h(x(10)) = 6.45503315072256774879e-02
x(20) = -5.00014150016893884705e+00, h(x(20)) = 6.50763635340734020886e-08
x(30) = -5.00000013819202308698e+00, h(x(30)) = 6.20653681745427534444e-14
x(40) = -5.00000000013495315543e+00, h(x(40)) = 5.91901510232947542152e-20
x(48) = -5.00000000000052716076e+00, h(x(48)) = 9.03170031078658445609e-25
x(49) = -5.00000000000026358038e+00, h(x(49)) = 2.25792507769639432201e-25
x(50) = -5.00000000000013179019e+00, h(x(50)) = 5.64481269424067106501e-26
delta = 1.31790191000000000000e-13
```

Man erkennt sehr schön die lineare Konvergenz bei Halbierung des Fehlers $|x^* - x^{(k)}|$ in jedem Schritt.

Mit dem Startwert $x^{(0)} = \pm 1$ folgt der Abbruch der Iteration

$$x^{(k+1)} = \frac{1}{2} \left(x^{(k)} - \frac{1}{x^{(k)}} \right).$$

bei $x^{(1)} = 0$ mit der Fehlermeldung wegen Nulldivision.

in Newton2 numeric exception: division by zero

Die exakte Iteration mit dem Startwert $x^{(0)} = \pm\sqrt{3}/3$ liefert die Folge $x^{(k)} = \pm(-1)^k\sqrt{3}/3$, oszilliert also zwischen den Werten $\sqrt{3}/3$ und $-\sqrt{3}/3$. Rechnet man numerisch mit sehr hoher Genauigkeit, z. B. mit $x^{(0)} = 0.577\,350\,269\,189\,625\,764\,509 \approx \sqrt{3}/3$, so bleibt diese Situation erhalten. Sobald jedoch der Startwert knapp daneben liegt, also z. B. $x^{(0)} = 0.577\,350\,269\,189\,625\,764\,510$ ist, macht sich ein aufschaukelnder Prozess bemerkbar, wo irgendwann mal die Iterierte "nach außen driftet". Sie kann dann entweder wie hier in der gewünschten Umgebung von x^* landen und konvergieren oder auch im ungünstigen positiven Bereich.

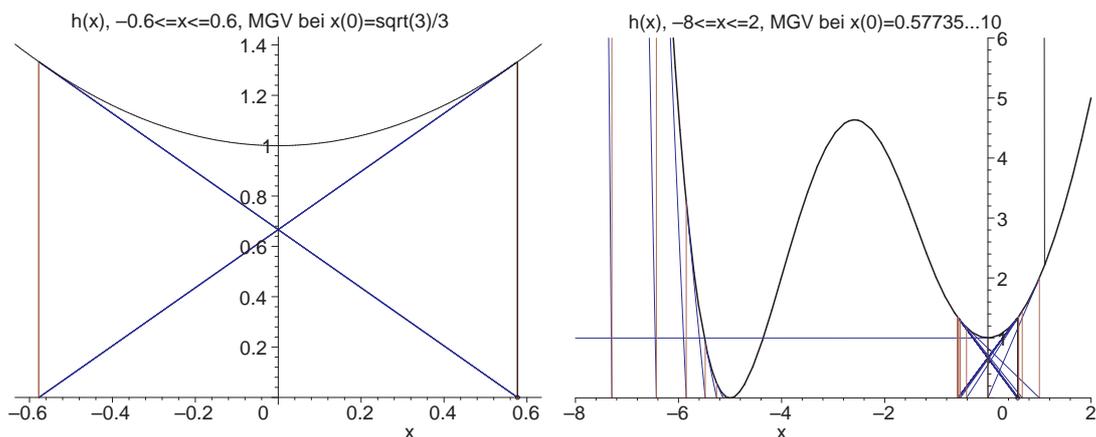
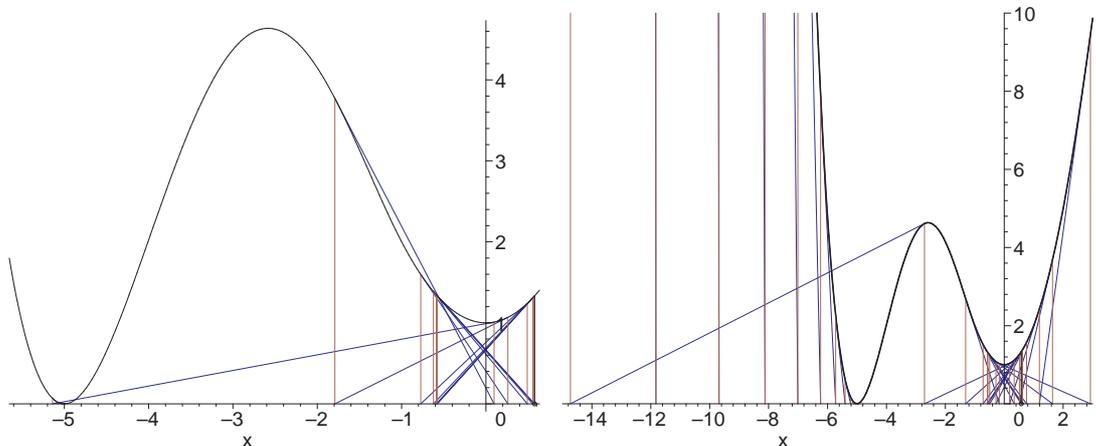


Abb. 3.23 Dateien *mgv_nl1_033b, c.ps*, MGv für $h(x)$,
 $x^{(0)} = \sqrt{3}/3$ und $x^{(0)} = 0.577\,350\,269\,189\,625\,764\,510$ (rechts)

Wir entfernen uns mit dem Startwert $x^{(0)}$ noch etwas weiter weg von $\sqrt{3}/3$.



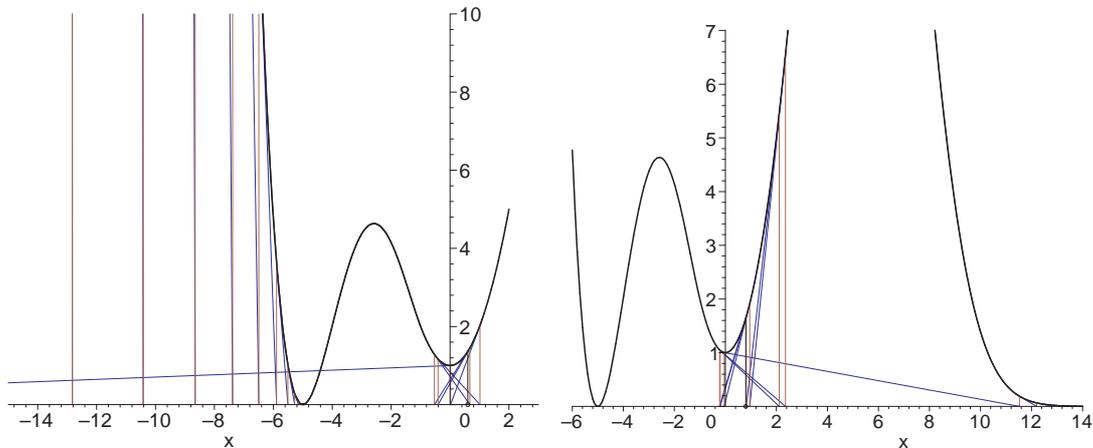


Abb. 3.24 Dateien *mgv_nl1_033d..g.ps*, MGV für $h(x)$, Startwerte $x^{(0)} = 0.577, 0.578, 0.6$ (Konvergenz gegen x^*), 0.8 (Divergenz)

MGV-Iteration mit $x^{(0)} = 0.8$ und langsamer Divergenz gegen $x^{(k)} \rightarrow \infty$.

Für hinreichend großes k wird die Gleitpunktarithmetik wegen Nenner gegen Null versagen.

Iterationsverlauf

$x(0) = +8.00000000000000000000e-01,$	$h(x) = 1.64000000000000000000e+00$
$x(1) = -2.25000000000000000000e-01,$	$h(x) = 1.05062500000000000000e+00$
$x(2) = +2.10972222222222222222e+00,$	$h(x) = 5.45084702639347969792e+00$
$x(3) = +8.17175905979223514768e-01,$	$h(x) = 1.66777646131296474971e+00$
$x(4) = -2.03275412463936465777e-01,$	$h(x) = 1.04132089331238349666e+00$
$x(5) = +2.35807935418086497382e+00,$	$h(x) = 6.55688011048484886203e+00$
$x(6) = +9.57638687312625226953e-01,$	$h(x) = 1.91707185543784799282e+00$
$x(7) = -4.32982426779713846760e-02,$	$h(x) = 1.00187473781900050262e+00$
$x(8) = +1.15261636552377949974e+01,$	$h(x) = 1.90259482391399461382e-01$
$x(9) = +1.21989847160585450078e+01,$	$h(x) = 6.62918030728981009244e-02$
$x(10) = +1.28058705648143515204e+01,$	$h(x) = 2.33330078417617668656e-02$
$x(20) = +1.72141278609927484419e+01,$	$h(x) = 8.35176975999856829360e-07$
$x(30) = +2.03491281052331023085e+01,$	$h(x) = 3.36799008632128751854e-11$
$x(40) = +2.29251620408075577348e+01,$	$h(x) = 1.41178543301711427443e-15$
$x(50) = +2.51653331417405565945e+01,$	$h(x) = 6.03446461676974798211e-20$
$x(60) = +2.71747074777159561354e+01,$	$h(x) = 2.60996950102643013767e-24$
$x(70) = +2.90127970116613830252e+01,$	$h(x) = 1.13781964437239157089e-28$
$x(80) = +3.07172052120118100968e+01,$	$h(x) = 4.98864898287842785249e-33$
$x(90) = +3.23134960175677598788e+01,$	$h(x) = 2.19661660645612065220e-37$
$x(98) = +3.35251653477283746187e+01,$	$h(x) = 7.20829884175114176517e-41$
$x(99) = +3.36729622934727704966e+01,$	$h(x) = 2.64484300118429350113e-41$
$x(100) = +3.38199884479974896430e+01,$	$h(x) = 9.70462714375855792185e-42$
$\text{delta} = 1.47026154524719146380e-01$	

Die Darstellung weiterer Iterationverläufe überlassen wir dem Leser. ◀

Man kann sich nun vorstellen, wie komplex die Situation im mehrdimensionalen nicht-linearen Fall werden kann.

3.2.3 Beispiele im \mathbb{R}^2

Beispiel 3.11 Sei

$$\begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} 10(x_2 - x_1^2) \\ 1 - x_1 \end{pmatrix}$$

und damit

$$h(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

sowie

$$h'(x_1, x_2) = \begin{pmatrix} -400(x_2 - x_1^2)x_1 - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{pmatrix}.$$

Die einzige Lösung des Problems ist $x^* = (1, 1)^T$.

Als Startvektor wählen wir $x^{(0)} = (-1, 1.5)^T$. Das MGV konvergiert.

$$\begin{aligned} x^{(10)} &= [-0.96087129263812713076539165, 1.4236591705474114853063300] \\ x^{(50)} &= [0.32007079870449812744445303, 0.0793646834640996769384751] \\ x^{(100)} &= [0.70092178006561393681945016, 0.5018593394242495943717640] \\ x^{(250)} &= [0.95477768560158496378344341, 0.9112267802054077988093529] \\ x^{(500)} &= [0.99731674586329102725123922, 0.9915872514692962463826948] \\ x^{(750)} &= [0.99979227134505630061132036, 0.9995720547135056703675339] \\ x^{(1000)} &= [0.99999482271707304062975279, 0.9999896753402655837624423] \end{aligned}$$

MGV - Iterationsverlauf, $x(0)=[-1, 1.5]$

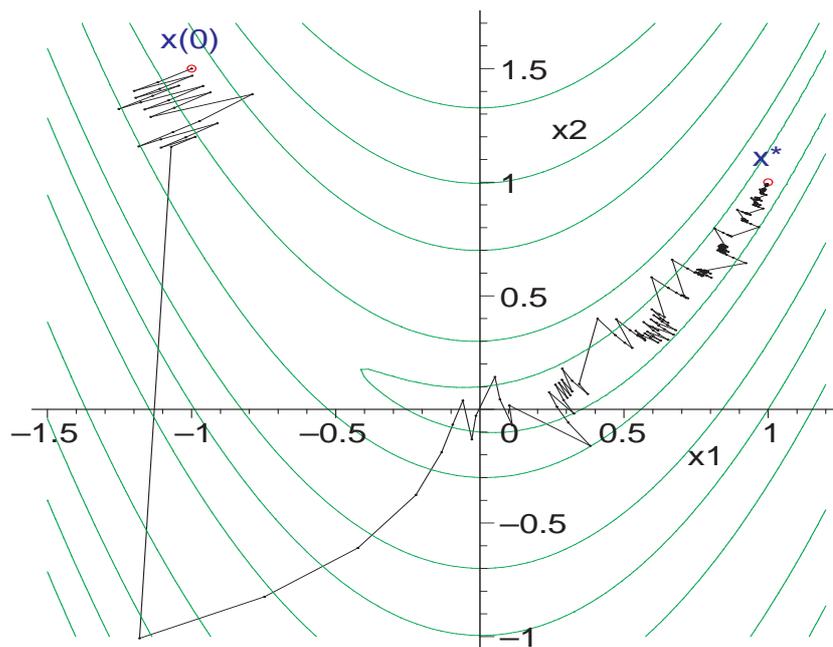


Abb. 3.25

Datei

mgv_non_lin_01a.ps

Konvergenz

des MGV ◀

Beispiel 3.12 Sei

$$\begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} x_1 - x_2 \\ 1 + x_1^5 \end{pmatrix}$$

und damit

$$h(x_1, x_2) = (x_1 - x_2)^2 + (1 + x_1^5)^2$$

sowie

$$h'(x_1, x_2) = \begin{pmatrix} 2(x_1 - x_2) + 10(1 + x_1^5)x_1^4 \\ -2(x_1 - x_2) \end{pmatrix}.$$

Die einzige Lösung des Problems ist $x^* = (-1, -1)$.

Als Startvektor wählen wir $x^{(0)} = (-1, 1.5)^T$. Das MGW konvergiert.

$$\begin{aligned} x^{(10)} &= [-0.7547152305407388789239313, 0.41780116776678951681228233] \\ x^{(20)} &= [-0.9712046435170322299474998, -0.92159689383380393573802420] \\ x^{(50)} &= [-0.9997748484325499712791636, -0.99574133399733965336221563] \\ x^{(100)} &= [-1.000000777297703935811588, -0.99999694383118638631484663] \\ x^{(250)} &= [-0.99999999999999996983798152, -0.9999999999999999784320879364] \\ x^{(500)} &= [-1.000000000000000000000000000000, -0.999999999999999999999999999997] \end{aligned}$$

MGW - Iterationsverlauf, $x(0)=[-1, 1.5]$

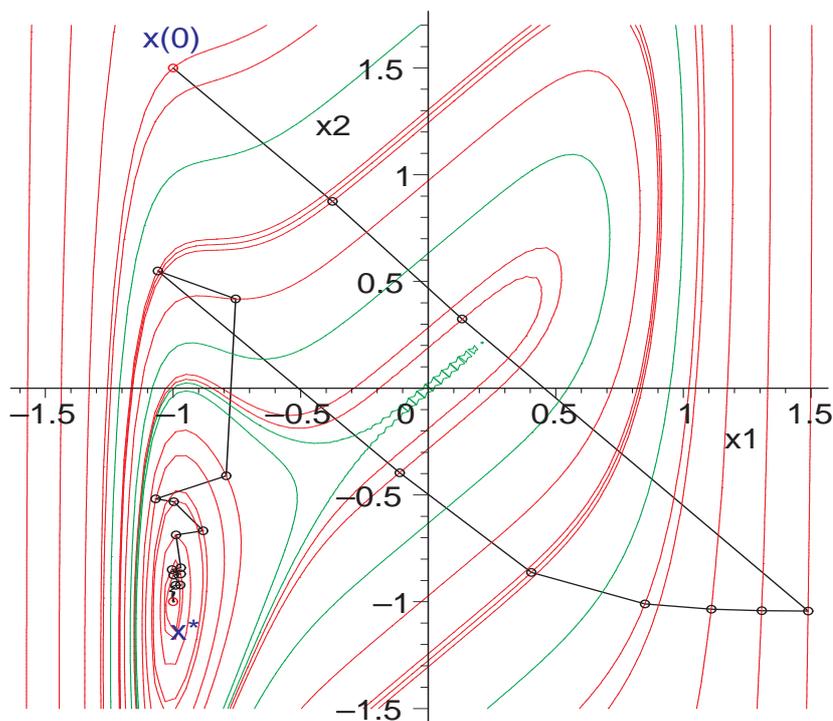


Abb. 3.26

Datei
mgv_non_lin_02a.ps
 Konvergenz
 des MGW ◀

Beispiel 3.13 Sei

$$\begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} x_2^3 - 3 \\ x_1^3 + 1 \end{pmatrix}$$

und damit

$$h(x_1, x_2) = (x_2^3 - 3)^2 + (x_1^3 + 1)^2$$

sowie

$$h'(x_1, x_2) = \begin{pmatrix} 6(x_1^3 + 1)x_1^2 \\ 6(x_2^3 - 3)x_2^2 \end{pmatrix}.$$

Die einzige Lösung des Problems ist $x^* = (-1, \sqrt[3]{3})^T = (-1, 1.442\,249\,570\,307\,408)^T$.
Als Startvektor wählen wir $x^{(0)} = (1, -1.5)^T$. Das MGV konvergiert.

$$\begin{aligned} x^{(1)} &= [0.9290550974576731524635630, -0.9911920270792496403246158] \\ x^{(2)} &= [0.6496277910294641193348961, -0.2897757422290427650878214] \\ x^{(5)} &= [-1.4749612615022241896181215, 1.0597286090744924912120888] \\ x^{(10)} &= [-1.0388510105812154758939702, 1.4461398121643650673575224] \\ x^{(20)} &= [-1.0002766174262617343769744, 1.4422937475165816639138331] \\ x^{(50)} &= [-1.0000000002032419960308214, 1.4422495703399241551905408] \\ x^{(100)} &= [-1.000000000000000000000000121871, 1.4422495703074083823235881] \\ x^{(250)} &= [-1.000000000000000000000000000001, 1.4422495703074083823216383] \end{aligned}$$

MGV - Iterationsverlauf, $x(0)=[1, -1.5]$

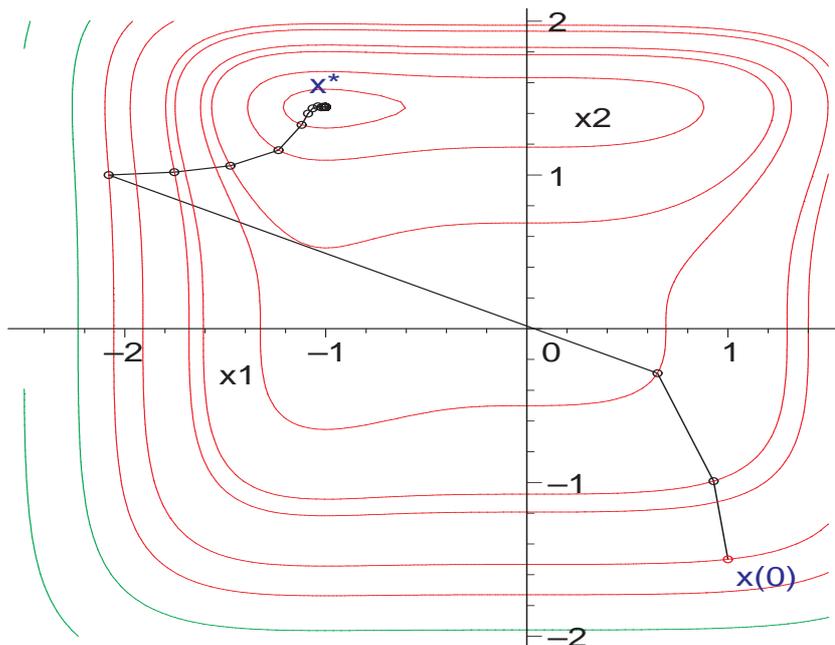


Abb. 3.27

Datei
mgv_non_lin_03a.ps
Konvergenz
des MGV ◀

Beispiel 3.14 Sei

$$\begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} 4x_1 - \sin(x_1 + x_2) \\ -3x_2 + \cos(x_1 - x_2) \end{pmatrix}$$

und damit

$$h(x_1, x_2) = (4x_1 - \sin(x_1 + x_2))^2 + (3x_2 - \cos(x_1 - x_2))^2$$

sowie

$$h'(x_1, x_2) = \begin{pmatrix} 2(4x_1 - \sin(x_1 + x_2))(4 - \cos(x_1 + x_2)) + 2(\cos(-x_1 + x_2) - 3x_2) \sin(-x_1 + x_2) \\ -2(4x_1 - \sin(x_1 + x_2)) \cos(x_1 + x_2) + 2(\cos(-x_1 + x_2) - 3x_2)(\sin(x_1 - x_2) - 3) \end{pmatrix}.$$

Die einzige Lösung des Problems ist $x^* = (0.104\,050\,629\,952, 0.325\,214\,281\,787)^T$.

Als Startvektoren wählen wir $x^{(0)} = (0.3, 1)^T$, $(1, 1)^T$, $(-1, -1)^T$.

Das MGW konvergiert.

$$\begin{aligned} x^{(1)} &= [0.000000000000000000, 0.1666666666666666667] \\ x^{(2)} &= [0.02675439238366732465, 0.25442447528052415889] \\ x^{(5)} &= [0.09382350829039410241, 0.31711365714042125923] \\ x^{(10)} &= [0.10373300395820723306, 0.32496053114090883904] \\ x^{(20)} &= [0.10405031983321070706, 0.32521403396499495092] \\ x^{(40)} &= [0.10405062995185679971, 0.32521428178717865153] \\ x^{(60)} &= [0.10405062995215255184, 0.32521428178741499320] \\ x^{(70)} &= [0.10405062995215255212, 0.32521428178741499343] \\ x^{(80)} &= [0.10405062995215255212, 0.32521428178741499343] \end{aligned}$$

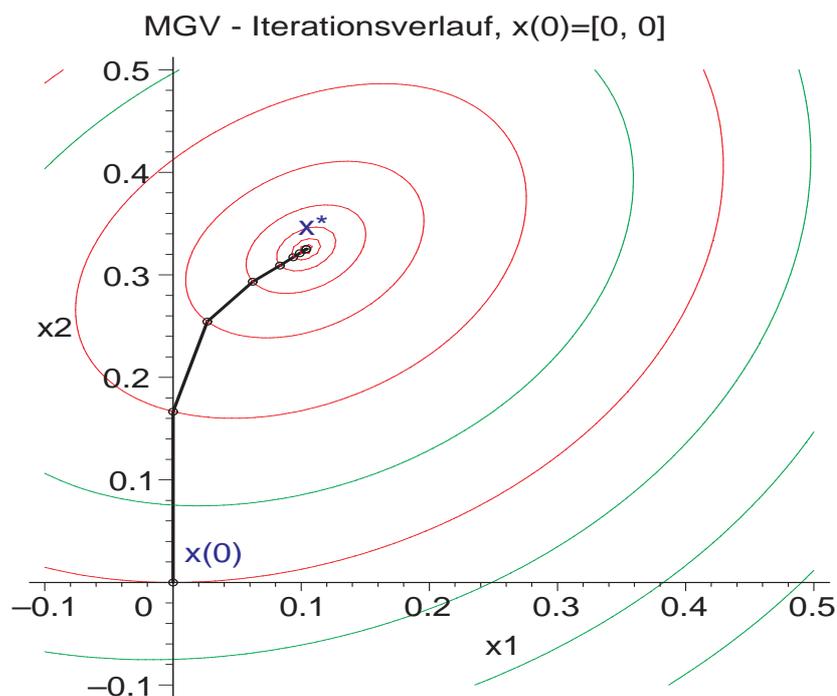


Abb. 3.28

Datei

mgv_non_lin_04a.ps

Konvergenz

des MGW,

$x^{(0)} = (0, 0)^T$

Sieht man das Höhenlinienbild von $h(x)$, so ist die Konvergenz und der Iterationsverlauf auch bei anderen Startpunkten qualitativ ähnlich.

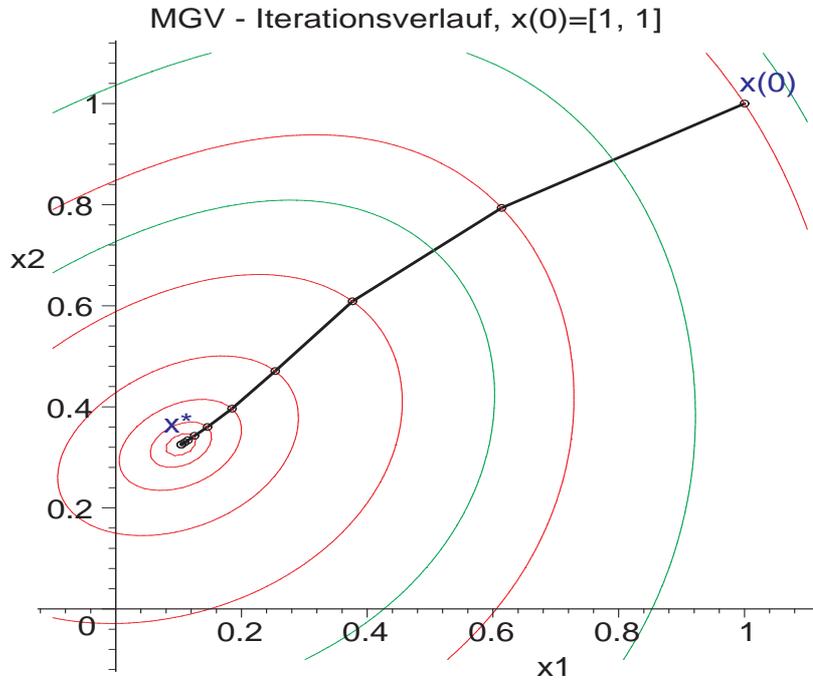


Abb. 3.29

Datei

mgv_non_lin_04b.ps

Konvergenz

des MGV,
 $x^{(0)} = (1, 1)^T$

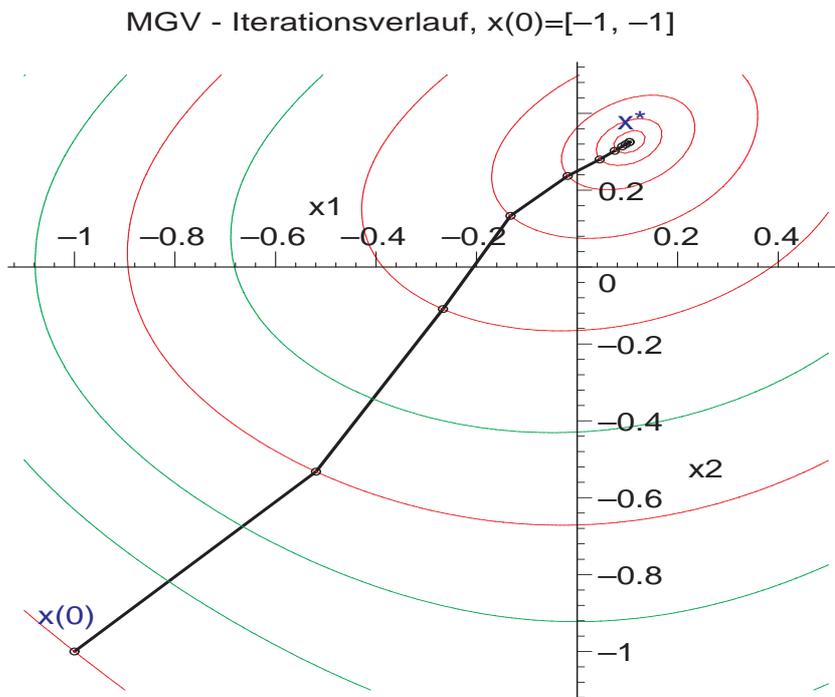


Abb. 3.30

Datei

mgv_non_lin_04c.ps

Konvergenz

des MGV,
 $x^{(0)} = (-1, -1)^T$ ◀

Beispiel 3.15 Sei

$$\begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} x_1 - 0.1x_1^2 - \sin(x_2) \\ x_2 - 0.1x_2^2 - \cos(x_1) \end{pmatrix}$$

und damit

$$h(x_1, x_2) = (x_1 - 0.1x_1^2 - \sin(x_2))^2 + (x_2 - 0.1x_2^2 - \cos(x_1))^2$$

sowie

$$h'(x_1, x_2) = \begin{pmatrix} 2(x_1 - 0.1x_1^2 - \sin(x_2))(1 - 0.2x_1) + 2(x_2 - 0.1x_2^2 - \cos(x_1)) \sin(x_1) \\ -2(x_1 - 0.1x_1^2 - \sin(x_2)) \cos(x_2) + 2(x_2 - 0.1x_2^2 - \cos(x_1))(1 - 0.2x_2) \end{pmatrix}.$$

Die vier Lösungen des Problems sind

$$x_1^* = (0.764\,070\,550\,813, 0.783\,396\,774\,300)^T, \quad x_2^* = (10.452\,424\,867\,167, -0.492\,572\,004\,614)^T, \\ x_3^* = (0.437\,769\,253\,711, 8.992\,869\,176\,184)^T, \quad x_4^* = (10.704\,529\,325\,694, 10.279\,160\,347\,170)^T.$$

Als Startvektoren wählen wir $x^{(0)} = (1, 1)^T, (2, 2)^T, (10, 2)^T, (2, 7.5)^T, (8.3, 10)$ und $(5, 5)^T, (8, 10)^T$. Das MGW konvergiert.

$$\begin{aligned} x^{(0)} &= [1.000000000000000000, 1.000000000000000000] \\ x^{(1)} &= [0.87639099461689987674, 0.90941140670907994030] \\ x^{(2)} &= [0.81667227848482806340, 0.85194822394364174365] \\ x^{(3)} &= [0.78917552147978884172, 0.81893699820876868973] \\ x^{(4)} &= [0.77637104900172687582, 0.80138927969985886986] \\ x^{(5)} &= [0.77020670355477469632, 0.79240954171444713851] \\ x^{(6)} &= [0.76716148048370147396, 0.78788938641001072961] \\ x^{(10)} &= [0.76427468337859665428, 0.78367007940840024063] \\ x^{(20)} &= [0.76407077520916524953, 0.78339702116092842471] \\ x^{(30)} &= [0.76407055104931519717, 0.78339677452485564897] \\ x^{(38)} &= [0.76407055081370149488, 0.78339677430131156436] \\ x^{(39)} &= [0.76407055081322182608, 0.78339677430089230048] \\ x^{(40)} &= [0.76407055081298091315, 0.78339677430068389052] \end{aligned}$$

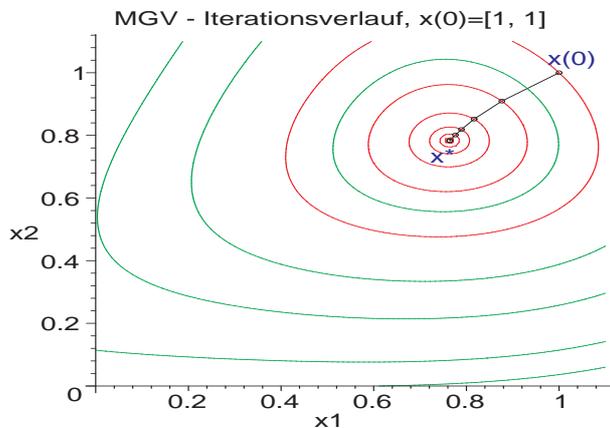


Abb. 3.31

Datei
mgv_non_lin_05a.ps
Konvergenz
des MGW,
 $x^{(0)} = (1, 1)^T$

Das lokale Bild von moderaten Höhenlinien täuscht über seine komplizierte Struktur in einem größeren Bereich hinweg.

Wir haben 4 globale Minima der Funktion $h(x)$ im Bereich $[0, 11] \times [0, 11]$.

Außerhalb dieses Quadrats gibt es keine weiteren u.a. wegen $|\sin(x_i)|, |\cos(x_i)| \leq 1$ und nicht verschwindenden Größen $|x_i - 0.1 x_i^2| = |x_i(1 - 0.1 x_i)|$.

Aber es gibt im genannten Quadrat weitere 5 lokale Extrema und damit auch Sattelpunkte.

- 3 Minima \square : $(7.371, 1.145)^T$, $h(7.371, 1.145) = 1.357$,
 $(7.023, 8.255)^T$, $h(7.023, 8.255) = 1.862$,
 $(11.714, 4.552)^T$, $h(11.714, 4.552) = 4.360$,
- 2 Maxima $+$: $(3.440, 4.763)^T$, $h(3.440, 4.763) = 22.501$,
 $(8.957, 4.787)^T$, $h(8.957, 4.787) = 15.209$.

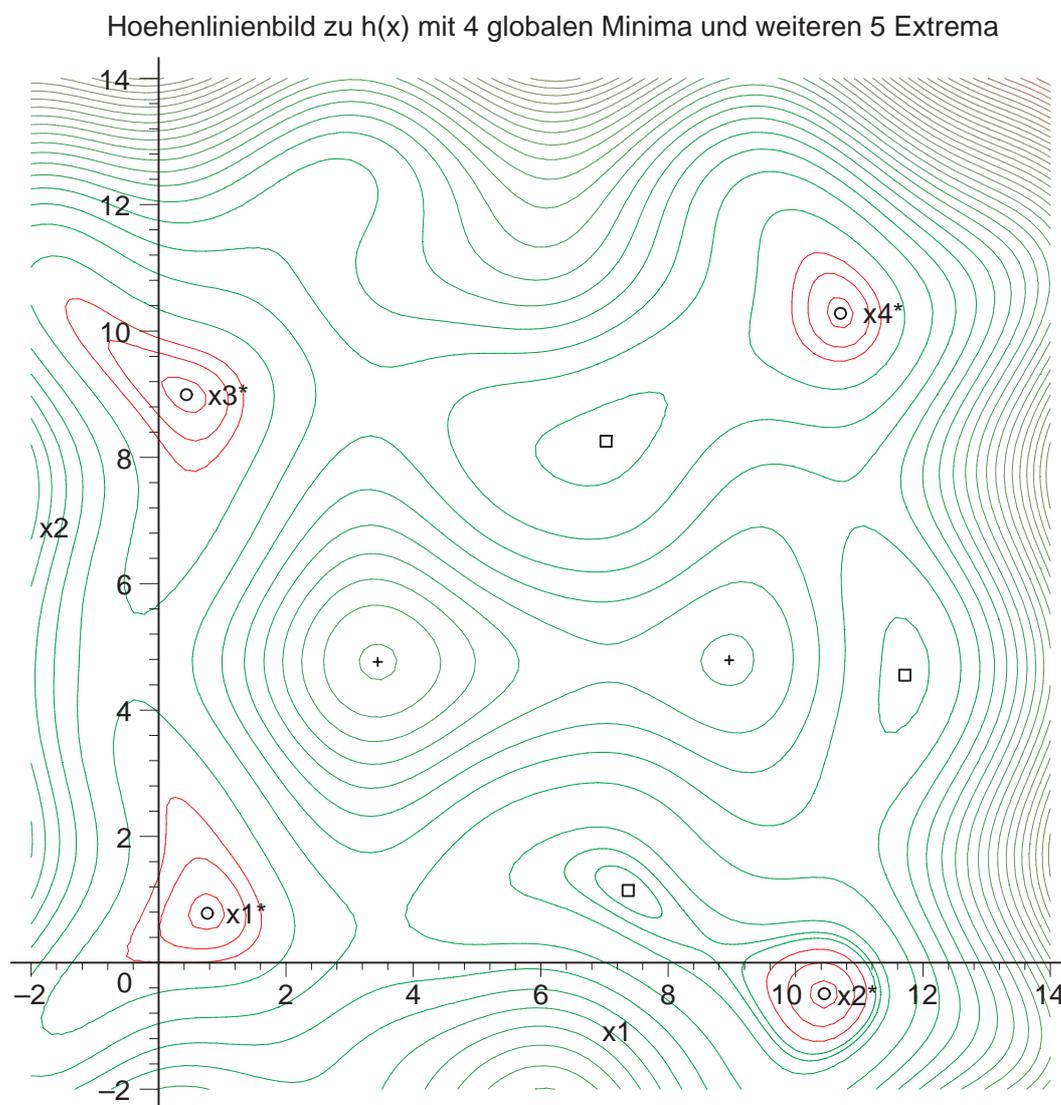


Abb. 3.32 Datei *mgv_non_lin_05b.ps*

Höhenlinienbild zu $h(x)$ mit 4 globalen Minima und weiteren 5 Extrema

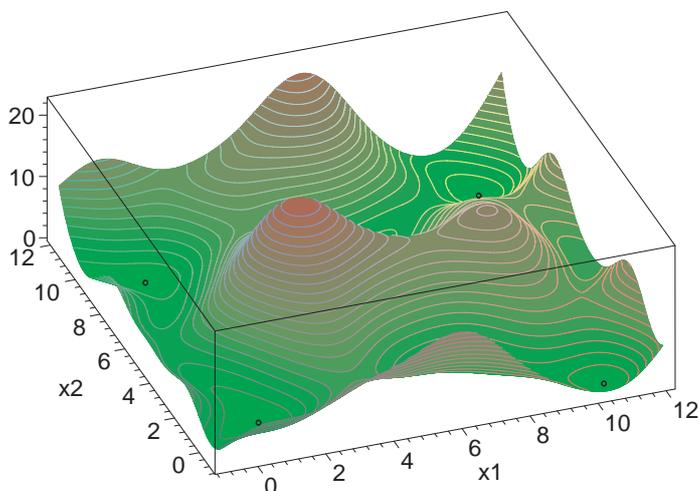


Abb. 3.33
 Datei *mgv_non_lin_05c.ps*
 3D-Höhenlinienbild
 zu $h(x)$ mit
 $x \in [-1, 12] \times [-1, 12]$

Wir betrachten den lokalen Iterationsverlauf des MGV zu den 4 globalen Minima.

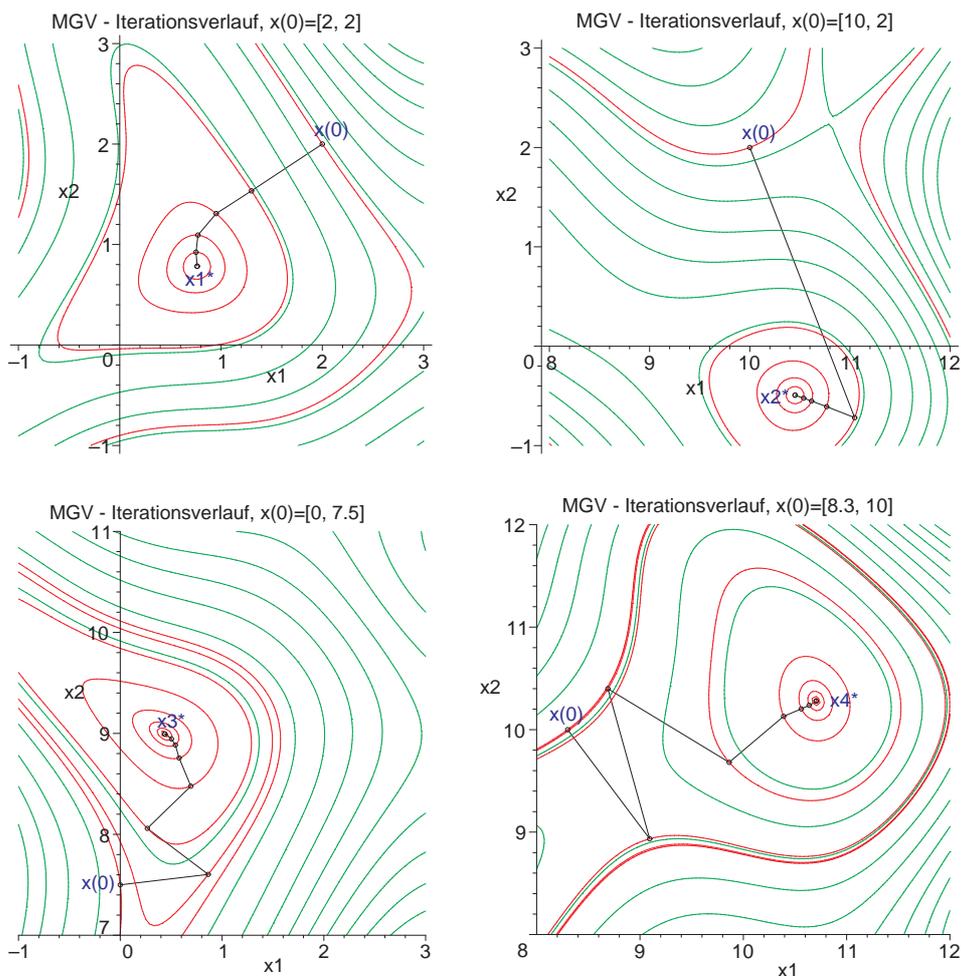


Abb. 3.34 Dateien *mgv_non_lin_05d..g.ps*
 Konvergenz des MGV, $x^{(0)} = (2, 2)^T, (10, 2)^T, (2, 7.5)^T, (8.3, 10)^T$

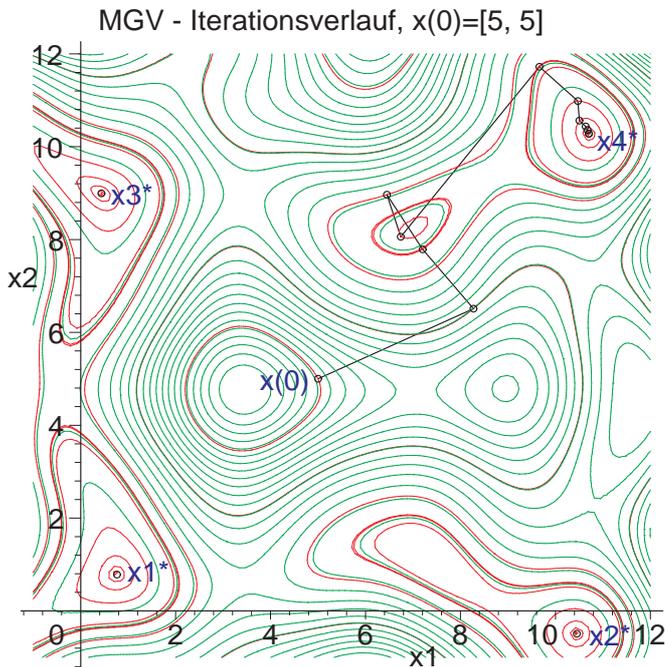


Abb. 3.35

Datei *mgv_non_lin_05h.ps*

Konvergenz

des MGV,

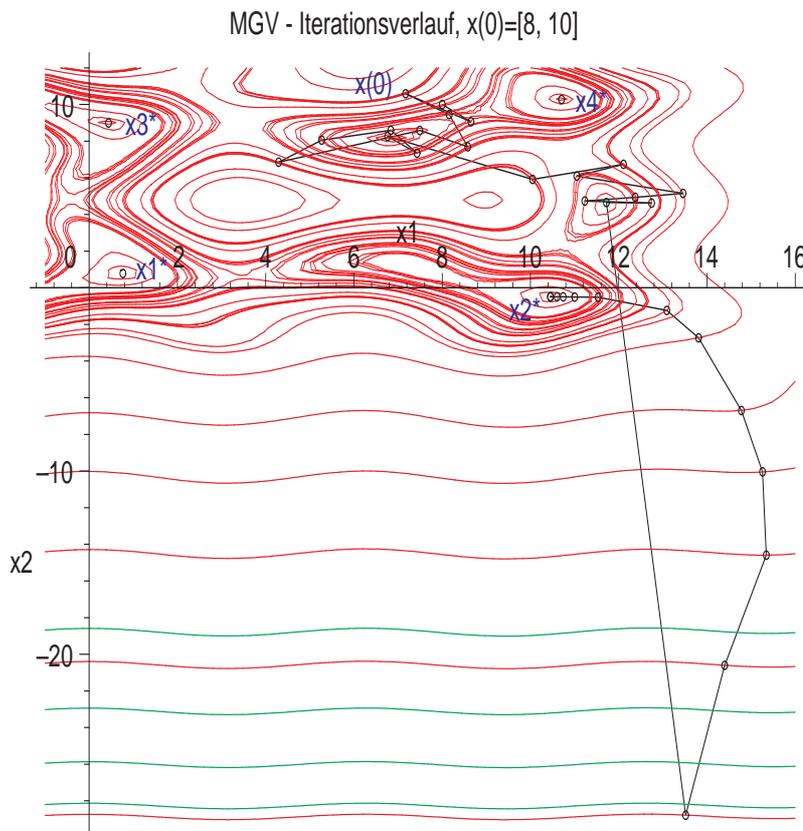
 $x^{(0)} = (5, 5)^T$ ◀

Abb. 3.36

Datei

mgv_non_lin_05i.ps

Konvergenz

des MGV,

 $x^{(0)} = (8, 10)^T$

Durch die wellige Oberfläche der Funktion $h(x)$ können die Iterationen für weiter entfernte Startvektoren recht eigenwillige Verläufe annehmen. ◀

Beispiel 3.16 Sei

$$\begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} \frac{1}{4}(x_1 \sin(x_2) + x_2) - x_1 \\ \arctan\left(\frac{4}{x_1 + x_2}\right) - x_2 \end{pmatrix}$$

und damit

$$h(x_1, x_2) = \left(\frac{1}{4}(x_1 \sin(x_2) + x_2) - x_1 \right)^2 + \left(\arctan\left(\frac{4}{x_1 + x_2}\right) - x_2 \right)^2$$

sowie

$$h'(x_1, x_2) = \begin{pmatrix} 2z_1\left(\frac{1}{4}\sin(x_2) - 1\right) - 8\frac{z_2}{z_3} \\ 2z_1\frac{1}{4}(x_1 \cos(x_2) + 1) - 2z_2\left(\frac{4}{z_3} + 1\right) \end{pmatrix},$$

wobei

$$z_1 = \frac{1}{4}x_1 \sin(x_2) + \frac{1}{4}x_2 - x_1, \quad z_2 = \arctan\left(\frac{4}{x_1 + x_2}\right) - x_2, \quad z_3 = (x_1 + x_2)^2 \left(1 + \frac{16}{(x_1 + x_2)^2}\right).$$

Die zwei Lösungen des Problems sind

$$x_1^* = (0.388\,899\,948\,8, 1.193\,984\,154\,6)^T \text{ und } x_2^* = (-0.246\,902\,265\,8, -1.219\,426\,089\,3)^T.$$

Als Startvektoren wählen wir $x^{(0)} = (0, 0.5)^T$, $(0, -0.5)^T$. $x^{(0)} = (c, -c)^T$ führt zum Abbruch. Das MGW konvergiert.

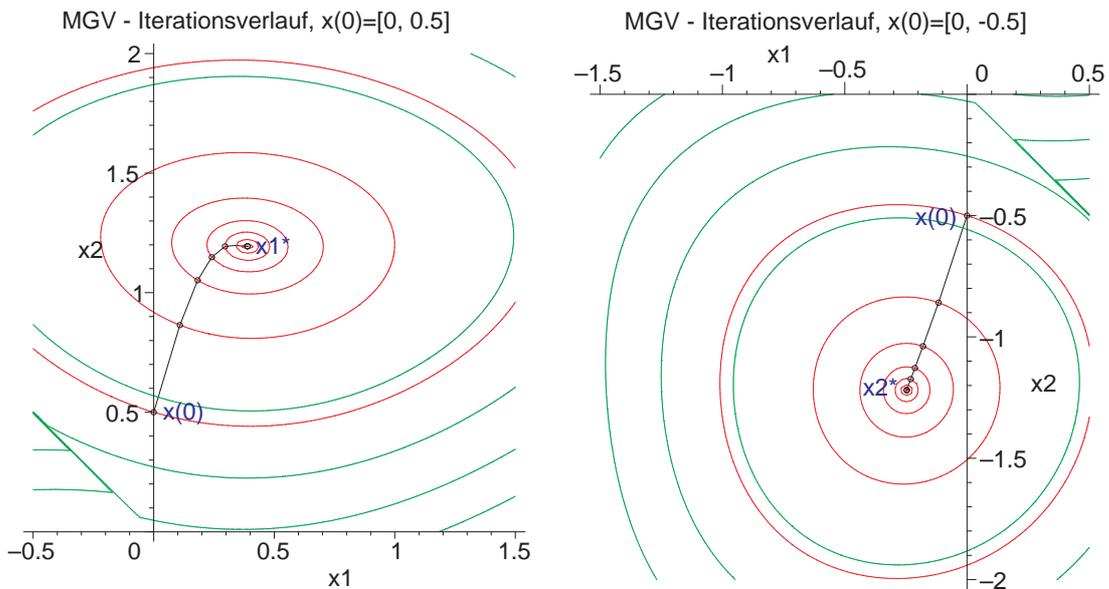


Abb. 3.37 Dateien *mgv_non_lin_06a, b.ps*
MGW mit $x^{(0)} = (0, 0.5)^T$ und $(0, -0.5)^T$

x_1^* und x_2^* sind die einzigen beiden globalen Minima.

Man kann leicht den Bereich abschätzen, indem sich die Lösungen befinden müssen. Wegen $|\sin(x_i)| \leq 1$ und $|\arctan(z)| < \pi/2$ sind die Schranken dafür

$$|x_2| < \pi/2 \quad \text{und} \quad |x_1| < \pi/6.$$

Über der Geraden $x_2 = -x_1$ hat die Funktion $h(x)$ den Verlauf der quadratischen Parabeln $x_2^2 + (\pi/2 \pm x_2)^2$.

Zum Koordinatenursprung gehört die Höhenlinie $h(x) = (\pi/2)^2 = 2.467\,401$.

Höhenlinienbild zu $h(x)$ mit 2 globalen Minima

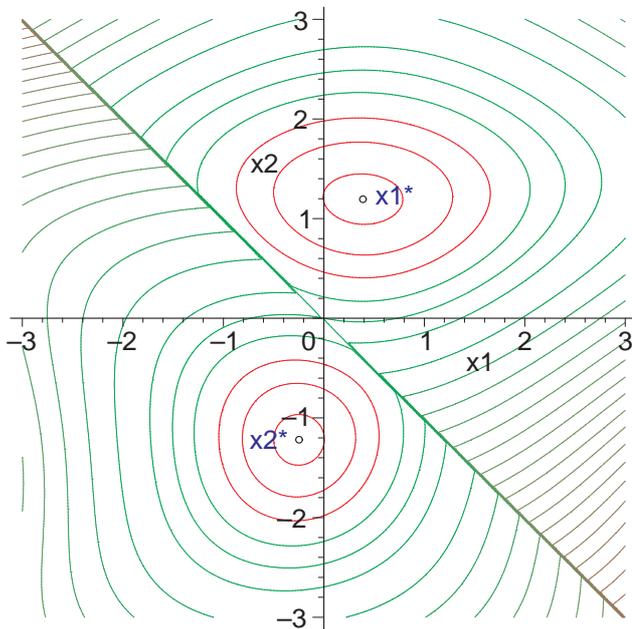


Abb. 3.38

Datei

mgv_non_lin_06c.ps

Höhenlinienbild

zu $h(x)$ mit

$x \in [-3, 3] \times [-3, 3]$

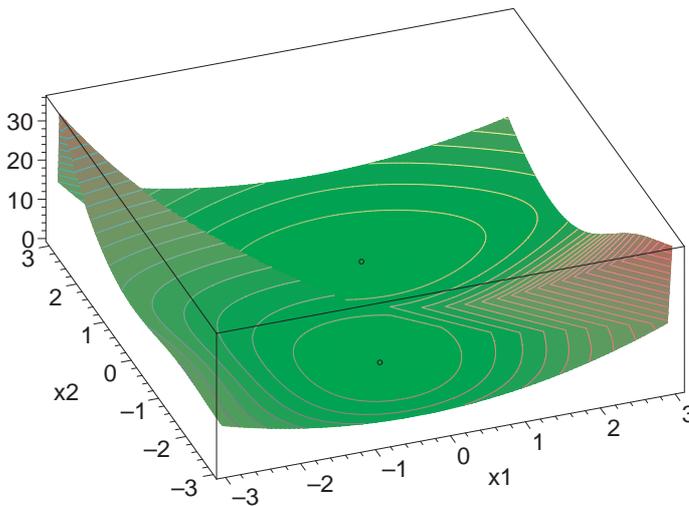


Abb. 3.39

Datei

mgv_non_lin_06d.ps

3D-Höhenlinienbild

zu $h(x)$ mit

$x \in [-3, 3] \times [-3, 3]$ ◀

Beispiel 3.17 Sei

$$\begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} x_1 - \frac{x_2}{4 - \sin(x_2)} \\ x_1 + x_2 - \frac{4}{\tan(x_2)} \end{pmatrix}$$

und damit

$$h(x_1, x_2) = \left(x_1 - \frac{x_2}{4 - \sin(x_2)} \right)^2 + \left(x_1 + x_2 - \frac{4}{\tan(x_2)} \right)^2$$

sowie

$$h'(x_1, x_2) = \begin{pmatrix} 4x_1 + 2x_2 - \frac{2x_2}{4 - \sin(x_2)} - \frac{8}{\tan(x_2)} \\ 2\left(x_1 - \frac{x_2}{4 - \sin(x_2)}\right) \left(-\frac{1}{4 - \sin(x_2)} - \frac{x_2 \cos(x_2)}{(4 - \sin(x_2))^2}\right) + \\ + 2\left(x_1 + x_2 - \frac{4}{\tan(x_2)}\right) \left(1 + 4\frac{1 + \tan^2(x_2)}{\tan^2(x_2)}\right) \end{pmatrix}.$$

Das NLGS entsteht durch eine Umformung des NLGS aus Beispiel 3.16.

Dabei vergrößert sich die Lösungsmannigfaltigkeit. Es gibt unendlich viele Lösungen, die Schnittpunkte der Kurven

$$x_1 = y_1(x_2) = \frac{x_2}{4 - \sin(x_2)} \quad \text{und} \quad x_1 = y_2(x_2) = -x_2 + \frac{4}{\tan(x_2)}$$

sind. Zu den beiden Lösungen x_1^* und x_2^* aus Beispiel 3.16 kommen “nach links unten und rechts oben“ unendlich viele hinzu. Die nächsten sind

$$x_3^* = (0.827\,776\,700\,0, 3.849\,135\,478\,5)^T \quad \text{und} \quad x_4^* = (-1.133\,180\,103\,3, -3.820\,844\,281\,7)^T.$$

Als Startvektor wählen wir $x^{(0)} = (0, 0.5)^T, (0, -0.5)^T, (0, 3.2)^T, (5, 3.2)^T$.

Das MGW konvergiert.

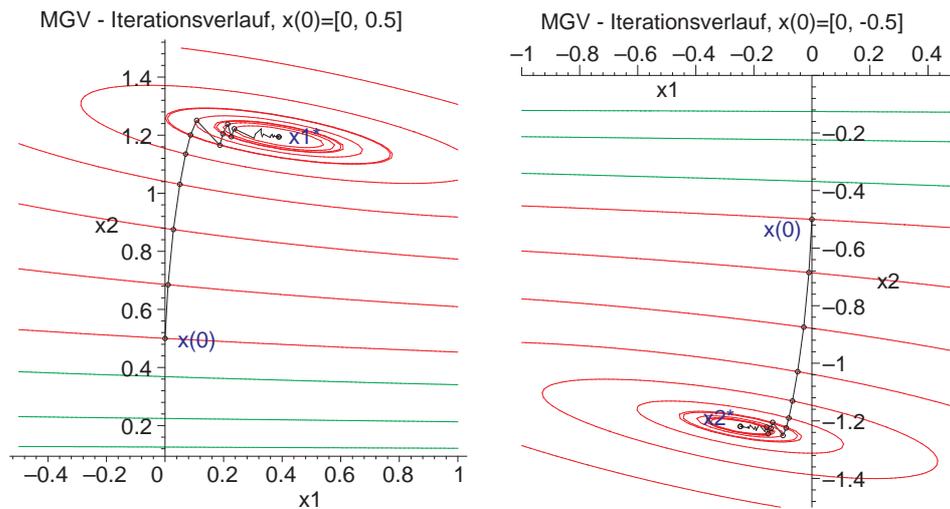


Abb. 3.40 Dateien *mgv_non_lin_07a, b.ps*, MGW mit $x^{(0)} = (0, 0.5)^T, (0, -0.5)^T$

Die globalen Minima liegen in relativ flachen “Senken“ und sind nur nach sehr vielen Schritten zu erreichen. Dabei kann es im Iterationsverlauf durch flache Tangentialebenen und damit flache Abstiegsrichtungen an den Iterierten zu einem Wechsel zwischen den Senken kommen.

Wegen der Polstellen $\tan(x_2) = 0$ für $x_2 = k\pi$ ist es beim Kontur-Plot angebracht, den Gesamtbereich geeignet zu unterteilen.

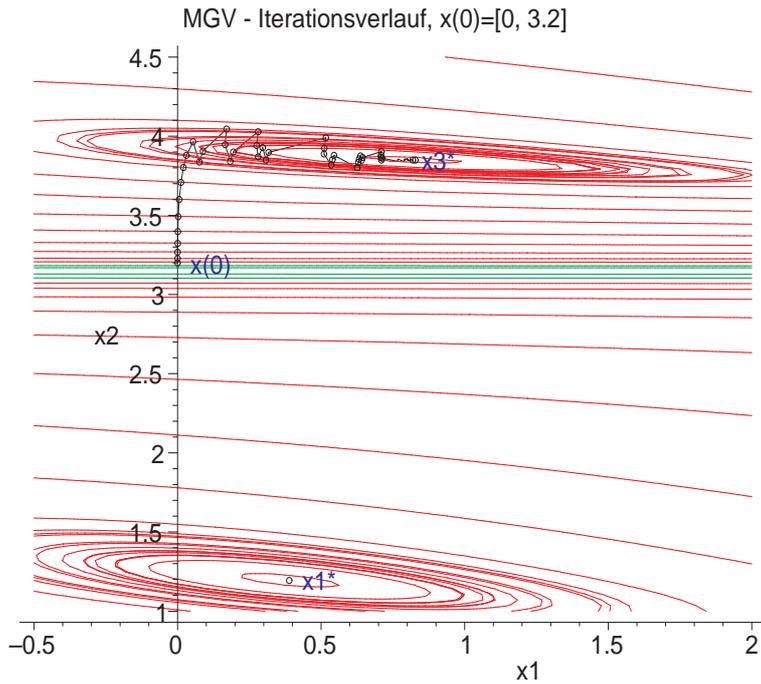


Abb. 3.41

Datei

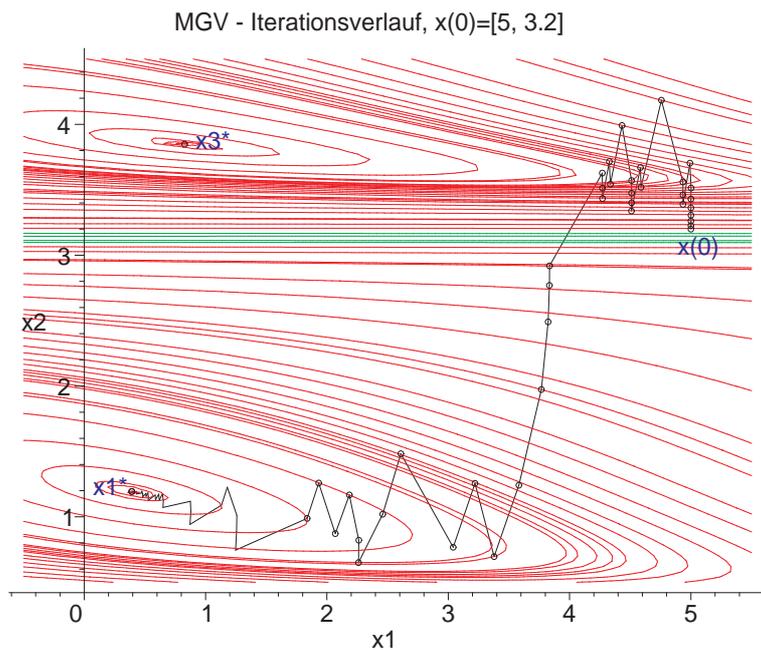
*mgv_non_lin_07d.ps*MGV mit $x^{(0)} = (0, 3.2)^T$ 

Abb. 3.42

Datei

*mgv_non_lin_07e.ps*MGV mit $x^{(0)} = (5, 3.2)^T$

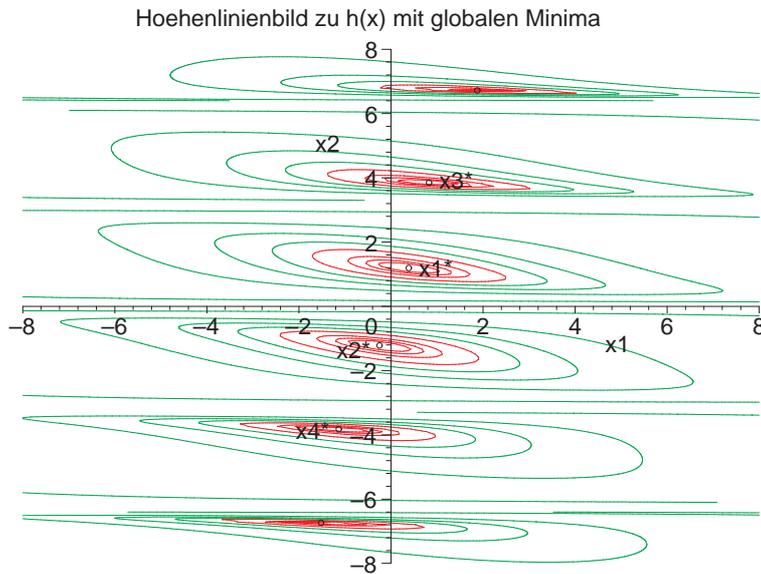


Abb. 3.43

Datei
mgv_non_lin_07c.ps
 Höhenlinienbild
 zu $h(x)$ mit
 $x \in [-8, 8] \times [-8, 8]$

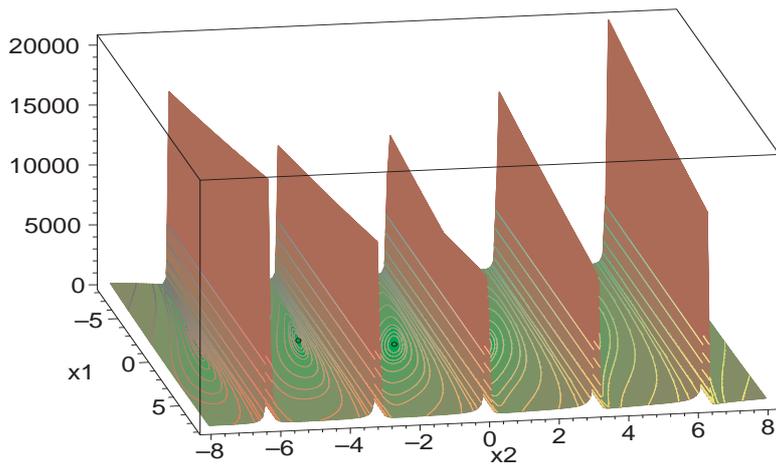


Abb. 3.44

Datei
mgv_non_lin_07f.ps
 3D-Höhenlinienbild
 zu $h(x)$ mit
 $x \in [-8, 8] \times [-8, 8]$
 (Datei ca 10 MByte)

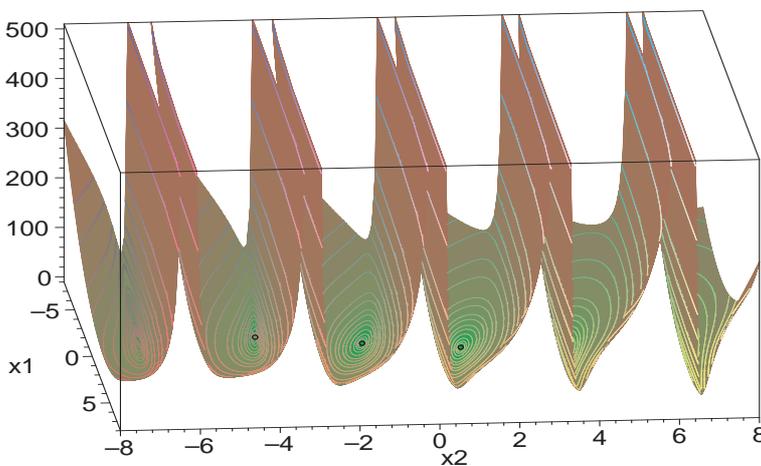


Abb. 3.45

Datei
mgv_non_lin_07g.ps
 3D-Höhenlinienbild
 (abgeschnitten)
 zu $h(x)$ mit
 $x \in [-8, 8] \times [-8, 8]$
 (Datei ca 13 MByte) ◀

Kapitel 4

Zusammenfassung

Wie sich durch die vorangegangenen Betrachtungen bestätigt hat, schaffen es beide Verfahren, das GV unter bestimmten Voraussetzungen und das MGV immer, die an sie gestellte Forderung der Lösung des LGS zu erfüllen.

Es zeigte sich aber auch, dass die Konvergenz des MGV in den meisten Fällen spürbar langsamer als die des konvergenten GV war. Dies ist verständlich, da man beim MGV der Abstiegsgeraden an das Funktional nicht bis zu dem Punkt folgt, an welchem das Funktional einen minimalen Wert annimmt, sondern (meist schon vorher, eventuell aber auch nachher) hält, nämlich wenn die z -Koordinate der Abstiegsgeraden 0 wird. Dies hat eine (zum Teil wesentlich) kürzere oder zu lange "Schrittweite" als beim GV zur Folge, dementsprechend mehr Schritte braucht man bis zur Lösung bzw. zur Minimumstelle.

Allerdings macht dies das MGV auch "stabiler" als das GV. Das GV zur symmetrisierten Form, also zum NGS, rechnet aber in jedem Schritt bis an die "eindimensionale" Minimumstelle und kann somit mit dem MGV leicht konkurrieren.

Beim MGV ist eine Symmetrisierung des LGS bei nicht symmetrischer Matrix A nicht notwendig, da das Funktional $h(x)$ schon mit der symmetrisierten Form arbeitet. Die Indefinitheit der Systemmatrix stört grundsätzlich nicht das MGV.

Auch im nichtlinearen Fall, wo man mit einer großen Lösungsmannigfaltigkeit des Systems rechnen muss, kann das MGV erfolgreich zur Berechnung einzelner oder ausgewählter Lösungen eingesetzt werden. Es kann im Falle der Konvergenz jedoch nicht mit dem Newton-Verfahren konkurrieren.

Für welches Verfahren man sich aber letztendlich entscheidet, hängt sowohl von den Voraussetzungen als auch vom Aspekt des vertretbaren Aufwands ab, aber auch von persönlichen Vorlieben ab. Schließlich gibt es noch eine lange Reihe weiterer Verfahren, die alle demselben Zweck dienen: dem Lösen LGS und NLGS.

Literaturverzeichnis

- [1] NEUNDORF, W.: *Lösungsmethoden mit Maple*. Preprint M 08/03
IfMath der TU Ilmenau, 2003.
- [2] NEUNDORF, W.: *Grundlagen der numerischen linearen Algebra*. Preprint M 04/04
IfMath der TU Ilmenau, 2004.
- [3] NEUNDORF, W.: *Abstiegsverfahren Teil I*. Preprint M 19/04
IfMath der TU Ilmenau, 2004.
- [4] NEUNDORF, W.: *Abstiegsverfahren Teil II*. Preprint M 20/04
IfMath der TU Ilmenau, 2004.
- [5] NEUNDORF, W.: *Abstiegsverfahren Teil III*. Preprint M 09/05
IfMath der TU Ilmenau, 2004.
- [6] NEUNDORF, W.: *Zur Konvergenz des Gradientenverfahrens*. Preprint M 13/05
IfMath der TU Ilmenau, 2005.
- [7] BABOVSKY, H.; NEUNDORF, W.: *Numerik großer Gleichungssysteme*. Vorlesungsskript
WS 2005/06, IfMath der TU Ilmenau, 2005.
- [8] BRONSTEIN, I.N. (u. a.): *Taschenbuch der Mathematik*. 5. Auflage
Verlag Harri Deutsch, 2000.
- [9] NEUNDORF, W.: *Numerische Mathematik*. Vorlesungen, Übungen, Algorithmen und
Programme. Shaker Verlag Aachen, 2002.