

3 Lösungsverfahren für Optimierungsprobleme ohne Nebenbedingungen

Übersicht

3.1	Numerische Grundlagen	75
3.2	Das Newton-Verfahren	90
3.3	Ein allgemeines Abstiegsverfahren mit Richtungssuche	99
3.4	Modifizierte Newton-Verfahren	143
3.5	Quasi-Newton-Verfahren	175
3.6	Verfahren der konjugierten Gradienten (CG-Verfahren)	207
3.7	Trust-Region-Verfahren (TR-Verfahren)	235
3.8	Verfahren für diskrete Approximationsprobleme	268
3.9	Übungsaufgaben zu Kapitel 3	284

3.1 Numerische Grundlagen

3.1.1 Konvergenzgeschwindigkeit

Für die Bewertung von iterativen Verfahren ist die Geschwindigkeit, mit der eine Iterationsfolge $\{\mathbf{x}^k\}_{k \in \mathbb{N}} \subset \mathbb{R}^n$ gegen eine (lokale oder globale) Lösung $\mathbf{x}^* \in \mathbb{R}^n$ des Optimierungsproblems konvergiert, ein wichtiges Kriterium.

Definition 3.1 (Q-Konvergenzgeschwindigkeit)

Es seien $\{\mathbf{x}^k\}_{k \in \mathbb{N}} \subset \mathbb{R}^n$ und $\lim_{k \rightarrow \infty} \mathbf{x}^k = \mathbf{x}^*$. Die Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ konvergiert gegen \mathbf{x}^*

- (a) *Q-sublinear*, wenn eine Folge $\{c_k\}_{k \in \mathbb{N}}$ mit $\lim_{k \rightarrow \infty} c_k = 1$ und ein $k_0 \in \mathbb{N}$ existieren, sodass

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq c_k \|\mathbf{x}^k - \mathbf{x}^*\|$$

- (b) *Q-linear mit dem Konvergenzfaktor C*, wenn ein $C \in (0, 1)$ und ein $k_0 \in \mathbb{N}$ existieren, sodass

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq C \|\mathbf{x}^k - \mathbf{x}^*\|$$

- (c) *Q-superlinear*, wenn eine positive Nullfolge $\{c_k\}_{k \in \mathbb{N}}$ und ein $k_0 \in \mathbb{N}$ existieren, sodass

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq c_k \|\mathbf{x}^k - \mathbf{x}^*\|$$

- (d) *Q-quadratisch*, wenn ein $C > 0$ und ein $k_0 \in \mathbb{N}$ existieren, sodass

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq C \|\mathbf{x}^k - \mathbf{x}^*\|^2$$

für alle $k \in \mathbb{N}$ mit $k \geq k_0$ gilt.

Offensichtlich gilt: Eine Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}} \subset \mathbb{R}^n$ konvergiert genau dann Q-sublinear, Q-linear, Q-superlinear bzw. Q-quadratisch gegen $\mathbf{x}^* \in \mathbb{R}^n$, wenn

- die Folge $\{\mathbf{x}^k - \mathbf{x}^*\}_{k \in \mathbb{N}} \subset \mathbb{R}^n$ mit der entsprechenden Q-Konvergenzgeschwindigkeit gegen $\mathbf{0} \in \mathbb{R}^n$ konvergiert.
- für alle $\alpha \in \mathbb{R} \setminus \{0\}$ die Folge $\{\alpha \mathbf{x}^k\}_{k \in \mathbb{N}} \subset \mathbb{R}^n$ mit der entsprechenden Q-Konvergenzgeschwindigkeit gegen $\alpha \mathbf{x}^*$ konvergiert.

Weiterhin möchten wir bemerken, dass im Gegensatz zur Q-linearen und Q-superlinearen Konvergenz bei der Definition der Q-sublinearen und Q-quadratischen Konvergenz einer Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ gegen \mathbf{x}^* die Voraussetzung $\lim_{k \rightarrow \infty} \mathbf{x}^k = \mathbf{x}^*$ explizit gefordert werden muss und die Eigenschaft der linearen Konvergenz abhängig von der gewählten Norm ist.

Die aufgeführten Definitionen zur Q-Konvergenzgeschwindigkeit basieren auf dem Quotientenkriterium zur absoluten Konvergenz von Reihen. Analog gibt es Definitionen zur R-Konvergenzgeschwindigkeit, die sich auf das Wurzelkriterium beziehen (siehe z. B. Ortega und Rheinboldt (1970)).

Definition 3.2 (R-Konvergenzgeschwindigkeit)

Es seien $\{\mathbf{x}^k\}_{k \in \mathbb{N}} \subset \mathbb{R}^n$ und $\lim_{k \rightarrow \infty} \mathbf{x}^k = \mathbf{x}^*$. Die Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ konvergiert *R-linear*, *R-superlinear* bzw. *R-quadratisch* gegen \mathbf{x}^* , wenn eine Q-lineare, Q-superlineare bzw. Q-quadratische positive Nullfolge $\{\varepsilon_k\}_{k \in \mathbb{N}}$ (als Majorante) und ein $k_0 \in \mathbb{N}$ existieren, sodass $\|\mathbf{x}^k - \mathbf{x}^*\| \leq \varepsilon_k$ für alle $k \geq k_0$ gilt.

Offensichtlich folgt bei Vorliegen einer Q-Konvergenzgeschwindigkeit die entsprechende R-Konvergenzgeschwindigkeit. Die hier definierte R-Konvergenzgeschwindigkeit ist genau genommen eine Mindestgeschwindigkeit (siehe Abb. 3.2). Weitergehende Betrachtungen findet man in Kosmol (1993) und detailliert in Schwetlick (1979). In den Abbildungen 3.1 und 3.2 zeigen wir anhand einfacher Folgen, wie eine entsprechende Q- bzw. R-Konvergenzgeschwindigkeit in einer halblogarithmischen Darstellung zu erkennen ist.

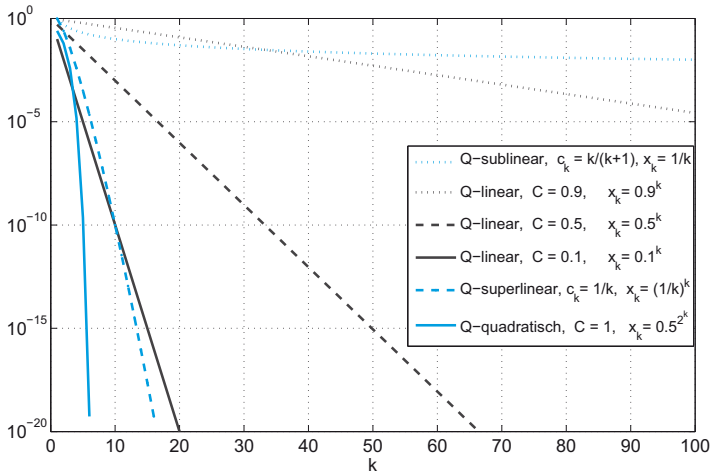


Abb. 3.1 Q-Konvergenzgeschwindigkeiten in halblogarithmischer Darstellung

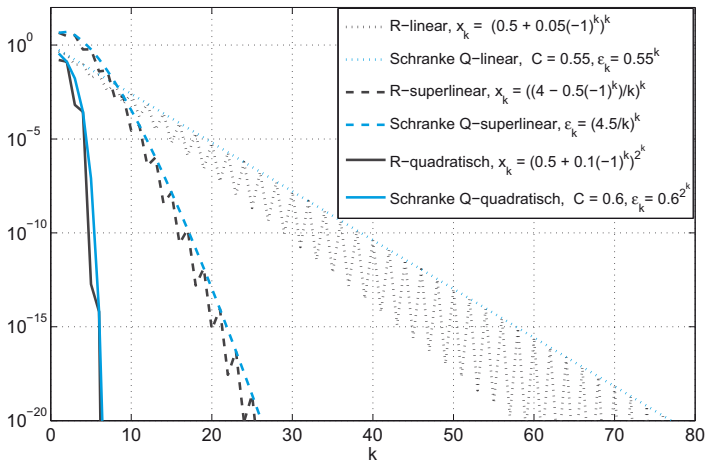


Abb. 3.2 R-Konvergenzgeschwindigkeiten in halblogarithmischer Darstellung

Für Funktionen $f \in C^1(\mathbb{R}^n, \mathbb{R})$ bzw. $f \in C^2(\mathbb{R}^n, \mathbb{R})$ kann aus der Q-Konvergenzgeschwindigkeit einer Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ gegen \mathbf{x}^* auf die entsprechende R-Konvergenzgeschwindigkeit der zugehörigen Funktionswertfolge $\{f(\mathbf{x}^k)\}_{k \in \mathbb{N}}$ gegen $f(\mathbf{x}^*)$ bzw. der zugehörigen Gradientenfolge $\{\nabla f(\mathbf{x}^k)\}_{k \in \mathbb{N}}$ gegen $\nabla f(\mathbf{x}^*)$ geschlossen werden, wie der folgende Satz zeigt.

Satz 3.3

Es seien $f \in C^1(\mathbb{R}^n, \mathbb{R})$ und $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ eine gegen \mathbf{x}^* Q-linear, Q-superlinear bzw. Q-quadratisch konvergente Folge. Dann gilt:

- (a) Die Folge der Funktionswerte $\{f(\mathbf{x}^k)\}_{k \in \mathbb{N}}$ konvergiert in der entsprechenden R-Konvergenzgeschwindigkeit gegen $f(\mathbf{x}^*)$.

Für $f \in C^2(\mathbb{R}^n, \mathbb{R})$ gilt zusätzlich:

- (b) Die Folge der Gradienten $\{\nabla f(\mathbf{x}^k)\}_{k \in \mathbb{N}}$ konvergiert in der entsprechenden R-Konvergenzgeschwindigkeit gegen $\nabla f(\mathbf{x}^*)$.

Beweis: Es sei $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ eine gegen \mathbf{x}^* Q-linear, Q-superlinear bzw. Q-quadratisch konvergente Folge. Wegen $\lim_{k \rightarrow \infty} \mathbf{x}^k = \mathbf{x}^*$ existiert ein \hat{k}_0 mit

$$\|o(\mathbf{x}^* - \mathbf{x}^k)\| \leq \|\mathbf{x}^* - \mathbf{x}^k\|$$

für alle $k \geq \hat{k}_0$.

Zu (a): Für $f \in C^1(\mathbb{R}^n, \mathbb{R})$ gilt

$$f(\mathbf{x}^*) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T (\mathbf{x}^* - \mathbf{x}^k) + o(\|\mathbf{x}^* - \mathbf{x}^k\|),$$

und es existiert ein $K \geq 0$ mit $\|\nabla f(\mathbf{x}^k)\| \leq K$ für alle $k \in \mathbb{N}$. Somit folgt

$$\begin{aligned} \|f(\mathbf{x}^k) - f(\mathbf{x}^*)\| &= \|\nabla f(\mathbf{x}^k)^T (\mathbf{x}^* - \mathbf{x}^k) + o(\|\mathbf{x}^* - \mathbf{x}^k\|)\| \\ &\leq \|\nabla f(\mathbf{x}^k)^T (\mathbf{x}^* - \mathbf{x}^k)\| + \|o(\|\mathbf{x}^* - \mathbf{x}^k\|)\| \\ &\leq \|\nabla f(\mathbf{x}^k)\| \|\mathbf{x}^* - \mathbf{x}^k\| + \|\mathbf{x}^* - \mathbf{x}^k\| \\ &= (K + 1) \|\mathbf{x}^k - \mathbf{x}^*\| \end{aligned}$$

für alle $k \geq \hat{k}_0$, womit die Konvergenz der Folge $\{f(\mathbf{x}^k)\}_{k \in \mathbb{N}}$ gegen $f(\mathbf{x}^*)$ in der entsprechenden R-Konvergenzgeschwindigkeit gezeigt ist.

Zu (b): Für $f \in C^2(\mathbb{R}^n, \mathbb{R})$ existiert ein $M \geq 0$ mit $\|\nabla^2 f(\mathbf{x}^k)\| \leq M$ für alle $k \in \mathbb{N}$, und es folgt analog in diesem Fall die Abschätzung

$$\|\nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^*)\| \leq (M + 1) \|\mathbf{x}^k - \mathbf{x}^*\|$$

für alle $k \geq \hat{k}_0$ und somit die Konvergenz der Folge $\{\nabla f(\mathbf{x}^k)\}_{k \in \mathbb{N}}$ gegen $\nabla f(\mathbf{x}^*)$ in der entsprechenden R-Konvergenzgeschwindigkeit. \square

Wenn die Funktion f in einer konvexen Umgebung eines stationären Punktes \mathbf{x}^* gleichmäßig konvex ist, dann kann wiederum unter zusätzlichen Differenzierbarkeitsvoraussetzungen aus der Konvergenzgeschwindigkeit der Funktionswerte oder Gradienten auch auf die Konvergenzgeschwindigkeit der Iterierten geschlossen werden. Aus diesem Grunde ist die (lokale) gleichmäßige Konvexität der Zielfunktion eine entscheidende Voraussetzung bei Konvergenzsätzen für Verfahren der Optimierung.

Satz 3.4

Die Funktion $f \in C^1(\mathbb{R}^n, \mathbb{R})$ sei gleichmäßig konvex über einer konvexen Umgebung U von $\mathbf{x}^* \in \mathbb{R}^n$ mit $\nabla f(\mathbf{x}^*) = \mathbf{0}$. Ferner existiere ein k_0 mit $\mathbf{x}^k \in U$ für alle $k \geq k_0$. Dann gilt:

- (a) Konvergiert die Folge $\{f(\mathbf{x}^k) - f(\mathbf{x}^*)\}_{k \in \mathbb{N}}$ Q-linear, Q-superlinear bzw. Q-quadratisch gegen 0, dann konvergiert die Folge $\{\|\mathbf{x}^k - \mathbf{x}^*\|\}_{k \in \mathbb{N}}$ in der entsprechenden R-Geschwindigkeit gegen 0.

Für $f \in C^2(\mathbb{R}^n, \mathbb{R})$ gilt zusätzlich:

- (b) Konvergiert eine der Folgen

$$\{f(\mathbf{x}^k) - f(\mathbf{x}^*)\}_{k \in \mathbb{N}}, \{\|\nabla f(\mathbf{x}^k)\|\}_{k \in \mathbb{N}} \text{ oder } \{\|\mathbf{x}^k - \mathbf{x}^*\|\}_{k \in \mathbb{N}}$$

Q-superlinear bzw. Q-quadratisch gegen 0, dann sind die jeweils beiden anderen Folgen ebenfalls Q-superlinear bzw. Q-quadratisch gegen 0 konvergent.

- (c) Konvergiert eine der Folgen

$$\{f(\mathbf{x}^k) - f(\mathbf{x}^*)\}_{k \in \mathbb{N}}, \{\|\nabla f(\mathbf{x}^k)\|\}_{k \in \mathbb{N}} \text{ oder } \{\|\mathbf{x}^k - \mathbf{x}^*\|\}_{k \in \mathbb{N}}$$

Q-linear gegen 0, dann sind die jeweils beiden anderen Folgen R-linear gegen 0 konvergent.

Wir möchten an dieser Stelle auf den Beweis von Satz 3.4 verzichten und verweisen diesbezüglich auf Aufgabe 3.2.

Bemerkung 3.5

Bei den später betrachteten numerischen Verfahren werden wir anhand von Experimenten auch solche charakteristischen Eigenschaften der Verfahren aufzeigen, die sich nicht in mathematischen Sätzen formulieren lassen. In den entstehenden Tabellen und Abbildungen findet man endlich viele Glieder der Folgen der Funktionswerte („fiter“) und der Gradientennormen („err“), wobei sich der maximale Index k durch ein entsprechendes

Abbruchkriterium ergibt. Aus den ersten $k + 1$ Gliedern einer Folge mit Startpunktindex $k = 0$ kann man natürlich nicht auf die Konvergenzgeschwindigkeit der gesamten Folge schließen, sondern nur eine Tendenz feststellen. Diese Tendenzen erkennt man häufig bereits bis zum Abbruch des Verfahrens. Bei superlinearer bzw. quadratischer Konvergenz ist diese Tendenz oftmals jedoch erst kurz vor dem Abbruch des Verfahrens festzustellen. **Wir formulieren dann in den Experimenten, dass man in den Tabellen bzw. Abbildungen die lineare, superlineare bzw. quadratische Konvergenzgeschwindigkeit erkennen kann, meinen aber immer, dass eine Tendenz dazu feststellbar ist.** Auch spezifizieren wir hier (im Gegensatz zu den theoretischen Konvergenzaussagen) nicht, ob Q- oder R-lineare Konvergenz vorliegt. Wir sprechen in den Experimenten einfach von linearer, superlinearer bzw. quadratischer Konvergenz. Man erkennt bei einer abgebrochenen positiven Nullfolge, beispielsweise der Folge der Gradientennormen, eine Tendenz zur linearen (superlinearen) Konvergenz an der linearen (schneller als linearen) Zunahme der führenden (Nachkomma-)Nullen. Bei quadratischer Konvergenz verdoppeln sich faktisch von Iteration zu Iteration die korrekten führenden Nullen. Bei Exponentialschreibweise der Zahlen erkennt man dies gut an dem Exponenten in der Zehnerpotenz. ■

3.1.2 Symbolische, automatische und numerische Differenziation

Symbolische Differenziation (SD)

Wenn wir von symbolischer Differenziation sprechen, dann verstehen wir darunter, dass die analytischen Formeln für die Gradienten und Hesse-Matrizen aus der analytischen Darstellung der zugehörigen Funktion berechnet werden (also z. B. unter MATLAB als m-File). Diese analytischen Ausdrücke können aufwendig per Hand oder maschinell durch Formelmanipulationssoftware wie z. B. **Mathematika**^{®1} oder **Maple**^{™2} bestimmt werden. Unter EDOPTLAB wird dies mit der **Symbolic Math Toolbox** von MATLAB realisiert, indem aus einem m-File zur Beschreibung des Optimierungsproblems (Problem m-File) die m-Files für die Problemfunktion (**func0.m**) und ihre ersten (**grad0.m**) und zweiten partiellen Ableitungen (**hess0.m**) erzeugt werden.

Automatische Differenziation (AD)

Die automatische Differenziation wurde in den vergangenen 25 Jahren entwickelt. Es existieren eine Reihe von diesbezüglichen Software-Paketen, wie z. B. ADOL C (siehe Griewank et. al. (1996)) bzw. INTLAB (siehe Rump (1999)) zur Einbindung unter C++

¹Mathematika[®] ist ein eingetragenes Warenzeichen der Firma Wolfram Research, U.S.A., <http://www.wolfram.com>

²Maple[™] ist ein eingetragenes Warenzeichen der Firma Waterloo Maple Inc., Kanada, www.maplesoft.com

bzw. MATLAB. Im Folgenden geben wir eine kurze Einführung zur AD. Für ein tieferes Studium empfehlen wir Griewank und Walther (2008).

Um die AD benutzen zu können, benötigt man wie bei der SD einen Code für die analytische Darstellung der Funktion. Unter MATLAB ist dies z. B. ein m-File. Die Software zur AD berechnet eine analytische Darstellung der ersten bzw. zweiten Ableitungen.

Die AD zerlegt zu diesem Zweck eine Funktion $f \in C^1(\mathbb{R}^n, \mathbb{R})$, deren Gradient zu ermitteln ist, unter Berücksichtigung bekannter Differenzierungsregeln in eine endliche Anzahl „elementarer Funktionen“ f_i (unäre und binäre Funktionen), sodass sich die Funktion f und ihr Gradient sukzessive aus diesen f_i und ihren Ableitungen an einer gewünschten Stelle \mathbf{x} berechnen lassen.

Es seien $f \in C^1(\mathbb{R}^n, \mathbb{R})$ und $f_i \in C^1(\mathbb{R}^{n_i}, \mathbb{R})$, $i = n+1, \dots, m$ ($m > n$) gemäß vorliegenden der AD-Software auswertbare elementare Funktionen. Jede dieser Funktionen f_i sei von $x_k, k \in J_i, |J_i| = n_i \leq 2$ Variablen abhängig, und f sei wie folgt aus diesen elementaren Funktionen berechenbar:

$$\begin{aligned} &\text{for } i = n+1 : 1 : m \\ &\quad x_i := f_i(x_k, k \in J_i); \\ &\text{end;} \\ &f(\mathbf{x}) = x_m; \end{aligned} \tag{3.1}$$

Im Folgenden erläutern wir am Beispiel der sogenannten *zweidimensionalen Rosenbrock-Funktion* (wegen der Gestalt ihrer Höhenlinien auch *Bananen-Funktion* genannt)

$$f : \mathbb{R}^2 \rightarrow \mathbb{R} \text{ mit } f(\mathbf{x}) := 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

das Prinzip der AD, wenn eine Zerlegung der zu differenzierenden Funktion in elementare Funktionen bekannt ist. Wie die AD-Software erkennt, aus welchen elementaren Bestandteilen sich die Funktion zusammensetzt, beschreiben wir an dem Beispiel nicht. Zunächst definieren wir für das Beispiel zusätzliche Variablen x_3 bis x_9 bzw. Funktionen f_3 bis f_9 wie folgt:

$$\begin{aligned} x_3 &= f_3(x_1) &:= x_1^2, \\ x_4 &= f_4(x_2, x_3) &:= x_3 - x_2, \\ x_5 &= f_5(x_4) &:= x_4^2, \\ x_6 &= f_6(x_5) &:= 100x_5, \\ x_7 &= f_7(x_1) &:= 1 - x_1, \\ x_8 &= f_8(x_7) &:= x_7^2, \\ x_9 &= f_9(x_6, x_8) &:= x_6 + x_8, \\ f(x_1, x_2) &= x_9. \end{aligned}$$

Diese Zerlegung von f wird durch den gerichteten Graphen $G = (V, E)$ gemäß Abb. 3.3 repräsentiert.

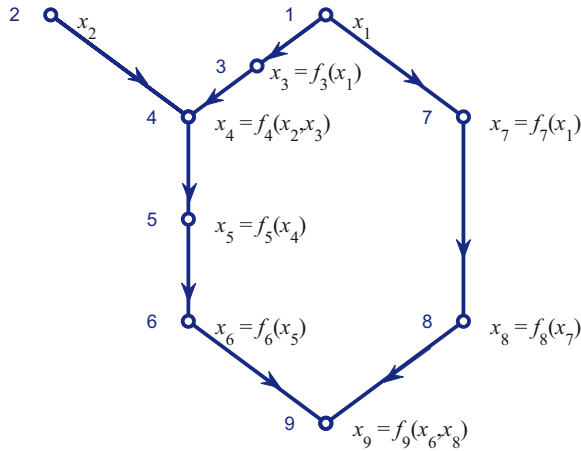


Abb. 3.3 Graph der Zerlegung für die zweidimensionale Rosenbrock-Funktion in elementare Funktionen

Die Indextmengen J_i aus (3.1) für $i = 3, \dots, 9$ entsprechen dabei den Knotenmengen der Bögen $(j, i) \in E$, die den Endknoten i besitzen (Indexmenge der unmittelbaren Vorgänger des Knotens i).

Es gilt für das Beispiel:

$$J_3 = \{1\}, J_4 = \{2, 3\}, J_5 = \{4\}, J_6 = \{5\}, J_7 = \{1\}, J_8 = \{7\}, J_9 = \{6, 8\}.$$

Wir können nun zusammen mit der sukzessiven Berechnung (der Werte) der Funktionen f_i entsprechend (3.1) bzw. der $x_i, i = 3, \dots, 9$ auch die (totalen) Gradienten $\nabla x_i := \nabla f_i(x_k, k \in J_i)$ der elementaren Funktionen bzgl. der Variablen x_1, x_2 berechnen.

Wir erhalten zunächst

$$\nabla x_1 = \left(\frac{\partial x_1}{\partial x_1}, \frac{\partial x_1}{\partial x_2} \right)^T = (1, 0)^T, \quad \nabla x_2 = (0, 1)^T$$

und berechnen weiter mit den üblichen Differenziationsregeln

$$\begin{aligned} \nabla x_3 &= 2x_1 \nabla x_1, \\ \nabla x_4 &= \nabla x_3 - \nabla x_2, \\ \nabla x_5 &= 2x_4 \cdot \nabla x_4, \\ \nabla x_6 &= 100 \nabla x_5, \\ \nabla x_7 &= -\nabla x_1, \\ \nabla x_8 &= 2x_7 \nabla x_7, \\ \nabla x_9 &= \nabla x_6 + \nabla x_8, \\ \nabla f &= \nabla x_9. \end{aligned}$$

Sukzessives Einsetzen bestätigt das erwartete Resultat:

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

Setzt man voraus, dass die Struktur des Graphen G und die Bewertungen f_i der Knoten V gespeichert sind, so können Funktionswerte und partielle Ableitungen in einfacher Weise in dem sogenannten *Vorwärts-Mode* (engl. forward mode) wie folgt „synchron“ berechnet werden:

$x_1 = 2, x_2 = 2$	$\nabla x_1 = (1, 0)^T, \nabla x_2 = (0, 1)^T$
.....
$x_3 = 4$	$\nabla x_3 = (4, 0)^T$
$x_4 = 2$	$\nabla x_4 = (4, -1)^T$
$x_5 = 4$	$\nabla x_5 = (16, -4)^T$
$x_6 = 400$	$\nabla x_6 = (1600, -400)^T$
$x_7 = -1$	$\nabla x_7 = (-1, 0)^T$
$x_8 = 1$	$\nabla x_8 = (2, 0)^T$
$x_9 = 401$	$\nabla x_9 = (1602, -400)^T$
.....
$f = 401$	$\nabla f = (1602, -400)^T$

Man erkennt, dass simultan zur Berechnung des Gradienten auch der Funktionswert berechnet wird und bei der Berechnung der Hesse-Matrix gleichzeitig sowohl der Gradient als auch der Funktionswert berechnet werden. Bei der SD werden im Gegensatz dazu der Funktionswert, der Gradient und die Hesse-Matrix unabhängig voneinander berechnet. Der Vorwärtsmode der AD wird von dem unter EDOPTLAB verwendeten AD-Tool INT-LAB V 5.5 (siehe Rump (1999)) benutzt. Der numerische Aufwand zur Berechnung des Gradienten liegt lt. Theorie dabei in der gleichen Größenordnung wie der für eine numerische Approximation des Gradienten – ist also proportional zur Anzahl n der Variablen der Funktion f . Das Experiment 3.1.1 zeigt aber bezüglich der benötigten CPU-Zeiten große Unterschiede.

Neben dem Vorwärts-Mode ist der sogenannte *Rückwärts-Mode* (engl. reverse mode) von Bedeutung. Dabei beginnt man mit der letzten elementaren Funktion, im Beispiel $x_9 = f_9(x_8, x_6)$ und leitet diese Funktion unter Benutzung der Kettenregel und der elementaren Zerlegung von f nach allen Veränderlichen x_8, \dots, x_1 ab. Zum Schluss entstehen dann im Beispiel die gesuchten Ableitungen nach x_1 bzw. x_2 . Der Nachteil bei dieser Methode ist, dass alle Zwischenergebnisse geeignet gespeichert werden müssen. Bemerk-

kenswert ist dagegen, dass für den Rückwärts-Mode in Griewank und Walther (2000) gezeigt wurde, dass der Aufwand für die Berechnung des Gradienten das Fünffache des Rechenaufwandes für den Funktionswert von f nicht übersteigt, wenn die f_i nicht von n abhängen.

Eine effektive Implementierung von Algorithmen zur AD unter unterschiedlichen Gesichtspunkten (Rechenzeit, Speicherbedarf) und für unterschiedliche Anwendungen (Optimierung, Differenzialgleichungen, Optimale Steuerung) beschäftigt Informatiker und Mathematiker seit etwa zwei Jahrzehnten. Es gibt im Internet unter dem Stichwort „automatic differentiation“ eine Fülle von Hinweisen zu freier und kommerzieller Software.

Numerische Differenziation (ND)

Der wesentliche Vorteil bei der SD und AD besteht darin, dass die Ableitungen im Rahmen der möglichen Floating-Point Arithmetik (8, 16 oder mehr Stellen bei C++ oder ca. 16 Stellen bei MATLAB) genau berechnet werden. Im Gegensatz dazu entstehen bei der numerischen Differenziation Fehler durch die Approximation der Ableitungen mittels finiter Differenzen. In der Optimierung verwendet man fast immer nur *Vorwärtsdifferenzen* und approximiert

- Gradienten durch *erste Vorwärtsdifferenzen der Funktionswerte*:

$$\frac{\partial f(\mathbf{x})}{\partial x_k} = \frac{f(\mathbf{x} + h\mathbf{e}_k) - f(\mathbf{x})}{h} + O(\|h\|) ,$$

- Hesse-Matrizen durch *erste Vorwärtsdifferenzen der Gradienten*:

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_k \partial x_j} = \frac{\frac{f(\mathbf{x} + h\mathbf{e}_k)}{\partial x_j} - \frac{f(\mathbf{x})}{\partial x_j}}{h} + O(\|h\|) ,$$

- Hesse-Matrizen durch *zweite Vorwärtsdifferenzen der Funktionswerte*:

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_k \partial x_j} = \frac{f(\mathbf{x} + h(\mathbf{e}_k + \mathbf{e}_j)) - f(\mathbf{x} + h\mathbf{e}_k) - f(\mathbf{x} + h\mathbf{e}_j) + f(\mathbf{x})}{h^2} + O(\|h\|) .$$

Mit ersten zentralen Differenzen (siehe Hoffmann et al. (2005, 2006), S. 581) erreicht man einen kleineren Fehler von $O(\|h\|^2)$, wobei sich der Aufwand in etwa verdoppelt. Die ND findet dort Anwendung, wo es unmöglich oder nur mit einem unververtretbar hohen Aufwand möglich ist, die Ableitungen analytisch zu bestimmen. Außerdem kann man die ND zu Kontrollzwecken benutzen, insbesondere dann, wenn die Ableitungen vom Nutzer „per Hand“ berechnet worden sind. An zwei Experimenten demonstrieren wir, welchen Einfluss die Wahl des Inkrementes h auf die Genauigkeit der Approximation der Ableitung durch erste Vorwärtsdifferenzen hat.

Experiment 3.1.1 (Numerische Approximation der 1. Ableitung)

gradapprox.m: Die Ableitung der Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ mit $f(x) := \sin(x)$ wird an der Stelle $x = 2$ durch (erste) Vorwärts-Differenzenquotienten

$$\frac{f(x+h) - f(x)}{h}$$

approximiert. In Tab. 3.1 wird der Fehler zwischen exakter Ableitung und dem Differenzenquotienten in Abhängigkeit vom Inkrement h protokolliert.

Das Experiment bestätigt in etwa die Empfehlung (siehe Dennis und Schnabel (1983)) für die Wahl des Inkrementes h gemäß

$$h := \max \left\{ \sqrt{\text{macheps}} |x_i|, \sqrt{\text{macheps}} \right\} \text{sign}(x_i) \quad (3.2)$$

bei ersten Differenzenquotienten, wobei *macheps* die Maschinengenauigkeit bezogen auf die Zahl 1 darstellt. Dabei versucht man mit dem Faktor $\text{sign}(x_i)$ der Auslöschung von Stellen zu begegnen, wenn x nahe Null ist. In Tab. 3.1 ist das gegensätzliche Wirken des Approximationsfehlers $\left| \frac{f(x+h)-f(x)}{h} - f'(x) \right|$ für die Ableitung und des Auslöschungsfehlers $2 \left| \frac{f(x)\delta}{h} \right|$ beim Differenzenquotienten gut zu erkennen. Wo beide Fehler sich die Waage halten, erhalten wir die beste Approximation der ersten Ableitung. ■

h	df/dx	Vorw. Diff	error
1e-001	-0.416146837	-0.460880602	4.47e-002
1e-002	-0.416146837	-0.420686350	4.54e-003
1e-003	-0.416146837	-0.416601416	4.55e-004
1e-004	-0.416146837	-0.416192301	4.55e-005
1e-005	-0.416146837	-0.416151383	4.55e-006
1e-006	-0.416146837	-0.416147291	4.55e-007
1e-007	-0.416146837	-0.416146881	4.49e-008
1e-008	-0.416146837	-0.416146839	2.66e-009
1e-009	-0.416146837	-0.416146895	5.82e-008
1e-010	-0.416146837	-0.416147117	2.80e-007
1e-011	-0.416146837	-0.416144896	-1.94e-006
1e-012	-0.416146837	-0.416222612	7.58e-005
1e-013	-0.416146837	-0.416333634	1.87e-004
1e-014	-0.416146837	-0.421884749	5.74e-003
1e-015	-0.416146837	-0.333066907	-8.31e-002
1e-016	-0.416146837	0.000000000	-4.16e-001
Die Wurzel aus macheps=2.22e-016 ergibt: 1.49e-008			

Tab. 3.1 Erste Vorwärtsdifferenzen zur Approximation der 1. Ableitung von $\sin(x)$ an der Stelle $x = 2$ im Exp. 3.1.1

Experiment 3.1.2 (Numerische Approximation der Hesse-Matrix)

hessapprox.m: Die Hesse-Matrix der Funktion

$f : \mathbb{R}^2 \rightarrow \mathbb{R}$ mit $f(x) := (x_1 - 2)^4 + x_2^2(x_1 - 2)^2 + (x_2 + 1)^2$

wird an der Stelle $x = (1, 1)^T$ durch erste Vorwärtsdifferenzen der Gradienten und zweite Vorwärtsdifferenzen der Funktionswerte approximiert, wobei bei den zweiten Vorwärtsdifferenzen für das Inkrement $h := \max\{EPS|x_i|, EPS\}$ die folgenden Werte von EPS gewählt werden:

$EPS = 10^{-8}$, $EPS = \sqrt{macheps} \approx 1.5 \times 10^{-8}$ und $EPS = \sqrt[3]{macheps} \approx 6.1 \times 10^{-6}$.
Offensichtlich ist die Wahl von h gemäß (3.2) für zweite Vorwärtsdifferenzen zur Approximation der Hesse-Matrix nicht geeignet. Der größere Wert (siehe Dennis und Schnabel (1983))

$$h := \max \left\{ \sqrt[3]{macheps} |x_i|, \sqrt[3]{macheps} \right\} \text{sign}(x_i)$$

liefert hier eine „halbwegs brauchbare“ Approximation (siehe Tab. 3.2).

Als Schlussfolgerung ergeben sich bei Anwendung der zweiten Differenzenquotienten zur Approximation der Hesse-Matrix für Lösungsverfahren starke Einschränkungen in Bezug auf die Genauigkeit. ■

Berechnung der exakten Hesse-Matrix:	

14.0000000000000000	-4.0000000000000000
-4.0000000000000000	4.0000000000000000
Approximation: Inkrement h=max(EPS*abs(x),EPS)	

1. Vorwärtsdiff. der Gradienten und EPS= 1.5E-008:	

13.9999998211860660	-4.0000000000000000
-4.0000000000000000	4.0000000000000000
2. Vorwärtsdiff. der Funktionswerte und EPS= 1.0E-008:	

8.8817841970012505	-6.6613381477509375
-6.6613381477509375	0.0000000000000000
2. Vorwärtsdiff. der Funktionswerte und EPS= 1.5E-008:	

12.0000000000000000	-3.0000000000000000
-3.0000000000000000	8.0000000000000000
2. Vorwärtsdiff. der Funktionswerte und EPS= 6.1E-006:	

13.9998715884840480	-2.9999932448047044
-2.9999932448047044	3.9999183276195103

Tab. 3.2 Erste und zweite Vorwärtsdifferenzen zur Approximation der Hesse-Matrix für die Funktion aus Exp. 3.1.2 an der Stelle $x = (1, 1)^T$

Experiment 3.1.3 (CPU-Zeit-Vergleich der Differenziationstechniken)

vergldiff01.m, vergldiff02.m: Zur Demonstration der Geschwindigkeiten für die einzelnen Differenziationstechniken unter MATLAB 7.4. betrachten wir die *n*-dimensionale *Rosenbrock-Funktion* (Problem Nr. 50, Dimension $n = 20, 30, \dots, 100$)

$$f : \mathbb{R}^2 \rightarrow \mathbb{R} \text{ mit } f(\mathbf{x}) := \sum_{k=1}^{n-1} (100(x_k^2 - x_{k+1})^2 + (1 - x_k)^2) \quad (3.3)$$

und berechnen die Gradienten (siehe Abb. 3.4) und Hesse-Matrizen (siehe Abb. 3.5) mit Vorwärtsdifferenzen, AD (INTLAB V 5.5 AD Tool ohne Sparse - Technik) sowie SD (Erzeugung eines m-Files für die jeweiligen Ableitungen).

Sehr deutlich zeigt sich die Überlegenheit sowohl der AD als auch der SD gegenüber der ND. Bei der Berechnung der Hesse-Matrizen verstärkt sich dieser Effekt bei Benutzung der zweiten Differenzen. Berücksichtigt man zusätzlich die Zeiten für die einmalige Erzeugung der m-Files des Gradienten und der Hesse-Matrizen bei der SD (im Beispiel und für den verwendeten Rechner bei Dimension 50 ca. 15 Sekunden und bei Dimension 100 ca. 30 Sekunden für Gradient und Hesse-Matrix), dann ist die AD gegenüber der SD zu bevorzugen, erst recht, wenn Sparse-Techniken bei der AD genutzt werden können. Die Benutzung von Vorwärtsdifferenzen sollte nur in dem Fall erfolgen, wenn keine AD oder SD möglich ist. In EDOPTLAB ist die AD aus den oben genannten Gründen als Standard eingestellt. Durch Änderung des Parameters „diffmode“ kann sowohl die SD als auch die ND verwendet werden. ■

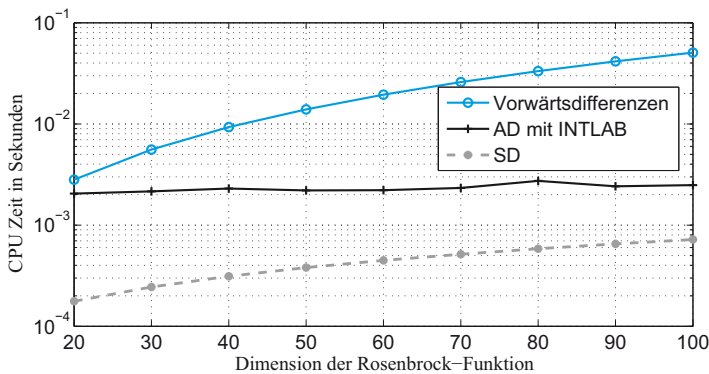


Abb. 3.4 CPU-Zeit zur Berechnung der Gradienten der Rosenbrock-Funktion mit ersten Vorwärtsdifferenzen, AD und SD im Exp. 3.1.3

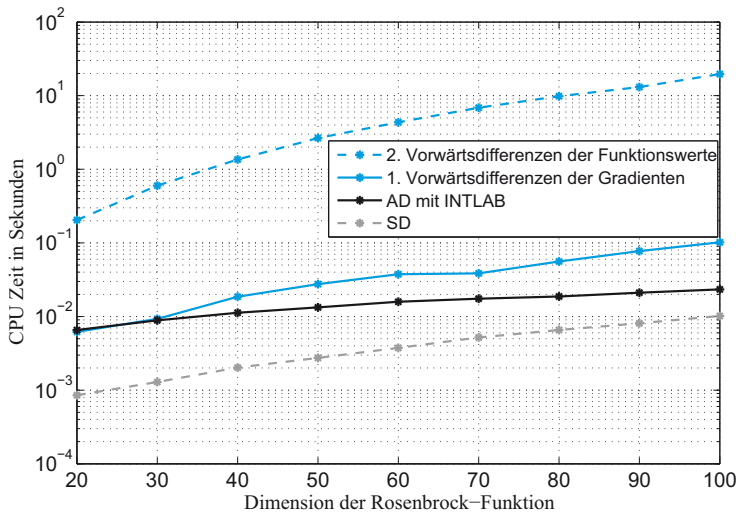


Abb. 3.5 CPU-Zeit zur Berechnung der Hesse-Matrizen der Rosenbrock-Funktion mit ersten und zweiten Vorwärtsdifferenzen, SD und AD im Exp. 3.1.3

3.1.3 Abbruchkriterien für Verfahren zur Lösung von Optimierungsproblemen ohne Nebenbedingungen

Bei einer Implementierung von Algorithmen benötigen wir Kriterien für den Abbruch der zu berechnenden Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$, wobei der letzte Iterationspunkt eine geeignete Näherungslösung der betrachteten Minimierungsaufgabe sein soll (Bertsekas (1999), S. 36/37, Fletcher (1987), S. 23, Schwetlick (1979), S. 80, 126 und Dennis und Schnabel (1983), S. 159-161). Zunächst erscheint es sinnvoll, für einen vorgegebenen absoluten Abstand $\varepsilon_x > 0$ zur Lösung \mathbf{x}^* bzw. $\varepsilon_f > 0$ zum Minimum $f(\mathbf{x}^*)$ die Kriterien

$$\|\mathbf{x}^k - \mathbf{x}^*\| < \varepsilon_x \text{ und } |f(\mathbf{x}^k) - f(\mathbf{x}^*)| < \varepsilon_f$$

zu verwenden. Wegen der Unkenntnis der Lösung bzw. des Funktionswertes am Lösungspunkt sind diese Kriterien i. Allg. nicht handhabbar. Aus den inkrementellen Kriterien

$$\|\mathbf{x}^{k+1} - \mathbf{x}^k\| < \varepsilon \text{ und } |f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)| < \varepsilon$$

kann man ohne spezielles Wissen über die theoretische Konvergenzgeschwindigkeit der Iterationsfolge nichts über die Nähe der Iterationspunkte zur Lösung aussagen. Setzt man z. B. $\mathbf{x}^k = \sum_{j=1}^k \frac{1}{j}$, so folgt $\lim_{k \rightarrow \infty} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| = \lim_{k \rightarrow \infty} \frac{1}{k+1} = 0$, aber $\lim_{k \rightarrow \infty} \mathbf{x}^k = \infty$. Wird die Folge der Iterationspunkte z. B. über einen kontrahierenden Operator T gemäß $\mathbf{x}^{k+1} = T\mathbf{x}^k$ mit der globalen Kontraktionsbedingung $\|T\mathbf{x} - T\mathbf{y}\| \leq c\|\mathbf{x} - \mathbf{y}\|$ und mit der

festen Konstante $0 \leq c < 1$ erzeugt, dann gelten nach dem Banachschen Fixpunktsatz die Abschätzungen

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq \frac{c}{1-c} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| \text{ sowie } \|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq \frac{c^{n+1}}{1-c} \|\mathbf{x}^1 - \mathbf{x}^0\| ,$$

und die inkrementelle Abbruchbedingung ist ein probates Mittel, eine Näherungslösung im Rahmen der vorgegebenen Genauigkeit zu bestimmen.

Gilt $\mathbf{x}^k \neq \mathbf{x}^*$ für alle k , so ist die Q-superlineare Konvergenz äquivalent zu

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} = 0 ,$$

und es folgt

$$0 = \lim_{k \rightarrow \infty} \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} = \lim_{k \rightarrow \infty} \left| \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\| - \|\mathbf{x}^k - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} \right| = \lim_{k \rightarrow \infty} \left| \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} - 1 \right|$$

bzw.

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} = 1.$$

Im Fall Q-superlinearer Konvergenz kann somit als Abbruchkriterium für numerische Lösungsverfahren die Bedingung $\|\mathbf{x}^{k+1} - \mathbf{x}^k\| \leq \varepsilon$ mit einem fest vorgegebenen (kleinen) ε prinzipiell genutzt werden. Eine quantitative Abstandsabschätzung zum Lösungspunkt wie beim Banachschen Fixpunktsatz ist damit aber nicht gegeben.

Deshalb nutzt man bei der restriktionsfreien Minimierung als Abbruchbedingung die näherungsweise Erfüllung der notwendigen Optimalitätsbedingung, d. h. man bricht ab, wenn

$$\|\nabla f(\mathbf{x}^k)\| < \varepsilon \tag{3.4}$$

gilt. Bei unseren folgenden Experimenten benutzen wir in den Tabellen die Bezeichnung $\mathbf{err} = \mathbf{norm}(\mathbf{g}) = \|\nabla f(\mathbf{x}^k)\|$. Jedoch erhalten wir durch (3.4) i. Allg. keine Information über den Abstand von \mathbf{x}^k zur Lösung \mathbf{x}^* . Im Falle der gleichmäßigen Konvexität von f gilt zwar die Abschätzung

$$\|\mathbf{x}^k - \mathbf{x}^*\| \leq \frac{1}{m} \|\nabla f(\mathbf{x}^k)\| ,$$

jedoch ist in der Regel die Konstante m nicht bekannt. Die oft benutzte Abbruchbedingung (3.4) besitzt noch eine Reihe weiterer Nachteile. Oft ist es besser, das relative Abbruchkriterium

$$\|\nabla f(\mathbf{x}^k)\| < \varepsilon \|\nabla f(\mathbf{x}^{typ})\| ,$$

zu verwenden, wobei \mathbf{x}^{typ} einen für die Funktion f „typischen“ x-Wert darstellt. Mit diesem relativen Abbruchkriterium würde aber bei der Minimierung von f oder αf , $\alpha > 0$ ein Verfahren jeweils nach der gleichen Anzahl von Iterationen abbrechen, was bei einem absoluten Abbruchkriterium natürlich nicht der Fall ist.

Wir benutzen in unseren Experimenten stets die Abbruchbedingung (3.4), auch wenn sie die aufgeführten Nachteile besitzt. Ist die Bedingung (3.4) erfüllt, dann wurde erfolgreich eine Näherungslösung im Rahmen der vorgegebenen Genauigkeit $\varepsilon > 0$ gefunden. Es kann jedoch passieren, dass ein Verfahren keinen stationären Punkt findet oder die Anzahl der Iterationen sehr groß wird. Um auch diese Fälle zu berücksichtigen, werden Absicherungen zur Beendigung des Verfahrens nach endlicher Zeit – sogenannte *Safeguards* – getroffen. Das sind z. B.

- 1 Überschreiten einer vorgegebenen Iterationsanzahl „maxit“
- 2 Überschreiten einer vorgegebenen Anzahl von Funktionswertberechnungen
- 3 Überschreiten einer vorgegebenen CPU-Zeit
- 4 Unterschreiten eines vorgegebenen Abstandes zwischen aufeinanderfolgenden Iterationspunkten oder Funktionswerten
- 5 Unterschreiten einer vorgegebenen positiven Schrittlänge (siehe Abschnitt 3.3)
- 6 Unterschreiten einer vorgegebenen Länge der berechneten Suchrichtung (siehe Abschnitt 3.3)
- 7 Unterschreiten einer vorgegebenen Schranke für die Funktionswerte
- 8 Auftreten von nicht verarbeitbaren Zahlenformate (nan's, inf etc. in MATLAB)

Die Safeguards **1**, **4**, **5**, **6**, **7** und **8** werden z. T. auch bei unseren Experimenten benutzt. Die unter ED_{OPT}LAB programmierten Verfahren zeigen das jeweilige Abbruchkriterium an. In den „theoretischen“ Algorithmen benutzen wir als Abbruchbedingung $\nabla f(\mathbf{x}^k) = 0$, die nur in Ausnahmefällen für endliches k erreicht wird.

3.2 Das Newton-Verfahren

Es sei $f \in C^2(\mathbb{R}^n, \mathbb{R})$. Dann gilt für alle $\mathbf{x} \in \mathbb{R}^n$ und $\mathbf{x}^0 \in \mathbb{R}^n$ mit der Taylor-Formel 2. Ordnung in Landau-Symbolik

$$f(\mathbf{x}) = f(\mathbf{x}^0) + \nabla f(\mathbf{x}^0)^T (\mathbf{x} - \mathbf{x}^0) + \frac{1}{2} (\mathbf{x} - \mathbf{x}^0)^T \nabla^2 f(\mathbf{x}^0) (\mathbf{x} - \mathbf{x}^0) + o(\|\mathbf{x} - \mathbf{x}^0\|^2).$$

Approximiert man die Funktion f in einer Umgebung von \mathbf{x}^0 lokal durch ihr Taylor-Polynom 2. Grades mit der Entwicklungsstelle \mathbf{x}^0 gemäß

$$T_2(f, \mathbf{x}^0, \mathbf{x}) := f(\mathbf{x}^0) + \nabla f(\mathbf{x}^0)^T (\mathbf{x} - \mathbf{x}^0) + \frac{1}{2} (\mathbf{x} - \mathbf{x}^0)^T \nabla^2 f(\mathbf{x}^0) (\mathbf{x} - \mathbf{x}^0),$$

so lautet nach Satz 2.2 eine notwendige Bedingung für das Vorliegen einer Minimalstelle von $T_2(f, \mathbf{x}^0, \mathbf{x})$

$$\nabla T_2(f, \mathbf{x}^0, \mathbf{x}) = \nabla f(\mathbf{x}^0) + \nabla^2 f(\mathbf{x}^0) (\mathbf{x} - \mathbf{x}^0) = \mathbf{0}.$$

Ist die Hesse-Matrix $\nabla^2 f(\mathbf{x}^0)$ invertierbar, so ergibt sich

$$\mathbf{x} = \mathbf{x}^0 - (\nabla^2 f(\mathbf{x}^0))^{-1} \nabla f(\mathbf{x}^0).$$

Sukzessive Anwendung dieser Beziehung liefert, ausgehend von einem Startpunkt $\mathbf{x}^0 \in \mathbb{R}^n$, für $k \geq 0$ die Iterationsvorschrift des *Newton-Verfahrens*

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\nabla^2 f(\mathbf{x}^k))^{-1} \nabla f(\mathbf{x}^k).$$

In der algorithmischen Umsetzung des Newton-Verfahrens vermeidet man die explizite Berechnung der inversen Hesse-Matrix. Stattdessen wird in jedem Iterationsschritt zunächst eine Lösung $\mathbf{d}^k \in \mathbb{R}^n$ der *Newton-Gleichung* $\nabla^2 f(\mathbf{x}^k) \mathbf{d} = -\nabla f(\mathbf{x}^k)$ bestimmt und anschließend $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}^k$ gesetzt. Das so bestimmte \mathbf{d}^k wird als *Newton-Richtung* von f im Punkt \mathbf{x}^k bezeichnet. Somit lässt sich das Newton-Verfahren wie folgt formulieren:

Algorithmus 1 (Newton-Verfahren)

- S0** Wähle $\mathbf{x}^0 \in \mathbb{R}^n$, und setze $k := 0$.
- S1** Wenn $\nabla f(\mathbf{x}^k) = \mathbf{0}$, dann STOPP.
- S2** Bestimme eine Lösung \mathbf{d}^k der Newton-Gleichung $\nabla^2 f(\mathbf{x}^k) \mathbf{d} + \nabla f(\mathbf{x}^k) = \mathbf{0}$.
- S3** Setze $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}^k$ sowie $k := k + 1$, und gehe zu **S1**.

Es sei bemerkt, dass man das Newton-Verfahren auch als approximative Nullstellenbestimmung von ∇f durch das Taylor-Polynom 1. Grades von ∇f mit Entwicklungsstelle \mathbf{x}^0 deuten kann. Für den Beweis eines Konvergenzsatzes für das Newton-Verfahren benötigen wir das folgende Lemma.

Lemma 3.6

Es seien $f \in C^2(\mathbb{R}^n, \mathbb{R})$, $\mathbf{x}^* \in \mathbb{R}^n$ und $\nabla^2 f(\mathbf{x}^*)$ invertierbar. Dann existieren ein $\varepsilon > 0$ und eine Konstante $C > 0$, sodass für alle $\mathbf{x} \in U_\varepsilon(\mathbf{x}^*)$ die Hesse-Matrix $\nabla^2 f(\mathbf{x})$ ebenfalls invertierbar ist und außerdem $\|\nabla^2 f(\mathbf{x})^{-1}\| \leq C$ gilt.

Beweis: Wegen $f \in C^2(\mathbb{R}^n, \mathbb{R})$ existiert ein $\varepsilon > 0$ mit

$$\|\nabla^2 f(\mathbf{x}^*) - \nabla^2 f(\mathbf{x})\| \leq \frac{1}{2} \frac{1}{\|\nabla^2 f(\mathbf{x}^*)^{-1}\|}$$

für alle $\mathbf{x} \in U_\varepsilon(\mathbf{x}^*)$. Somit gilt

$$\begin{aligned} \|E_n - \nabla^2 f(\mathbf{x}^*)^{-1} \nabla^2 f(\mathbf{x})\| &= \|\nabla^2 f(\mathbf{x}^*)^{-1} (\nabla^2 f(\mathbf{x}^*) - \nabla^2 f(\mathbf{x}))\| \\ &\leq \|\nabla^2 f(\mathbf{x}^*)^{-1}\| \|\nabla^2 f(\mathbf{x}^*) - \nabla^2 f(\mathbf{x})\| \\ &\leq \frac{1}{2} \end{aligned}$$

ebenfalls für alle $\mathbf{x} \in U_\varepsilon(\mathbf{x}^*)$. Mit Lemma 1.24 (b) folgt die Invertierbarkeit von $\nabla^2 f(\mathbf{x})$ und

$$\|\nabla^2 f(\mathbf{x})^{-1}\| \leq \frac{\|\nabla^2 f(\mathbf{x}^*)^{-1}\|}{1 - \|E_n - \nabla^2 f(\mathbf{x}^*)^{-1} \nabla^2 f(\mathbf{x})\|} \leq 2 \|\nabla^2 f(\mathbf{x}^*)^{-1}\|$$

für alle $\mathbf{x} \in U_\varepsilon(\mathbf{x}^*)$. Mit $C := 2\|\nabla^2 f(\mathbf{x}^*)^{-1}\|$ ist die Aussage bewiesen. \square

Bezüglich der Konvergenz des Newton-Verfahrens gilt der folgende Satz.

Satz 3.7

Es seien $f \in C^2(\mathbb{R}^n, \mathbb{R})$, $\mathbf{x}^* \in \mathbb{R}^n$ ein stationärer Punkt von f und $\nabla^2 f(\mathbf{x}^*)$ invertierbar. Dann existiert eine ε -Umgebung $U_\varepsilon(\mathbf{x}^*)$, sodass für jeden Startpunkt $\mathbf{x}^0 \in U_\varepsilon(\mathbf{x}^*)$ das Newton-Verfahren durchführbar ist und die durch den Algorithmus 1 erzeugte Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ Q-superlinear gegen \mathbf{x}^* konvergiert. Gilt darüber hinaus, dass $\nabla^2 f(\mathbf{x}^*)$ in einer Umgebung von \mathbf{x}^* Lipschitz-stetig ist, dann konvergiert die durch den Algorithmus 1 erzeugte Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ Q-quadratisch gegen \mathbf{x}^* .

Beweis: Wegen $f \in C^2(\mathbb{R}^n, \mathbb{R})$ gilt $\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}) - \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x})\| = o(\|\mathbf{x} - \mathbf{y}\|)$ für beliebige $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. Mit der Iterationsvorschrift und $\nabla f(\mathbf{x}^*) = 0$ folgt unter der Voraussetzung der Durchführbarkeit des Newton-Verfahrens für alle $k \geq 0$

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^*\| &= \left\| \mathbf{x}^k - (\nabla^2 f(\mathbf{x}^k))^{-1} \nabla f(\mathbf{x}^k) - \mathbf{x}^* \right\| \\ &= \left\| \mathbf{x}^k - \mathbf{x}^* - \nabla^2 f(\mathbf{x}^k)^{-1} [\nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^*)] \right\| \\ &= \left\| \nabla^2 f(\mathbf{x}^k)^{-1} [\nabla f(\mathbf{x}^*) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^* - \mathbf{x}^k)] \right\| \\ &\leq \left\| \nabla^2 f(\mathbf{x}^k)^{-1} \right\| \left\| \nabla f(\mathbf{x}^*) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^* - \mathbf{x}^k) \right\| \\ &= \left\| \nabla^2 f(\mathbf{x}^k)^{-1} \right\| o(\|\mathbf{x}^k - \mathbf{x}^*\|). \end{aligned}$$

Wegen $f \in C^2(\mathbb{R}^n, \mathbb{R})$ und der Invertierbarkeit von $\nabla^2 f(\mathbf{x}^*)$ folgt mit Lemma 3.6 die Existenz eines $r_1 > 0$ und einer Konstanten $C > 0$, sodass für alle $\mathbf{x} \in U_{r_1}(\mathbf{x}^*)$ einerseits die Hesse-Matrix $\nabla^2 f(\mathbf{x})$ invertierbar ist und andererseits $\|\nabla^2 f(\mathbf{x})^{-1}\| \leq C$ gilt. Aufgrund von $\lim_{h \rightarrow 0} \frac{o(\|\mathbf{h}\|)}{\|\mathbf{h}\|} = 0$ gibt es ein $r_2 > 0$ mit $o(\|\mathbf{h}\|) \leq \frac{1}{2C}\|\mathbf{h}\|$ für alle \mathbf{h} mit $\|\mathbf{h}\| < r_2$. Mit $\varepsilon := \min\{r_1, r_2\}$ folgt für alle $\mathbf{x}^0 \in U_\varepsilon(\mathbf{x}^*)$

$$\|\mathbf{x}^1 - \mathbf{x}^*\| \leq C \frac{1}{2C} \|\mathbf{x}^0 - \mathbf{x}^*\| \leq \frac{1}{2} \varepsilon$$

und schließlich $\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq \frac{1}{2} \varepsilon$ für alle $k \geq 0$, womit die Durchführbarkeit des Newton-Verfahrens und wegen $\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq C o(\|\mathbf{x}^k - \mathbf{x}^*\|)$ auch die Q-superlineare Konvergenz der Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ für alle $\mathbf{x}^0 \in U_\varepsilon(\mathbf{x}^*)$ gezeigt ist. Existiert darüberhinaus ein $r_3 > 0$, sodass für alle $\mathbf{x} \in U_{r_3}(\mathbf{x}^*)$ die Hesse-Matrix $\nabla^2 f(\mathbf{x})$ Lipschitz-stetig mit Lipschitz-Konstante $L > 0$ ist, dann existiert wegen $\lim_{k \rightarrow \infty} \mathbf{x}^k = \mathbf{x}^*$ ein $k_0 \in \mathbb{N}$ mit $\mathbf{x}^k \in U_{\min\{\varepsilon, r_3\}}(\mathbf{x}^*)$, und für alle $k \geq k_0$ sind die Ungleichungen

$$\|\nabla f(\mathbf{x}^*) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^k - \mathbf{x}^*)\| \leq \frac{L}{2} \|\mathbf{x}^* - \mathbf{x}^k\|^2$$

bzw.

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq \frac{CL}{2} \|\mathbf{x}^* - \mathbf{x}^k\|^2$$

erfüllt. Damit ist auch die Q-quadratische Konvergenz der Folge $\{\mathbf{x}^k\}_{k \in \mathbb{N}}$ gezeigt. \square

Das Newton-Verfahren konvergiert unter den Voraussetzungen von Satz 3.7 generell *lokal* superlinear bzw. *lokal* quadratisch gegen stationäre Punkte, die natürlich auch Maximalstellen sein können. Somit kann im Verlauf des Newton-Verfahrens $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$ für $k = 0, 1, 2, \dots$ nicht garantiert werden. Weitere Nachteile des Newton-Verfahrens bestehen einerseits im hohen rechentechnischen Aufwand, da in jedem Iterationsschritt eine Hesse-Matrix berechnet werden muss, und andererseits in dem Sachverhalt, dass die zur Bestimmung des $(k + 1)$ -ten Iterationspunktes zu lösende Newton-Gleichung unlösbar sein kann. Der hohe rechentechnische Aufwand wurde durch die Entwicklung der AD relativiert, sodass es heute durchaus möglich ist, Optimierungsprobleme hoher Dimension effektiv durch Verfahren zu lösen, welche Hesse-Matrizen benutzen. Im Fall der Nichtlösbarkeit der Newton-Gleichung bietet sich eine „least square“-Lösung der Newton-Gleichung an, um das Newton-Verfahren fortführen zu können. Im Rahmen der aufgeführten Konvergenzaussagen ist dieser so ermittelte Iterationspunkt als neuer Startpunkt des Newton-Verfahrens aufzufassen.

3.2.1 Numerische Experimente zum Newton-Verfahren

Experiment 3.2.1 (Quadratische Konvergenzgeschwindigkeit des Newton-Verfahrens)

Newton01.m: Wir betrachten die konvexe und beliebig oft stetig partiell differenzierbare Funktion (Problem Nr. 13) $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ mit

$$f(\mathbf{x}) = (x_1 - 2)^4 + x_2^2(x_1 - 2)^2 + (x_2 + 1)^2,$$

globaler Minimalstelle $\mathbf{x}^* = (2, -1)^T$ und $f(\mathbf{x}^*) = 0$. Zur Minimierung dieser Funktion wenden wir das Newton-Verfahren mit Startpunkt $\mathbf{x}^0 = (1, 1)^T$ und Abbruchtoleranz $\|\nabla f(\mathbf{x}^k)\| \leq 10^{-6}$ an. Da eine in einer beschränkten Umgebung von \mathbf{x}^* dreimal stetig differenzierbare Funktion dort auch eine Lipschitz-stetige 2. Ableitung besitzt, sind nach Satz 3.7 in einer (hinreichend kleinen) Umgebung von \mathbf{x}^* alle Voraussetzungen für eine quadratische Konvergenz der Iterierten \mathbf{x}^k bei Anwendung des Newton-Verfahrens erfüllt. Die Spalten der von EDOPTLAB erzeugten Tabelle 3.3 geben den Iterationsverlauf des Newton-Verfahrens wieder. Dabei bezeichnen die Spalte „*iter*“ die Iterationsnummer, die Spalte „*xiter*(1)“ bzw. „*xiter*(2)“ die 1. bzw. 2. Koordinaten der Iterationspunkte und die Spalte „*fiter*“ die zugehörigen Funktionswerte. In der Spalte „*fiter*“ ist die quadratische Konvergenz der Funktionswerte des Verfahrens gegen den optimalen Zielfunktionswert unmittelbar ersichtlich (Verdopplung der Anzahl der führenden Nullen ab der 4. Iteration bei jedem weiteren Iterationsschritt). Die Abbildung 3.6 illustriert den Iterationsverlauf und die Norm der zugehörigen Gradienten. \blacksquare

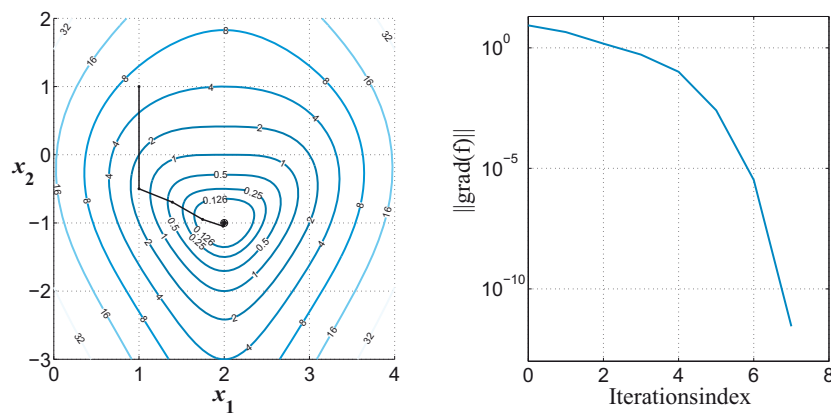


Abb. 3.6 Iterationsverlauf des Newton-Verfahrens im Exp. 3.2.1, quadratische Konvergenz

iter	xiter (1)	xiter (2)	fiter
0	1.0000000000	1.0000000000	6.0000000000e+000
1	1.0000000000	-0.5000000000	1.5000000000e+000
2	1.3913043478	-0.6956521739	4.0920737133e-001
3	1.7459441208	-0.9487980942	6.4891623477e-002
4	1.9862783400	-1.0482080866	2.5309302111e-003
5	1.9987342021	-1.0001699932	1.6316892669e-006
6	1.9999995657	-1.0000016017	2.7540453499e-012
7	2.0000000000	-1.0000000000	1.9714253277e-024

Tab. 3.3 Iterationsverlauf des Newton-Verfahrens im Exp. 3.2.1, quadratische Konvergenz

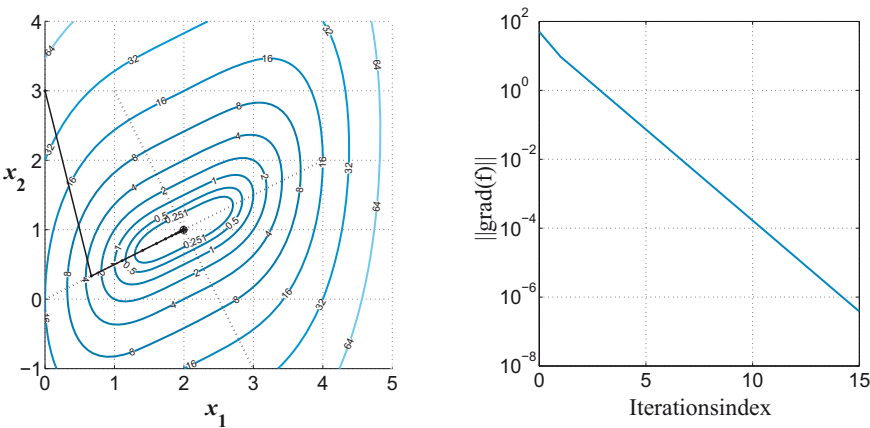
Experiment 3.2.2 (Lineare Konvergenzgeschwindigkeit des Newton-Verfahrens)

Newton02.m: Wir untersuchen die konvexe und beliebig oft stetig partiell differenzierbare Funktion (Problem Nr. 9) $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ mit

$$f(x) = (x_1 - 2)^4 + (x_1 - 2x_2)^2,$$

globaler Minimalstelle $x^* = (2, 1)^T$ und $f(x^*) = 0$. Das Newton-Verfahren wird mit Startpunkt $x^0 = (0, 3)^T$ und Abbruchtoleranz $\|\nabla f(x^k)\| \leq 10^{-6}$ angewendet. Die Hesse-Matrix $\nabla^2 f(x^*) = \begin{pmatrix} 2 & -4 \\ -4 & 8 \end{pmatrix}$ mit den Eigenwerten $\lambda_1 = 0$ und $\lambda_2 = 10$ ist nicht regulär, womit die Voraussetzungen des Satzes 3.7 für eine superlineare Konvergenz der Iterierten verletzt sind. In Tab. 3.4 bezeichnen „nf“, „ng“ und „nh“ die bis zur Iteration „iter“ erfolgte kumulierte Anzahl von Funktionswert-, Gradienten- und Hesse-Matrixberechnungen. Die Nullen in den Spalten „nd“ (steht für „non descent“) bzw. „LS“ (steht für „least square“) zeigen, dass in jeder Iteration $f(x_{iter+1}) \leq f(x_{iter})$ gilt bzw. dass in jeder Iteration die Newton-Gleichung lösbar ist, und damit keine approximative

Lösung mittels „least square“-Ansatz bestimmt wurde. In der Spalte „norm(g)“ ist die eu-
klidische Norm des Gradienten der Zielfunktion im jeweiligen Iterationspunkt aufgeführt.
Abb. 3.7 veranschaulicht den Iterationsverlauf. Man erkennt, dass das Newton-Verfahren
nach einer Iteration näherungsweise einen Punkt auf der Achse zur Eigenrichtung mit dem
Eigenwert $\lambda_1 = 0$ der Hesse-Matrix $\nabla^2 f(\mathbf{x}^*)$ (grau gestrichelte Linie) erreicht und dann
langsam entlang dieser Richtung $(2, 1)^T$ gegen den Minimalpunkt mit linearer Konvergenz-
geschwindigkeit bzgl. der Funktionswerte und bzgl. der Norm des Gradienten strebt
(siehe auch Tab. 3.4). ■



Experiment 3.2.3 (Mögliche Divergenz des Newton-Verfahrens)

Newton03.m: Die Funktion (Problem Nr. 21) $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ mit

$$f(\mathbf{x}) = |x_1| - \ln(1 + |x_1|) + |x_2| - \ln(1 + |x_2|)$$

ist zweimal Lipschitz-stetig differenzierbar für alle $\mathbf{x} \in \mathbb{R}^2$ und außerdem streng konvex. Nach Satz 3.7 folgt (bei exakter Rechnung) die lokale quadratische Konvergenz des Newton-Verfahrens gegen die globale Minimalstelle $\mathbf{x}^* = (0, 0)^T$. Insbesondere gilt für die Koordinaten x_i mit $i \in \{1, 2\}$ im Verlauf des Newton-Verfahrens:

- Ist $x_i^0 = -1$ bzw. $x_i^0 = 1$, so folgt $x_i^k = (-1)^{k+1}$ bzw. $x_i^k = (-1)^k$.
- Ist $|x_i^0| < 1$, so folgt mit quadratischer Konvergenz $\lim_{k \rightarrow \infty} x_i^k = 0$ (oszillierend).
- Ist $|x_i^0| > 1$, so folgt $\lim_{k \rightarrow \infty} |x_i^k| = \infty$ (oszillierend).

Zum Nachweis dieser Aussagen sei auf Aufgabe 3.6 verwiesen. Wir demonstrieren dieses Verhalten des Newton-Verfahrens für die drei Startpunkte $\mathbf{x}^0 = (1, -1)^T$ (siehe Abb. 3.8), $\mathbf{x}^0 = (1, -0.97)^T$ (siehe Abb. 3.9) und $\mathbf{x}^0 = (1.01, -0.97)^T$ (siehe Abb. 3.10), indem wir jeweils den Iterationsverlauf von x_1^k und x_2^k für die ersten acht Iterationen darstellen. Zusammenfassend stellen wir also fest, dass trotz der strengen Konvexität der Funktion f das Newton-Verfahren außerhalb einer gewissen Umgebung des Minimalpunktes (hier also $\|\mathbf{x}^0\| < 1$) divergiert und nur bei Startpunkten innerhalb dieser Umgebung konvergent ist. ■

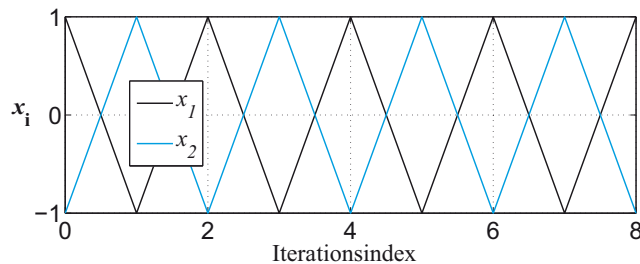


Abb. 3.8 Divergenz des Newton-Verfahrens mit Oszillation $x_i^k \in \{-1, 1\}$ für $i = 1, 2$ im Exp. 3.2.3 bei Startpunkt $\mathbf{x}^0 = (1, -1)^T$

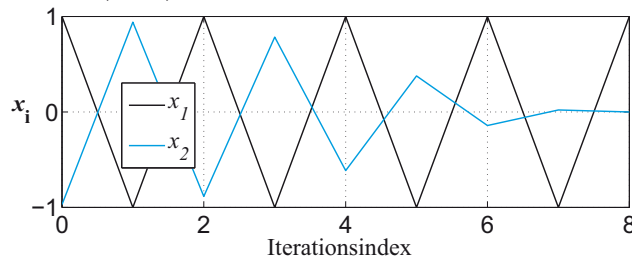


Abb. 3.9 Divergenz des Newton-Verfahrens mit Oszillation $x_1^k \in \{-1, 1\}$ und Konvergenz für x_2^k im Exp. 3.2.3 bei Startpunkt $\mathbf{x}^0 = (1, -0.97)^T$

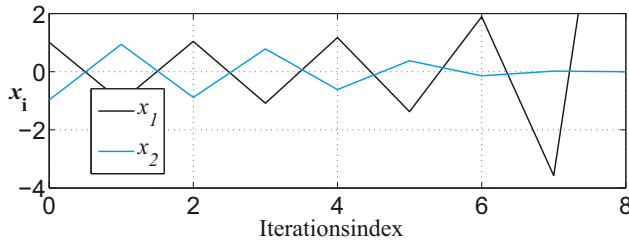


Abb. 3.10 Divergenz des Newton-Verfahrens mit Oszillation $\lim_{k \rightarrow \infty} |x_1^k| = \infty$ und Konvergenz für x_2^k im Exp. 3.2.3 bei Startpunkt $\mathbf{x}^0 = (1.01, -0.97)^T$

Experiment 3.2.4 („Theorie vs. Praxis“)

Newton04.m: Wir betrachten die zweimal stetig differenzierbare Funktion (Problem Nr. 19) $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ mit

$$f(\mathbf{x}) = (2x_1^2 + x_2^2)^2 + (x_1^2 + 2x_2^2)^{\frac{5}{4}},$$

globaler Minimalstelle $\mathbf{x}^* = (0, 0)^T$ und $f(\mathbf{x}^*) = 0$. Die 2. Ableitung im Optimalpunkt \mathbf{x}^* ist nicht Lipschitz-stetig. Nach Satz 3.7 folgt in einer (hinreichend kleinen) Umgebung von \mathbf{x}^* bei exakter Arithmetik superlineare Konvergenz der Iterierten, und nach den Ausführungen in Abschnitt 3.1.1 gilt $\lim_{k \rightarrow \infty} c_k = 0$ für $c_k := \frac{\|\nabla f(\mathbf{x}^{k+1})\|}{\|\nabla f(\mathbf{x}^k)\|}$. Die folgenden Abbildungen zeigen, dass für das Newton-Verfahren mit Startpunkt $\mathbf{x}^0 = (2, 2)^T$ und gewählter Abbruchbedingung $\|\nabla f(\mathbf{x}^k)\| \leq 10^{-16}$ eine superlineare Tendenz nur bis zu einer Genauigkeit von $\|\nabla f(\mathbf{x}^k)\| \geq 10^{-3}$, d. h. bis zur 8. Iteration (siehe Abb. 3.12), erkennbar ist. Der Grund liegt in den numerischen Ungenauigkeiten bei der Berechnung der Hesse-Matrix, die sowohl bei der symbolischen als auch bei der automatischen Differenziation für $\mathbf{x} \rightarrow \mathbf{0}$ einen „ $\frac{0}{0}$ “-Term enthält. Als Konsequenz wird nur lineare Konvergenz mit dem Konvergenzfaktor c_k von etwa 0.06 erzielt (siehe Abb. 3.11). Wir sehen, dass selbst die Berechnung der ersten und zweiten partiellen Ableitungen mithilfe der automatischen bzw. symbolischen Differenziation im Rahmen der 16-stelligen Gleitkommagenauigkeit unter MATLAB nicht ausreicht, um die an sich theoretisch vorhandene superlineare Konvergenz bis zu einer (moderateren) Abbruchbedingung von $\|\nabla f(\mathbf{x}^k)\| \leq 10^{-8}$ zu bestätigen, wenn die Formeln für die Ableitungen nicht sachgerecht vereinfacht werden. Die Tatsache, dass die theoretisch geltende Konvergenzgeschwindigkeit numerisch nicht erreicht wird, tritt sehr häufig bei praktischen Optimierungsproblemen auf. Deshalb sollten die Abbruchschranken nicht zu klein gewählt werden, insbesondere dann, wenn die Ergebnisse nicht mit hoher Genauigkeit benötigt werden. ■

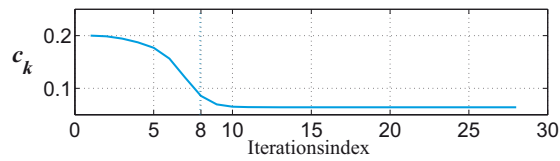


Abb. 3.11 Keine (praktische) superlineare Konvergenz des Newton-Verfahrens im Exp. 3.2.4

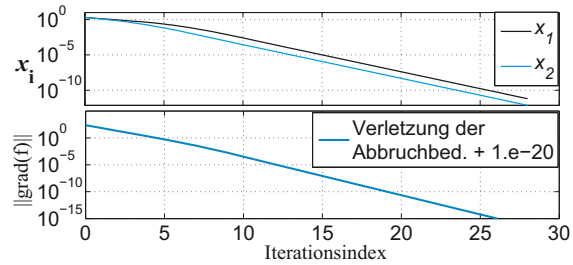


Abb. 3.12 Lineare Konvergenz des Newton-Verfahrens im Exp. 3.2.4

Experiment 3.2.5 (Nichtmonotonie des Newton-Verfahrens)

Newton05.m: Wir betrachten die dreidimensionale Rosenbrock-Funktion (Problem Nr. 50, Dimension $n = 3$) gemäß (3.3) mit globaler Minimalstelle $\mathbf{x}^* = (1, 1, 1)^T$ und $f(\mathbf{x}^*) = 0$. Das Newton-Verfahren mit Startpunkt $\mathbf{x}^0 = (-1.2, 1, -1.2)^T$ und Abbruchbedingung $\|\nabla f(\mathbf{x}^k)\| \leq 10^{-16}$ liefert $f(\mathbf{x}^{k+1}) > f(\mathbf{x}^k)$ für $k = 5$ (in Abb. 3.13 nicht erkennbar), $k = 7$ und $k = 9$. Ab $k = 9$ gilt $f(\mathbf{x}^{k+1}) \leq 70f(\mathbf{x}^k)^2$ und $\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq 30\|\mathbf{x}^k - \mathbf{x}^*\|^2$, d. h. im Rahmen unserer numerischen Genauigkeit liegt sogar quadratische Konvergenz bezüglich der Funktionswerte und Iterationspunkte vor. ■

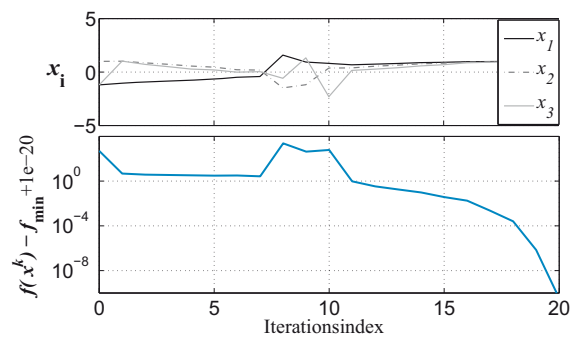


Abb. 3.13 Nichtmonotonie des Newton-Verfahrens im Exp. 3.2.5

Experiment 3.2.6 (Konvergenz gegen stationäre Punkte)

Newton06.m: Wie bereits erwähnt, konvergiert das Newton-Verfahren nicht nur gegen

lokale Minimalstellen, sondern auch gegen ggf. existierende andere stationäre Punkte. Wir betrachten die beliebig oft stetig partiell differenzierbare Funktion (Problem Nr. 6) $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ mit

$$f(\mathbf{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$

Diese Funktion besitzt vier lokale Minimalstellen, eine lokale Maximalstelle und vier Sattelpunkte. In Abhängigkeit vom Startpunkt \mathbf{x}^0 konvergiert das Newton-Verfahren gegen jeden dieser neun stationären Punkte (siehe Abb. 3.14). ■

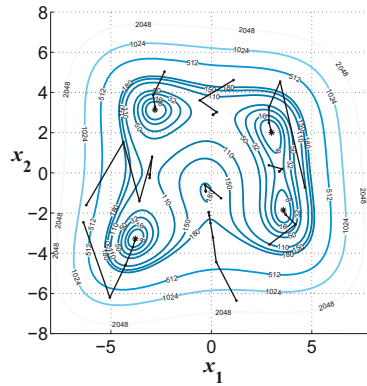


Abb. 3.14 Konvergenz des Newton-Verfahrens gegen stationäre Punkte im Exp. 3.2.6

3.3 Ein allgemeines Abstiegsverfahren mit Richtungssuche

Wie bereits aufgeführt kann im Verlauf des Newton-Verfahrens auch für konvexe Funktionen $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$ nicht garantiert werden. Verfahren, die diese Eigenschaft gewährleisten, werden als *Abstiegsverfahren* bezeichnet, und basieren bei den zunächst betrachteten Verfahren auf dem folgenden Begriff:

Definition 3.8

Es seien $f : \mathbb{R}^n \rightarrow \mathbb{R}$ und $\mathbf{x} \in \mathbb{R}^n$. Ein Vektor $\mathbf{d} \in \mathbb{R}^n$ heißt *Abstiegsrichtung* von f in \mathbf{x} , wenn ein $t_0 > 0$ existiert mit $f(\mathbf{x} + t\mathbf{d}) < f(\mathbf{x})$ für alle $t \in (0, t_0]$.

Ein allgemeines Abstiegsverfahren mit Richtungssuche kann wie folgt formuliert werden:

Nichtlineare Optimierung

Theorie, Numerik und Experimente

Reinhardt, R.; Hoffmann, A.; Gerlach, T.

2013, X, 383 S. 97 Abb. in Farbe., Hardcover

ISBN: 978-3-8274-2948-3