9 Nichtlineare Ausgleichsprobleme und Parameteridentifikation

A. Problemstellung

Zur Beschreibung der allgemeinen Problemstellung beginnen wir mit einem Parameteridentifikationsproblem aus der Pharmakokinetik.

Beispiel (9.1) (nach C. Geiger)
Zur Steuerung des Blutspiegels während einer medikamentösen Behandlung ist es notwendig, die Abhängigkeit des Blutspiegels (hier des sogenannten Eliminationsanteils) von den pharmakokinetischen Parametern zu kennen. In einem mathematischen Modell kann dieser Eliminationsanteil als Lösung der folgenden inhomogenen, linearen Differentialgleichung beschrieben werden

$$y'(t) \;=\; a\,k_1 \exp(-k_1\,t) \;-\; k_2\,y(t)\,.$$

Mit dem Anfangswert $y(0) = 0$ ergibt sich die Lösung

$$y(t) \;=\; \begin{cases} \dfrac{a\,k_1}{k_2 - k_1}\,(\exp(-k_1\,t) - \exp(-k_2\,t))\,, & \text{falls } k_1 \neq k_2 \\ a\,k_1\,t\exp(-k_1\,t), & \text{falls } k_1 = k_2\,. \end{cases} \tag{9.2}$$

Sinnvoll sind hierbei Parameter mit $a > 0$ und $0 < k_1 \leq k_2$. Eine durch (9.2) gegebene Funktion heißt auch Batman-Funktion. Sie hat in etwa die in Abbildung 9.1 angegebene Gestalt.

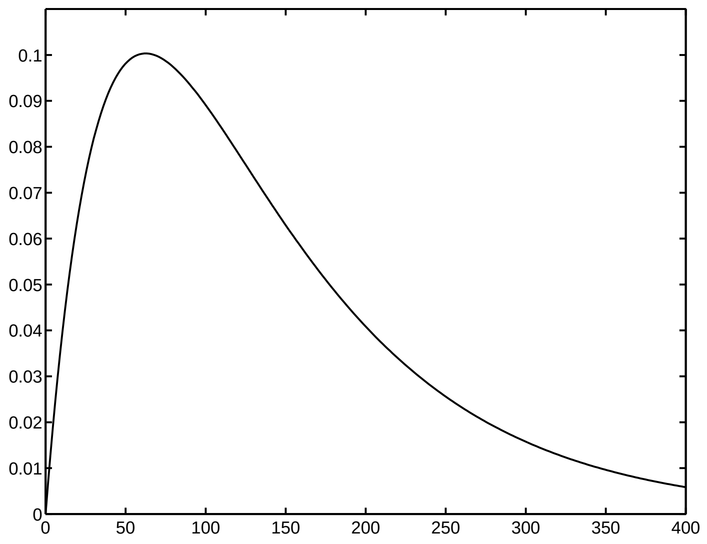

Abb. 9.1. Batman-Funktion für $a = 0.45,\ k_1 = 0.01,\ k2 = 0.024$

Zu gegebenen Meßwerten (t_i, y_i), $i = 1, \ldots, m$ sollen nun die Parameter a, k_1, k_2 so bestimmt werden, daß die Lösung $y(t; a, k_1, k_2)$ das überbestimmte (nichtlineare) Gleichungssystem

$$y(t_i; a, k_1, k_2) \;=\; y_i, \;\; i = 1, \ldots, m \tag{9.3}$$

möglichst gut erfüllt. (Eine exakte Lösung ist wegen $m \gg 3$ i.a. ausgeschlossen.)

Üblich ist es, dieses überstimmte Gleichungssystem nach der **Methode der kleinsten Fehlerquadrate** zu lösen (nach Carl Friedrich Gauß, 1802), d.h. die Parameter werden so bestimmt, daß die Funktion

$$f(a, k_1, k_2) \;:=\; \sum_{i=1}^{m} (\, y(t_i; a, k_1, k_2) - y_i \,)^2$$

minimal wird.

Anders formuliert: Definiert man den **Residuenvektor** $\boldsymbol{r}(a, k_1, k_2) \in \mathbb{R}^m$ durch

$$r_i(a, k_1, k_2) \;:=\; y(t_i; a, k_1, k_2) - y_i \,, \;\; i = 1, \ldots, m \,,$$

so sind die Parameter a, k_1, k_2 so zu bestimmen, daß die euklidische Norm des Residuenvektors $\|\boldsymbol{r}(a, k_1, k_2)\|_2$ minimal wird.

Wir sind damit bei folgender Formulierung für ein allgemeines Ausgleichsproblem angelangt:

Allgemeines Ausgleichsproblem (9.4)

Gegeben sei eine (hinreichend glatte) Funktion $\boldsymbol{r} : \mathbb{R}^n \to \mathbb{R}^m$, wobei $m \geq n$.
Gesucht ist ein Punkt $\boldsymbol{x}^* \in \mathbb{R}^n$, so daß $\|\boldsymbol{r}(\boldsymbol{x}^*)\|_2 \;\leq\; \|\boldsymbol{r}(\boldsymbol{x})\|_2$ für alle $\boldsymbol{x} \in \mathbb{R}^n$ gilt.

Hängt $\boldsymbol{r}$ affin–linear von $\boldsymbol{x}$ ab, d.h. gilt $\boldsymbol{r}(\boldsymbol{x}) \;=\; \boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}$, mit $\boldsymbol{A} \in \mathbb{R}^{(m,n)}$ und $\boldsymbol{b} \in \mathbb{R}^m$, so spricht man von einer **linearen Ausgleichsaufgabe**, andernfalls von einer **nichtlinearen Ausgleichsaufgabe**.

Das obige Beispiel (9.1) ist nichtlinear, vgl. dazu (9.2). Ein weiteres Beispiel haben wir schon wiederholt untersucht: die sogenannte Rosenbrock-Funktion, vgl. Arbeitsblatt2 und folgende. Hier ist $\boldsymbol{r} = (10(y - x^2), 1 - x)^{\mathrm{T}}$, es handelt sich also ebenfalls um eine nichtlineare Ausgleichsaufgabe. Wir sehen, daß die Herkunft aus einer Aufgabe der Parameteridentifikation nicht so entscheidend ist. Wesentlich ist vielmehr, daß eine Quadratsumme minimiert werden soll.

Natürlich könnte man daran denken, auch andere Normen, etwa die sogenannte Tschebyscheff-Norm $\|\boldsymbol{x}\|_\infty := \max |x_i|$ zu verwenden. Dieses Problem ist dann jedoch i.allg. wesentlich schwieriger zu lösen; hierzu sei auf Lehrbücher über Approximationstheorie verwiesen.

Im folgenden wollen wir die wesentlichen Aussagen und numerischen Ansätze für linearen Ausgleichsprobleme zusammenstellen. Wir können hier nur einen kurzen Überblick vermitteln ohne auf die numerischen Feinheiten eingehen zu können.

B. Lineare Ausgleichsprobleme

Betrachtet wird eine lineare Ausgleichsaufgabe der Form

$$\text{Minimiere } \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}\|_2, \;\; \boldsymbol{x} \in \mathbb{R}^n \tag{9.5}$$

mit $\boldsymbol{A} \in \mathbb{R}^{(m,n)}$, $\boldsymbol{b} \in \mathbb{R}^m$ und $m \geq n$.

Ein Vektor $\boldsymbol{x}^* \in \mathbb{R}^n$ löst genau dann die Optimierungsaufgabe (9.5), falls die **Normalgleichung**

$$\boldsymbol{A}^{\mathrm{T}}\,\boldsymbol{A}\,\boldsymbol{x}^* = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} \tag{9.6}$$

erfüllt ist.

Beweis: Für $\boldsymbol{r}(\boldsymbol{x}) := \boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}$ rechnet man leicht explizit aus

$$\|\boldsymbol{r}(\boldsymbol{x} + \boldsymbol{\Delta}\,\boldsymbol{x})\|_2^2 = \|\boldsymbol{r}(\boldsymbol{x})\|_2^2 + 2\,\boldsymbol{\Delta}\,\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}}\,\boldsymbol{r}(\boldsymbol{x})) + \|\boldsymbol{A}\,\boldsymbol{\Delta}\,\boldsymbol{x}\|_2^2\,,$$

so daß $\|\boldsymbol{r}(\boldsymbol{x})\|_2$ in der Tat minimal wird, wenn $\boldsymbol{A}^{\mathrm{T}}\,\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{0}$ gilt. Dies ist gerade (9.6). □

(9.6) ist ein quadratisches lineares Gleichungssystem der Dimension n. Hat die Matrix $\boldsymbol{A}$ maximalen Rang ($= n$), so ist die (symmetrische) Koeffizientenmatrix $\boldsymbol{A}^{\mathrm{T}}\,\boldsymbol{A}$ regulär und positiv definit. Das lineare Gleichungssystem (9.6) besitzt also eine eindeutig bestimmte Lösung.

Eine numerische Lösung von (9.6) – etwa mittels Cholesky–Zerlegung der Matrix $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ (das ist eine Variante des Gaußschen Eliminationsverfahrens für symmetrische Matrizen) – ist häufig numerisch instabil, da das lineare Gleichungssytem (9.6) zumeist schlechter konditioniert ist als das Ausgangsproblem.

Eine stabile numerische Lösung von (9.6) erfolgt daher zweckmäßig durch eine sogenannte ***QR*–Zerlegung** der Matrix $\boldsymbol{A}$. Dies ist eine Zerlegung der Form $\boldsymbol{A}\,\boldsymbol{P} = \boldsymbol{Q}\,\boldsymbol{R}$, wobei $\boldsymbol{P}$ eine Permutationsmatrix, $\boldsymbol{Q}$ eine orthogonale Matrix und $\boldsymbol{R}$ eine obere Dreiecksmatrix ist. Eine solche QR–Zerlegung findet man durch Multiplikation von $\boldsymbol{A}$ mit geeigneten orthogonalen Matrizen. Zumeist werden hierzu sogenannte Householder-Matrizen verwendet. Diese beschreiben geometrisch eine Spiegelung des $\mathbb{R}^n$ an einer Hyperebene. Da orthogonale Transformationen die Länge von Vektoren nicht ändern, folgt

$$\|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}\|_2 = \|\boldsymbol{R}\,\boldsymbol{P}^{\mathrm{T}}\,\boldsymbol{x} - \boldsymbol{Q}^{\mathrm{T}}\,\boldsymbol{b}\|_2,$$

so daß für die Bestimmung von $\boldsymbol{x}$ lediglich ein lineares Gleichungssystem mit einer oberen Dreiecksmatrix zu lösen ist.

In MATLAB läuft dieser Algorithmus bei der einfachen Anweisung $\boldsymbol{A}\backslash\boldsymbol{b}$ ab; für genauere Informationen informiere man sich über die MATLAB Anweisung **qr(X)**.

Es sei noch darauf aufmerksam gemacht, daß der Algorithmus nicht geeignet ist, falls die Matrix $\boldsymbol{A}$ nicht maximalen Rang hat (das Ausgleichsproblem ist in diesem Fall nicht eindeutig lösbar.) In einer derartigen Situation kann man anstelle einer QR-Zerlegung eine sogenannte Singulärwertzerlegung verwenden. Auch hierfür bietet MATLAB professionelle Software an; man informiere sich über die MATLAB Anweisung **svd(X)**.

Beispiel (9.7) (nach N. Köckler)
Die Bahnparameter einer Ellipse

$$\xi^2 \;=\; a\,\eta^2 + b\,\xi\,\eta + c\,\xi + d\,\eta + e$$

sind an gemessene Positionsdaten eines Himmelskörpers anzupassen. Die Meßdaten sind in der folgenden Tabelle angegeben.

Tabelle 9.1. : Meßdaten zu Beispiel 9.7

k	xi_k	eta_k
1	-1.024940	-0.389269
2	-0.949898	-0.322894
3	-0.866114	-0.265256
4	-0.773392	-0.216557
5	-0.671372	-0.177152
6	-0.559524	-0.147582
7	-0.437067	-0.128618
8	-0.302909	-0.121353
9	-0.155493	-0.127348
10	-0.007464	-0.148885

In der Normalform (9.5) lautet das lineare Ausgleichsproblem nun mit $m := 10$:

$$\begin{aligned} a(k,\cdot) &= (\eta_k^2,\ \xi_k\,\eta_k,\ \xi_k,\ \eta_k,\ 1), \quad && k = 1,\dots,m \\ b(k) &= \xi_k^2\,, && k = 1,\dots,m \\ \boldsymbol{x} &= (a,\ b,\ c,\ d,\ e)^{\mathrm{T}} \end{aligned}$$

Die entsprechende Besetzung von $\boldsymbol{A}$, $\boldsymbol{b}$ und der MATLAB Aufruf $\boldsymbol{x} = \boldsymbol{A} \setminus \boldsymbol{b}$ liefert das Ergebnis

$$\boldsymbol{x} \;=\; \begin{pmatrix} -1.383348865120148E+000 \\ -6.646496504868814E-001 \\ -6.711285453952144E-001 \\ -3.370907563742522E+000 \\ -4.750421470686427E-001 \end{pmatrix}.$$

In der Abbildung 9.2 ist die resultierende Ausgleichsellipse zusammen mit den Meßdaten dargestellt. Mit + ist der Mittelpunkt der Ellipse eingezeichnet, • ist der rechte Brennpunkt (Sonne).

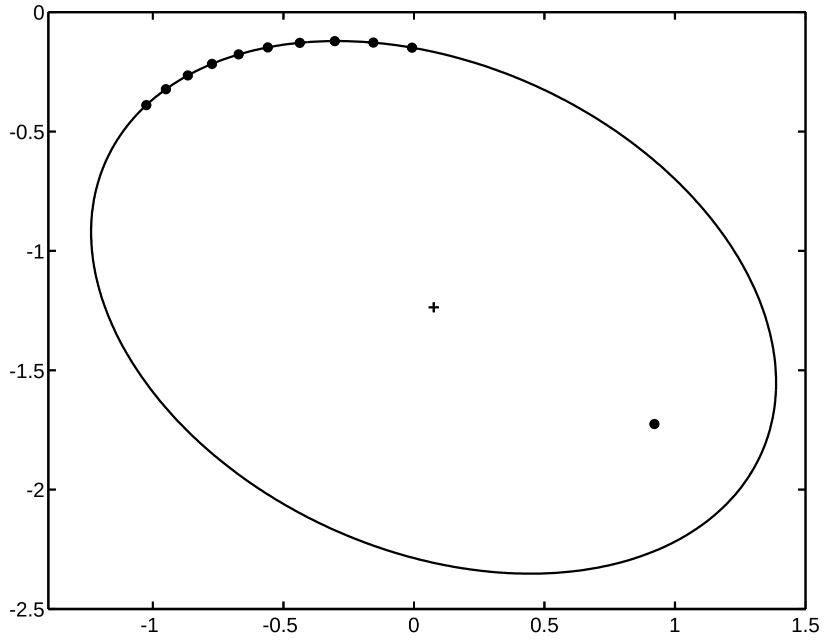

Abb. 9.2. Ausgleichsellipse von Beispiel (9.7)

C. Nichtlineare Ausgleichsprobleme

Wir betrachten nun allgemeine nichtlineare Ausgleichsaufgaben des Typs (9.4). Der Einfachheit halber minimieren wir das Quadrat der Euklidischen Norm, d.h die Funktion

$$f(\boldsymbol{x}) \;=\; \boldsymbol{r}(\boldsymbol{x})^{\mathrm{T}}\, \boldsymbol{r}(\boldsymbol{x}) \;=\; \sum_{i=1}^{m} \left(r_i(\boldsymbol{x})\right)^2 . \tag{9.8}$$

Dies ist nun eine unrestringierte Optimierungsaufgabe und wir können (hinreichende Glattheit vorausgesetzt) das Newton–Verfahren auf dieses Problem anwenden. Der k–te Iterationsschritt würde dann wir folgt aussehen

$$\begin{aligned} \boldsymbol{\nabla}^2 f(\boldsymbol{x}^{(k)})\, \boldsymbol{\Delta x}^{(k)} &= -\boldsymbol{\nabla} f(\boldsymbol{x}^{(k)}) \\ \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \alpha_k\, \boldsymbol{\Delta x}^{(k)} . \end{aligned} \tag{9.9}$$

Hierbei bezeichnet α_k eine geeignete Schrittlänge (vgl. Abschnitt 6).

Unter Ausnutzung der speziellen Struktur (9.8) von f lassen sich die Ableitungen (Gradient und Hesse-Matrix) explizit durch die Ableitungen von $\boldsymbol{r}$ ausdrücken. Man erhält hierfür

$$\begin{aligned} \boldsymbol{\nabla} f(\boldsymbol{x}) &= 2\, \boldsymbol{r}'(\boldsymbol{x})\, \boldsymbol{r}(\boldsymbol{x}) \\ \boldsymbol{\nabla}^2 f(\boldsymbol{x}) &= 2\, \boldsymbol{r}'(\boldsymbol{x})^{\mathrm{T}}\, \boldsymbol{r}'(\boldsymbol{x}) + 2 \sum_{i=1}^{m} r_i(\boldsymbol{x})\, \boldsymbol{\nabla}^2 r_i(\boldsymbol{x}) . \end{aligned} \tag{9.10}$$

Hierbei ist $\boldsymbol{r}'(\boldsymbol{x})$ die Jacobi-Matrix von $\boldsymbol{r}$ (Matrix der ersten Ableitungen) und $\boldsymbol{\nabla}^2 r_i(\boldsymbol{x})$ die Hesse-Matrix der i–ten Koordinatenfunktion von $\boldsymbol{r}$, $i = 1, \ldots, m$.

Man sieht insbesondere, daß die Hesse-Matrix sämtliche zweiten partiellen Ableitungen von $\boldsymbol{r}$ enthält; dies sind $m\, n^2/2$ Funktionsauswertungen für jeden Newton-Schritt. Andererseits werden die zweiten partiellen Ableitungen mit den $r_i(\boldsymbol{x})$ multipliziert und für ein sachgerechtes Modell ist zu erwarten, daß die r_i klein sind. Es ist daher naheliegend, den zweiten Summanden (und damit sämtliche zweiten Ableitungen!) in (9.10) zu vernachlässigen. Setzt man den so vereinfachten Ausdruck für die Hesse–Matrix von f in die Newton–Iteration (9.9) ein, so erhält man das lineare Gleichungssytem

$$\boldsymbol{r}'(\boldsymbol{x}^{(k)})^{\mathrm{T}}\, \boldsymbol{r}'(\boldsymbol{x}^{(k)})\, \boldsymbol{\Delta x}^{(k)} \;=\; -\boldsymbol{r}'(\boldsymbol{x}^{(k)})^{\mathrm{T}}\, \boldsymbol{r}(\boldsymbol{x}^{(k)})\,. \tag{9.11}$$

Für den Fall $n = m$ (d.h. genau so viele Messungen wie zu bestimmende Parameter) ist (9.11) gerade die mit $\boldsymbol{r}'(\boldsymbol{x}^{(k)})^{\mathrm{T}}$ multiplizierte Newton–Gleichung zur Bestimmung einer Nullstelle des nichtlinearen Gleichungssystems $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{0}$.

Für den „Normalfall“ $m > n$ hat (9.11) genau die Struktur der (i. allg. schlecht konditionierten) Normalgleichung (9.6). Mit anderen Worten, die Lösung $\boldsymbol{\Delta x}^{(k)}$ von (9.11) ist zugleich Lösung der **linearen Ausgleichsaufgabe**

$$\text{Minimiere } \|\boldsymbol{r}'(\boldsymbol{x}^{(k)})\, \boldsymbol{\Delta x} \;+\; \boldsymbol{r}(\boldsymbol{x}^{(k)})\|_2, \;\; \boldsymbol{\Delta x} \in \mathbb{R}^n \tag{9.12}$$

Aus Gründen der numerischen Stabilität ist es nun zu empfehlen, die Korrektur $\boldsymbol{\Delta x}_k$ direkt aus (9.12) mittels Orthogonalzerlegung (QR–Zerlegung) zu berechnen. Das **Gauß–Newton Verfahren** ist somit durch den folgenden Algorithmus gegeben:

Algorithmus (Gauß–Newton–Verfahren) (9.13)

(i) $\boldsymbol{x}_0 \in \mathbb{R}^n$: Startvektor

(ii) Für $k = 0, 1, \ldots$:

1.) Bestimmung von $\boldsymbol{\Delta x}_k$ als Lösung des linearen Ausgleichsproblems (9.12).
2.) Bestimmung einer Schrittlänge $\alpha_k > 0$ mit $\|\boldsymbol{r}(\boldsymbol{x}_k + \alpha_k\, \boldsymbol{s}_k)\|_2 \;<\; \|\boldsymbol{r}(\boldsymbol{x}_k)\|_2$.
3.) Neue Näherung $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k\, \boldsymbol{\Delta x}_k$.
4.) Abbruch, falls $\|\boldsymbol{\Delta x}_k\|$ hinreichend klein ist.

Bemerkungen (9.14)

a) Zur Bestimmung der Schrittlänge sind die Verfahren nach Armijo und Goldstein – oder Varianten hiervon – gebräuchlich. Wie beim Newton–Verfahren ist darauf zu achten, daß in der Konvergenzphase die Schrittlänge $\alpha_k = 1$ gewählt wird.

b) Das lineare Ausgleichsproblem (9.12) läßt sich auch als **Linearisierung** der ursprünglichen Zielfunktion $\|\boldsymbol{r}(\boldsymbol{x}_k)\|_2$ um die aktuelle Näherung $\boldsymbol{x}_k$ interpretieren (Linearisierung von $\boldsymbol{r}$ innerhalb der Norm).

c) Im Unterschied zum ursprünglichen Newton–Verfahrten (9.9) läßt sich für das Gauß–Newton–Verfahren (9.13) i.allg. keine quadratische Konvergenz erhalten (auch nicht bei $\alpha_k = 1$). Allerdings läßt sich ein zu Satz 6.7, S. 55 analoger lokaler Konvergenzsatz beweisen. Hierzu muß vorausgesetzt werden, daß die Jacobi-Matrix $\boldsymbol{r}'(\boldsymbol{x})$ vollen Rang (=n) besitzt. Ferner wird eine zusätzliche Bedingung benötigt, die im wesentlichen fordert, daß Daten und Ansatzfunktion „adäquat“ sind. Quadratische Konvergenz erhält man jedoch nur für sogenannte kompatible Modelle, d.h für $\boldsymbol{r}(\boldsymbol{x}^*) = \boldsymbol{0}$.

Beispiel (9.15) Wir betrachten nochmals das Beispiel (9.1). Als Ansatzfunktion wählen wir gemäß der früheren Betrachtungen

$$y(t; x_1, x_2, x_3) \;=\; x_3\;(\exp(-x_1\,t) - \exp(-x_2\,t))\,.$$

Die Meßdaten sind nach Blatt et al.(1975) in folgender Tabelle angegeben.

Tabelle 9.2. : Meßdaten zu Beispiel (9.15)

k	t_k	y_k
1	15.0	0.038
2	25.0	0.085
3	35.0	0.100
4	45.0	0.103
5	55.0	0.093
6	65.0	0.095
7	75.0	0.088
8	85.0	0.080
9	105.0	0.073
10	185.0	0.050
11	245.0	0.038
12	305.0	0.028
13	365.0	0.020

Ein Gauß–Newton–Verfahren liefert mit den Startdaten $x(1) = 0.01$, $x(2) = 0.024$, $x(3) = 0.32$ und der Genauigkeitsschranke $\mathrm{TOL} = 1.\mathrm{E}-7$ nach 18 Iterationen die optimalen Parameter:

$$\boldsymbol{x}^* \;\approx\; \begin{pmatrix} 0.57427\,75639\mathrm{E}-02 \\ 0.44406\,98224\mathrm{E}-01 \\ 0.14692\,52286\mathrm{E}+00 \end{pmatrix}$$

Für das Residuum ergibt sich $\|\boldsymbol{r}(\boldsymbol{x}^*)\| \approx 0.028179\,04182$. In der Abbildung 9.3 ist die resultierende Ausgleichsfunktion zusammen mit den Meßdaten dargestellt.

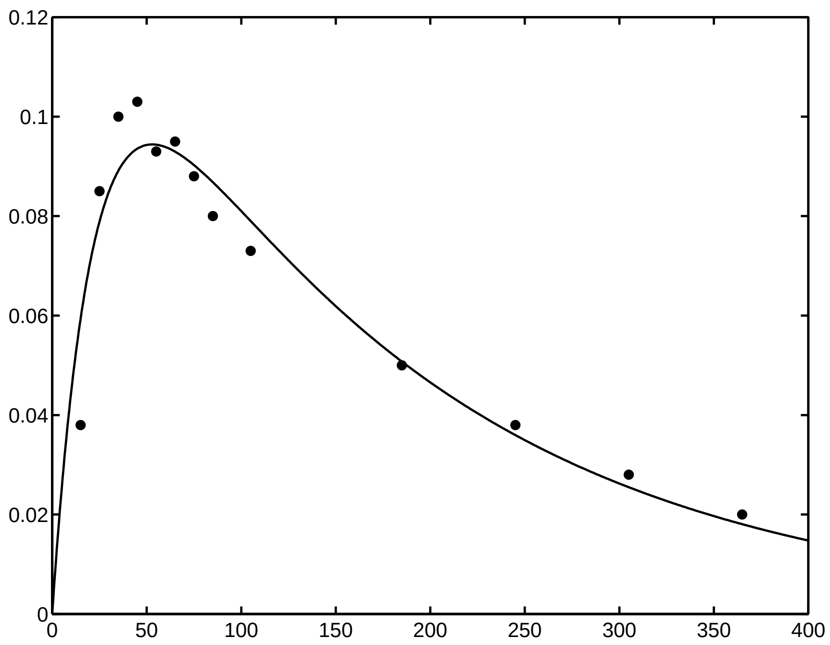

Abb. 9.3. Ausgleichsfunktion von Beispiel (9.14)

Wir beschreiben noch die Grundideen von zwei Modifikationen des Gauß–Newton–Verfahrens für nichtlineare Ausgleichsaufgaben, die Eingang in die aktuelle numerische Software gefunden haben, siehe z.B. die FORTRAN Programmbibliotheken IMSL und NAG und auch die in der Optimierungstoolbox von MATLAB eingebundenen Programme.

Das Verfahren von Levenberg und Marquardt

Wie schon erwähnt wurde, kann das lineare Hilfsproblem (9.12)

$$\text{Minimiere } \|\boldsymbol{r}'(\boldsymbol{x}^{(k)})\,\boldsymbol{\Delta x} + \boldsymbol{r}(\boldsymbol{x}^{(k)})\|_2, \;\; \boldsymbol{\Delta x} \in \mathbb{R}^n$$

als Linearisierung des Ausgangsproblems (innerhalb der Norm) interpretiert werden. Eine solche Linearisierung ist jedoch nur in der Nähe des Entwicklungspunktes $\boldsymbol{x}^{(k)}$ hinreichend genau. Es ist daher naheliegend, die obige Minimierung auf einen **Vertrauensbereich (trust region)** $\|\boldsymbol{\Delta x}\|_2 \leq \delta$ einzuschränken; vgl. auch den Abschnitt 8. Diese Idee führt auf ein gegenüber (9.11) modifiziertes lineares Gleichungssytem

$$\Big(\boldsymbol{r}'(\boldsymbol{x}^{(k)})^{\mathrm{T}}\,\boldsymbol{r}'(\boldsymbol{x}^{(k)}) + p_k\,\boldsymbol{I}_n\Big)\;\boldsymbol{\Delta x}^{(k)} \;=\; -\boldsymbol{r}'(\boldsymbol{x}^{(k)})^{\mathrm{T}}\,\boldsymbol{r}(\boldsymbol{x}^{(k)})\,. \tag{9.16}$$

Hierbei ist $p_k \geq 0$ ein Lagrange–Parameter, der so gewählt wird, daß die Koeffizientenmatrix in (9.16) positiv definit ist. Ein Nachteil des Verfahrens ist jedoch, daß in der

Konvergenzphase, d.h. für $p_k \approx 0$ mit (9.16) im wesentlichen die schlecht konditionierte Normalgleichung zu lösen ist.

Eine Realisierung des Levenberg–Marquard–Verfahrens nach Dennis und Schnabel findet man in der Programmbibliothek IMSL (Algorithmen UNLSJ/DUNLSJ) und in der Optimierungstoolbox von MATLAB. Das in MATLAB realisierte Programm **leastsq** schaltet selbständig bei schwierigeren Problemen vom Gauß–Newton–Verfahren auf das Verfahren nach Levenberg–Marquardt um.

Modifikation nach Gill und Murray

In der Variante des Gauß–Newton–Verfahrens nach Gill und Murray wird in jedem Iterationsschritt zunächst die Gauß–Newton–Korrektur $\boldsymbol{\Delta x}_{\mathrm{GN}}$ über das lineare Ausgleichsproblem (9.12) berechnet. Hierzu wird eine Singulärwertzerlegung der Jacobi–Matrix $\boldsymbol{r}'(\boldsymbol{x}^{(k)})$ berechnet

$$\boldsymbol{r}'(\boldsymbol{x}^{(k)}) \;=\; \boldsymbol{U} \begin{pmatrix} \boldsymbol{S} \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{v}^{\mathrm{T}}.$$

Dabei sind $\boldsymbol{U} \in \mathbb{R}^{(m,m)}$ und $\boldsymbol{v} \in \mathbb{R}^{(n,n)}$ orthogonale Matrizen, $\boldsymbol{S}$ ist eine Diagonalmatrix, $\boldsymbol{S} = \mathrm{diag}(s_1, \ldots, s_n)$ mit $s_1 \geq \ldots \geq s_n$ (Singulärwerte).

Liefert nun die Gauß–Newton–Richtung keinen ausreichenden Abstieg der Zielfunktion $\|\boldsymbol{r}(\boldsymbol{x})\|_2$, so wird anstelle von $\boldsymbol{\Delta x}_{\mathrm{GN}}$ die ursprüngliche Newton–Korrektur $\boldsymbol{\Delta x}_{\mathrm{N}}$ verwendet. Die Newton–Gleichung lautet nach (9.9), (9.10)

$$\Big(\boldsymbol{r}'(\boldsymbol{x}^{(k)})^{\mathrm{T}}\, \boldsymbol{r}'(\boldsymbol{x}^{(k)}) \;+\; \boldsymbol{B}_k\Big)\; \boldsymbol{\Delta x}_{\mathrm{N}}^{(k)} \;=\; -\boldsymbol{r}'(\boldsymbol{x}^{(k)})^{\mathrm{T}}\, \boldsymbol{r}(\boldsymbol{x}^{(k)})\,, \tag{9.17}$$

wobei $\boldsymbol{B}_k$ gegeben ist durch

$$\boldsymbol{B}_k \;=\; \sum_{i=1}^{m} r_i(\boldsymbol{x}^{(k)})\; \boldsymbol{\nabla}^2 r_i(\boldsymbol{x}^{(k)}).$$

Mit der Singulärwertzerlegung transformiert sich diese Gleichung zu

$$\begin{aligned} \Big(\boldsymbol{S}^2 \;+\; \boldsymbol{v}^{\mathrm{T}}\boldsymbol{B}_k\boldsymbol{v}\Big)\; \boldsymbol{\Delta z} \;&=\; -\,(\boldsymbol{S}\mid\boldsymbol{0})\, \boldsymbol{U}^{\mathrm{T}}\, \boldsymbol{r}(\boldsymbol{x}^k) \\ \boldsymbol{\Delta x}_{\mathrm{N}}^{(k)} \;&=\; \boldsymbol{v}\, \boldsymbol{\Delta z}\,. \end{aligned} \tag{9.18}$$

Durch eine geeignete Aufspaltung in „große“ und „kleine“ Singulärwerte läßt sich dieses Gleichungssystem numerisch stabil lösen.

Die Matrizen $\boldsymbol{B}_k$ sind aufwendig zu berechnen. In der Realisierung des Verfahrens nach Gill und Murray geschieht dies durch numerische Differentiation der vom Benutzer bereit zu stellenden Gradienten $\boldsymbol{\nabla} r_i$ bzw. durch update–Formeln (Quasi–Newton–Verfahren).

Realisierungen dieses Verfahrens findet man in der NAG Programmbibliothek (Algorithmen E04GCF und E04GEF).