

1 NICHTLINEARE AUSGLEICHSPROBLEME

In Numerik I haben wir uns bereits mit linearen Ausgleichsproblemen befasst. Wir erinnern uns daran, dass diese Probleme von der Form

$$\|Ax - d\|_2^2 \rightarrow \min \quad \text{mit } A \in \mathbb{R}^{m \times n}, d \in \mathbb{R}^m, m > n \quad (1.1)$$

sind. Es besteht also ein linearer Zusammenhang zwischen den „Messungen“ d und den Parametern x . Sie können mittels der Gaußschen Normalengleichung

$$A^T Ax = A^T d$$

bzw. numerisch mit der QR -Zerlegung gelöst werden. Solche Probleme entstehen beispielsweise bei einer linearen Regression durch mehrere Messpunkte. Betrachten wir nun anstelle einer linearen Regression folgendes Beispiel.

Beispiel 1.0.1 (Kreisinterpolation) Gegeben sei die Punktwolke $(s_i, t_i), i = 1, \dots, m$, mit dem Hinweis, dass alle diese Punkte näherungsweise auf einem Kreis liegen. Geht man von einem Kreis mit Mittelpunkt $M = (\mu_s, \mu_t)$ und Radius $r > 0$ aus, so müssten die Punkte

$$\sqrt{(s_i - \mu_s)^2 + (t_i - \mu_t)^2} \approx r \quad i = 1, \dots, m$$

erfüllen. Die Aufgabe besteht also darin, den Mittelpunkt M und den Radius r so zu bestimmen, dass die Summe über die Quadrate der einzelnen Abstände

$$e_i = \sqrt{(s_i - \mu_s)^2 + (t_i - \mu_t)^2} - r \quad i = 1, \dots, m$$

der Punkte (s_i, t_i) vom Kreis mit Mittelpunkt M und Radius r minimiert wird.

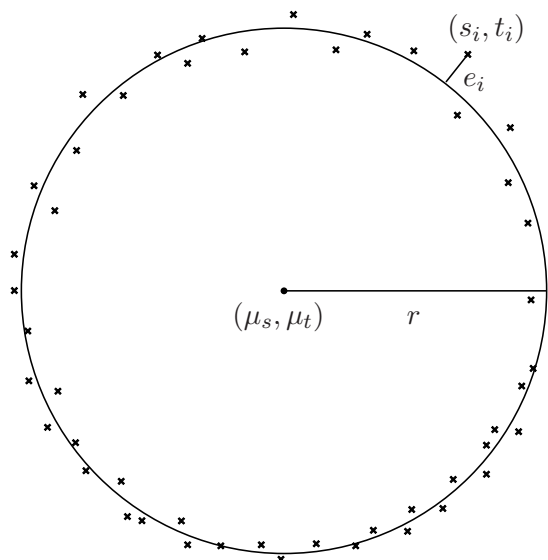


Abb. 1.1: Kreisinterpolation

Bezeichnet $e \in \mathbb{R}^m$ den Vektor (e_1, \dots, e_m) , so lässt sich das Problem wie folgt formulieren:

$$\|e\|_2^2 \rightarrow \min \Leftrightarrow g(\mu_s, \mu_t, r) := \sum_{i=1}^m \left(\sqrt{(s_i - \mu_s)^2 + (t_i - \mu_t)^2} - r \right)^2 \rightarrow \min.$$

Es besteht also in diesem Fall kein linearer Zusammenhang zwischen den Parametern (μ_s, μ_t, r) und den Punkten (s_i, t_i) .

Beispiel 1.0.2 (Wo war ich?) Haben Sie sich beim Anschauen Ihrer Urlaubsfotos¹ nicht schon einmal gefragt: Wo war ich da eigentlich? Wenn Sie sich nicht einmal daran erinnern, in welchem Land Sie waren, kann Ihnen die Mathematik auch nicht weiterhelfen. Aber wie sieht es aus, wenn Sie einige fotografierte Punkte so identifizieren können, dass Sie sie auf einer Landkarte wiederfinden? Die Problemstellung lautet nun: Man bestimme die Position des Fotografen auf der Karte und die Höhe über dem Meeresspiegel, die Brennweite und die Ausrichtung der Kamera zu den folgenden Daten.

Nr.	Berg	Höhe	Position in Metern in der Karte	
			x	y
1	Alpspitze	2627,6 m	3700	4125
2	Äußere Hölltalspitze	2720,0 m	2200	3175
3	Vorderer Waxenstein	2136,0 m	2350	6000
4	Mittagsscharte	2072,0 m	2175	5885
5	Großer Waxenstein	2276,5 m	1850	5950

Tab. 1.1: Angabe zur Lage der Gipfel

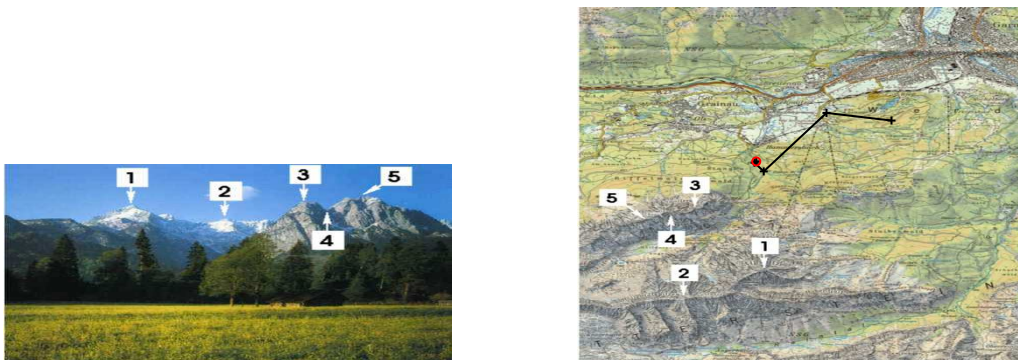


Abb. 1.2: Fotografie und Karte mit gekennzeichneten Gipfeln

Mathematisch formuliert führt dieses Problem auf ein nichtlineares Ausgleichsproblem: Der Vektor $w_i \in \mathbb{R}^3$ von der Linse im Objektiv zum Punkt u_i des Gipfels i auf dem Foto muss für alle $i = 1, \dots, 5$ mit dem Vektor $x - x_i$ der Differenz aus der Position des Fotografen $x \in \mathbb{R}^3$ und der Position des Berges $x_i \in \mathbb{R}^3$ auf der Karte korrespondieren.

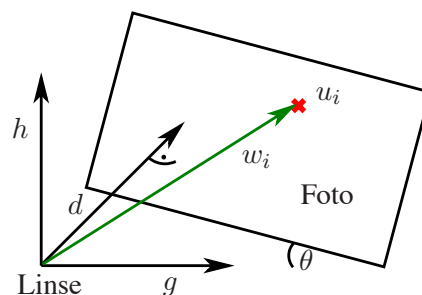


Abb. 1.3: Orthogonalsystem d, g, h , Drehwinkel θ und Vektor w_i von der Linse zum Gipfel i

¹Die Idee dieser Aufgabenstellung geht auf Ernst Hairer, Gerhard Wanner und Stephanie Cirilli von der Universität Genf zurück.

D.h. man möchte

$$\sum_{i=1}^5 \|w_i \times (x - x_i)\|_2^2 \rightarrow \min,$$

wobei \times das Kreuzprodukt bezeichnet.

Denkbar wären auch folgende Beispiele aus der Stochastik bzw. der Physik.

Beispiel 1.0.3 (Dichte der Gamma-Verteilung) Aus der Vorlesung Stochastik I kennen wir die Gamma-Verteilung $\Gamma(\lambda, p)$, $\lambda > 0, p > 0$, mit der Dichte

$$f(\lambda, p; t) = \begin{cases} \frac{\lambda^p t^{p-1}}{\Gamma(p)} e^{-\lambda t} & , t \geq 0 \\ 0 & , t < 0, \end{cases}$$

wobei $\Gamma(p)$ die Gamma-Funktion

$$\Gamma(p) := \int_0^{\infty} x^{p-1} e^{-x} dx, \quad p > 0$$

bezeichnet. Zu $t_i \geq 0, i = 1, \dots, m$, seien die Werte f_i gegeben und man vermutet, dass $f_i \approx f(\lambda, p; t_i)$ gilt mit geeigneten $\lambda, p > 0$. Man sucht daher

$$\arg \min_{\lambda, p > 0} \|F(\lambda, p)\|_2^2 \quad \text{mit} \quad F(\lambda, p) := (f(\lambda, p; t_i) - f_i)_{i=1, \dots, m}.$$

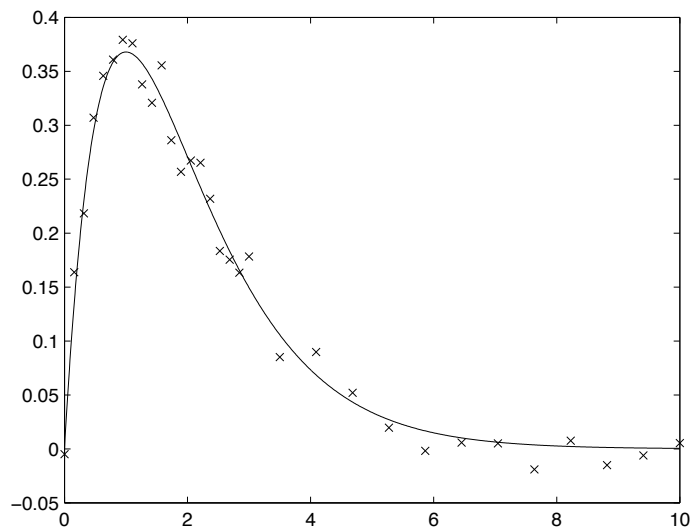


Abb. 1.4: Dichte der Gamma-Verteilung für $\lambda = 1, p = 2$ und Punktwolke

Beispiel 1.0.4 (Gedämpfte Schwingung) Die Auslenkung einer gedämpften Schwingung in Abhängigkeit von der Zeit lässt sich mit der Gleichung

$$u(t) = u_0 e^{-\delta t} \sin(\omega t + \varphi_0)$$

beschreiben, wobei u_0 ein Anfangswert, φ_0 die Anfangsphase der Schwingung, δ die Abklingkonstante und ω die Frequenz sind. Diese Parameter kann man aus der Masse m , der Federkonstanten

D sowie der Dämpfungskonstanten b unter Berücksichtigung der Gültigkeit der Differenzialgleichung

$$\ddot{u} + \frac{b}{m} \dot{u} + \frac{D}{m} u = 0$$

bestimmen. Liegen diese Werte nicht vor, so ist man auf eine näherungsweise Bestimmung von u_0, δ, ω und φ_0 aus Messwerten $(t_i, u_i), i = 1, \dots, m$, angewiesen.

Man betrachtet dazu die Fehlerfunktion

$$F : \mathbb{R}^4 \rightarrow \mathbb{R}^m, \quad (u_0, \delta, \omega, \varphi_0) \mapsto \left(u_0 e^{-\delta t_i} \sin(\omega t_i + \varphi_0) - u_i \right)_{i=1, \dots, m}$$

und möchte diese in der euklidischen Norm minimieren; man sucht also

$$\arg \min_{(u_0, \delta, \omega, \varphi_0) \in \mathbb{R}^4} \|F(u_0, \delta, \omega, \varphi_0)\|_2^2.$$

Alle diese Beispiele führen uns zu der Aufgabe, einen Parametervektor $x \in \mathbb{R}^n$ zu bestimmen, sodass

$$\|F(x)\|_2^2 \rightarrow \min \quad \Leftrightarrow \quad g(x) := \frac{1}{2} F(x)^T F(x) \rightarrow \min$$

mit einer (möglicherweise) nichtlinearen Funktion $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$.

1.1 GAUSS-NEWTON-VERFAHREN

Wir setzen nun voraus, dass $F : O \rightarrow \mathbb{R}^m$ zweimal stetig differenzierbar auf einer offenen Menge $O \subset \mathbb{R}^n$ ist und $m > n$ gilt, wir also nur den überbestimmten Fall betrachten. Unser Interesse gilt lokalen inneren Minima $x^* \in O$ von g , die den hinreichenden Bedingungen

$$\nabla g(x^*) = 0 \quad \text{und} \quad g''(x^*) \quad \text{positiv definit} \quad (1.2)$$

genügen. Bezeichnet $F'(x) \in \mathbb{R}^{m \times n}$ die Jacobi-Matrix von F an der Stelle x sowie $F_i''(x)$ die Hesse-Matrix von F_i an der Stelle x , d.h. $F_i''(x) := \left(\frac{\partial^2 F_i(x)}{\partial x_j \partial x_k} \right)_{1 \leq j, k \leq n} \in \mathbb{R}^{n \times n}$, dann gilt

$$\nabla g(x) = F'(x)^T F(x), \quad g''(x) = F'(x)^T F'(x) + \sum_{i=1}^m F_i(x) F_i''(x).$$

Um $g(x)$ zu minimieren, betrachten wir also das nichtlineare Gleichungssystem

$$\nabla g(x) = F'(x)^T F(x) \stackrel{!}{=} 0.$$

Wendet man hierauf das Newton-Verfahren an, so lautet die Iterationsvorschrift

$$\begin{aligned} g''(x_k) s_k &= -\nabla g(x_k) \\ x_{k+1} &= x_k + s_k \\ &= x_k - [g''(x_k)]^{-1} \nabla g(x_k) \\ &= x_k - \left[F'(x_k)^T F'(x_k) + \sum_{i=1}^m F_i(x_k) F_i''(x_k) \right]^{-1} F'(x_k)^T F(x_k) \end{aligned} \quad (1.3)$$

Unter Annahme (1.2) ist g'' in einer Umgebung von x^* positiv definit und daher invertierbar. Geht man von zum Modell kompatiblen Daten aus, d.h. gilt $F(x^*) = 0$, so ist

$$g''(x^*) = F'(x^*)^T F'(x^*).$$

Die Bedingung der positiven Definitheit der Hesse-Matrix $g''(x^*)$ fällt in diesem Fall zusammen mit der Bedingung, dass die Jacobi-Matrix $F'(x^*)$ vollen Rang n hat. Für (nahezu) kompatible Daten und für ein x aus einer Umgebung $U \subset O$ von x^* kann man also die Hesse-Matrix $g''(x)$ durch $F'(x)^T F'(x)$ approximieren (*Quasi-Newton-Verfahren*) und sich so die Berechnung von $F''(x)$ in (1.3) sparen.

Mit dieser Vorgehensweise erhalten wir den folgenden Algorithmus.

Algorithmus 1.1.1: Gauß-Newton-Verfahren

Input: x_0, F, F'

for $k = 0, 1, \dots$

i) Berechne $F(x_k), F'(x_k)$.

ii) Bestimme den Korrekturvektor s_k gemäß

$$F'(x_k)^T F'(x_k) s_k = -F'(x_k)^T F(x_k). \quad (1.4)$$

iii) Setze $x_{k+1} = x_k + s_k$.

end

Bemerkungen 1.1.1 i) Offensichtlich ist (1.4) die Normalengleichung zum Ausgangsproblem

$$\|F'(x_k) s_k + F(x_k)\|_2 \rightarrow \min. \quad (1.5)$$

Somit wurde die numerische Lösung eines nichtlinearen Gaußschen Ausgleichsproblems auf die numerische Lösung einer Folge von linearen Ausgleichsproblemen zurückgeführt.

ii) Man hätte den Algorithmus 1.1.1 auch direkt – ähnlich wie beim Newton-Verfahren – durch Taylorentwicklung und Abbrechen nach dem linearen Term herleiten können. Daher heißt dieses Vorgehen auch Gauß-Newton-Verfahren.

iii) Wegen der Vernachlässigung des Termes aus (1.3), der die zweiten Ableitungen von F enthält, geht die quadratische Konvergenz des Newton-Verfahrens verloren und das Gauß-Newton-Verfahren ist in der Regel nur linear konvergent.

Eine einfache mögliche Realisierung des Gauß-Newton-Verfahrens in Matlab ist nachfolgend dargestellt.

MATLAB-Funktion: gauss_newton.m

```

1  function x = gauss_newton(F,DF,x0,maxit,tol)
2      k=0;
3      x=x0;
4      s=-(DF(x0)'*DF(x0))\ (DF(x0)'*F(x0));
5      while norm(s)>tol && k<maxit
6          k=k+1;
7          x=x+s;
8          s=-(DF(x)'*DF(x))\ (DF(x)'*F(x));
9      end
10 end

```

1.2 KONVERGENZ DES GAUSS-NEWTON-VERFAHRENS

Zur Analyse des Gauß-Newton-Verfahrens sei nun x^* ein kritischer Punkt von g , d.h. $\nabla g(x^*) = 0$, der in einer Umgebung U eindeutig ist. Außerdem gelte

$$\text{Rang}(F'(x)) = n \quad \forall x \in U.$$

Dann hat die Normalengleichung (1.4) für $x_k \in U$ genau eine Lösung

$$s_k = -[F'(x_k)^T F'(x_k)]^{-1} F'(x_k)^T F(x_k)$$

und die Gauß-Newton-Iteration kann folgendermaßen formuliert werden:

$$\begin{aligned} x_{k+1} &= x_k - [F'(x_k)^T F'(x_k)]^{-1} F'(x_k)^T F(x_k) \\ &= \Phi(x_k) \end{aligned}$$

mit

$$\Phi(x) := x - [F'(x)^T F'(x)]^{-1} F'(x)^T F(x). \quad (1.6)$$

Das Gauß-Newton-Verfahren ist demzufolge eine Fixpunktiteration mit Iterationsfunktion Φ . Hinreichend für die lokale Konvergenz der Fixpunktiteration ist die Beschränktheit der Norm der Jacobi-Matrix im Fixpunkt

$$\|\Phi'(x^*)\| < 1$$

für eine beliebige Operatornorm $\|\cdot\|$. Da $\text{Rang}(F'(x^*)) = n$ gilt, ist die Matrix $F'(x^*)^T F'(x^*)$ positiv definit. Daher existiert eine (eindeutige) symmetrische, positiv definite Matrix $A \in \mathbb{R}^{n \times n}$, sodass

$$A^2 = F'(x^*)^T F'(x^*).$$

Wir definieren nun für die weitere Konvergenzuntersuchung die Matrix $K \in \mathbb{R}^{n \times n}$ durch

$$K := -A^{-1} \left(\sum_{i=1}^m \frac{F_i(x^*)}{\|F(x^*)\|_2} F_i''(x^*) \right) A^{-1}.$$

Da K ebenfalls eine symmetrische Matrix ist, sind alle Eigenwerte von K reell. Mit den Matrizen A und K kann man nun die Hesse-Matrix $g''(x^*)$ und die Jacobi-Matrix $\Phi'(x^*)$ folgendermaßen darstellen.

Lemma 1.2.1 *Es gilt*

$$g''(x^*) = A(I - \|F(x^*)\|_2 K)A, \quad (1.7)$$

$$\Phi'(x^*) = \|F(x^*)\|_2 A^{-1} K A. \quad (1.8)$$

Wenn x^* ein lokales Maximum oder ein Sattelpunkt von g ist, muss

$$\varrho(K) \|F(x^*)\|_2 \geq 1$$

gelten, wobei $\varrho(K)$ den Spektralradius von K , also den Betrag des betragsgrößten Eigenwertes von K , bezeichnet.

Beweis. Die Gleichungen (1.7) und (1.8) erhält man schnell durch Nachrechnen.

Damit $g''(x^*)$ nicht positiv definit ist, muss die Matrix $I - \|F(x^*)\|_2 K$ nichtpositive Eigenwerte besitzen, d.h. das Spektrum von $\|F(x^*)\|_2 K$ muss Werte ≥ 1 enthalten. Daher muss $\varrho(K) \|F(x^*)\|_2 \geq 1$ gelten. \square

Als ein weiteres Hilfsmittel definieren wir die Vektornorm $\|x\|_A := \|Ax\|_2$ sowie für $B \in \mathbb{R}^{n \times n}$ die zugehörige Matrixnorm

$$\|B\|_A = \max_{\|x\|_A=1} \|Bx\|_A = \|ABA^{-1}\|_2.$$

Satz 1.2.2 Für die Gauß-Newton-Iterationsfunktion Φ aus (1.6) gilt

$$\begin{aligned} \|\Phi'(x^*)\|_A &= \varrho(K)\|F(x^*)\|_2, \\ \|\Phi'(x^*)\| &\geq \varrho(K)\|F(x^*)\|_2 \quad \text{für jede Operatornorm } \|\cdot\|. \end{aligned}$$

Beweis. Aus Lemma 1.2.1 erhält man

$$\|\Phi'(x^*)\|_A = \|A\Phi'(x^*)A^{-1}\|_2 = \|F(x^*)\|_2\|K\|_2 = \|F(x^*)\|_2\varrho(K).$$

Für jede Operatornorm $\|\cdot\|$ und jede Matrix $B \in \mathbb{R}^{n \times n}$ gilt $\|B\| \geq \varrho(B)$. Damit ergibt sich


$$\|\Phi'(x^*)\| \geq \varrho(\Phi'(x^*)) = \|F(x^*)\|_2\varrho(A^{-1}KA) = \|F(x^*)\|_2\varrho(K).$$

□

Mit Satz 1.2.2 können wir nun folgende Aussagen über das Verhalten der Gauß-Newton-Iteration treffen.

Korollar 1.2.3 i) Im Normalfall ist $F(x^*) \neq 0$, $K \neq 0$ und deshalb $\Phi'(x^*) \neq 0$.

Falls das Gauß-Newton-Verfahren konvergiert, ist die Konvergenz im Allgemeinen nicht schneller als linear.

Dies steht im Gegensatz zum Newton-Verfahren (vgl. Numerik II), das in der Regel quadratisch konvergent ist. 

ii) Wenn der kritische Punkt x^* ein lokales Maximum oder ein Sattelpunkt ist, gilt $\varrho(K)\|F(x^*)\|_2 \geq 1$ und $\|\Phi'(x^*)\| \geq 1$ für jede Operatornorm $\|\cdot\|$.

Lokale Maxima und Sattelpunkte sind für das Gauß-Newton-Verfahren also abstoßend, was günstig ist, da ein lokales Minimum gesucht wird.

iii) Die Größe $\varrho(K)\|F(x^*)\|_2$ ist entscheidend für die lokale Konvergenz des Gauß-Newton-Verfahrens.

Für ein lokales Minimum x^* der Funktion g ist die lokale Konvergenz des Gauß-Newton-Verfahrens gesichert, falls das Residuum $\|F(x^*)\|_2$ und die Größe $\varrho(K)$ hinreichend klein sind, sodass die Bedingung $\varrho(K)\|F(x^*)\|_2 < 1$ erfüllt ist.

iv) Ist x^* allerdings ein lokales Minimum von g , für das $\varrho(K)\|F(x^*)\|_2 > 1$ gilt, so ist $\|\Phi'(x^*)\| > 1$ für jede Operatornorm $\|\cdot\|$.

Ein lokales Minimum von g kann für das Gauß-Newton-Verfahren abstoßend sein.

1.3 LEVENBERG-MARQUARDT-VERFAHREN

Das Problem des Gauß-Newton-Verfahrens ist, dass die Linearisierung nur für „kleine“ Schritte s_k zulässig ist, sofern die Iterierte x_k noch weit vom Minimum entfernt ist. Die folgende Methode versucht durch eine Steuerung der Länge des Korrekturvektors s_k eine Verbesserung zu erzielen.

Im Levenberg-Marquardt-Verfahren wird das Ausgleichsproblem (1.5) zur Bestimmung des Korrekturvektors s_k durch ein anderes leicht abgeändertes Minimierungsproblem

$$\|F'(x_k) s_k + F(x_k)\|_2^2 + \mu^2 \|s_k\|_2^2 \rightarrow \min \quad (1.9)$$

ersetzt, wobei $\mu > 0$ ein zu wählender Parameter ist. Als neue Annäherung wird dann wiederum

$$x_{k+1} = x_k + s_k$$

gesetzt. Aus der Gleichung

$$\left\| \begin{pmatrix} F'(x_k) \\ \mu I \end{pmatrix} s_k + \begin{pmatrix} F(x_k) \\ 0 \end{pmatrix} \right\|_2^2 = \|F'(x_k) s_k + F(x_k)\|_2^2 + \mu^2 \|s_k\|_2^2$$

folgt, dass die Minimierungsaufgabe (1.9) die äquivalente Formulierung

$$\left\| \begin{pmatrix} F'(x_k) \\ \mu I \end{pmatrix} s_k + \begin{pmatrix} F(x_k) \\ 0 \end{pmatrix} \right\|_2 \rightarrow \min \quad (1.10)$$

besitzt. Im Vergleich zum Ausgleichsproblem (1.5) im Gauß-Newton-Verfahren besitzt (1.10) immer eine eindeutige Lösung s_k , da die Matrix $\begin{pmatrix} F'(x_k) \\ \mu I \end{pmatrix}$ vollen Rang hat.

Für den Korrekturvektor $s_k \neq 0$ gilt

$$\begin{aligned} \mu^2 \|s_k\|_2^2 &\leq \|F'(x_k) s_k + F(x_k)\|_2^2 + \mu^2 \|s_k\|_2^2 \\ &= \min_{s \in \mathbb{R}^n} \{ \|F'(x_k) s + F(x_k)\|_2^2 + \mu^2 \|s\|_2^2 \} \leq \|F(x_k)\|_2^2 \end{aligned}$$

und daher

$$\|s_k\|_2 \leq \frac{\|F(x_k)\|_2}{\mu}.$$

Der Parameter $\mu > 0$ kann folglich eine Dämpfung der Korrektur s_k bewirken und durch eine geeignete Wahl von μ kann man eine zu große Korrektur vermeiden. Man kann zeigen, dass unter bestimmten Voraussetzungen an F das Levenberg-Marquardt-Verfahren für „hinreichend großes“ μ konvergiert. Um Konvergenz zu gewährleisten darf man μ also nicht zu klein wählen; auf der anderen Seite führt ein großes μ aber nur zu einer kleinen Korrektur und somit erhält man nur sehr langsame Konvergenz.

Im Folgenden wird ein mögliches Verfahren zur Bestimmung des Parameters μ vorgestellt, in dem in jedem Schritt überprüft wird, ob μ zu klein oder zu groß gewählt ist, und μ gegebenenfalls angepasst wird.

Sei dazu $x_k \in \mathbb{R}^n$, $s_k = s_k(\mu)$ die Korrektur aus (1.9) sowie

$$\varepsilon_\mu := \frac{\|F(x_k)\|_2^2 - \|F(x_k + s_k)\|_2^2}{\|F(x_k)\|_2^2 - \|F(x_k) + F'(x_k) s_k\|_2^2} =: \frac{\Delta R(x_k, s_k)}{\Delta \tilde{R}(x_k, s_k)},$$

wobei angenommen wird, dass $\Delta \tilde{R}(x_k, s_k) \neq 0$ gilt. ε_μ beschreibt die Änderung des tatsächlichen Residuums $\Delta R(x_k, s_k)$ in Relation zu Änderung des Residuums im linearen Modell (Gauß-Newton-Modell) $\Delta \tilde{R}(x_k, s_k)$. Aus (1.9) folgt, dass $\Delta \tilde{R}(x_k, s_k) \geq 0$.

Für eine akzeptable Korrektur muss außerdem $\Delta R(x_k, s_k) > 0$ gelten, also $\varepsilon_\mu > 0$. Aufgrund der Taylorentwicklung konvergiert $\varepsilon_\mu \rightarrow 1$ für $\mu \rightarrow \infty$. Gibt man sich nun β_0, β_1 mit $0 < \beta_0 < \beta_1 < 1$ vor (z.B. $\beta_0 = 0.3, \beta_1 = 0.9$), so kann man zur Parametersteuerung folgendes Kriterium verwenden:

- $\varepsilon_\mu \leq \beta_0$: s_k wird nicht akzeptiert; μ wird vergrößert, z.B. $\mu \mapsto 2\mu$, und die neue zugehörige Korrektur s_k wird berechnet (*Gewährleistung der Konvergenz*).
- $\beta_0 < \varepsilon_\mu < \beta_1$: s_k wird akzeptiert; bei der Berechnung von s_{k+1} wird als Anfangswert dasselbe μ gewählt.
- $\varepsilon_\mu \geq \beta_1$: s_k wird akzeptiert; bei der Berechnung von s_{k+1} wird als Anfangswert ein kleineres μ , z.B. $\frac{\mu}{2}$ genommen (*Effektivität*).

Insgesamt erhalten wir für das Levenberg-Marquardt-Verfahren folgenden Algorithmus.

Algorithmus 1.3.1: Levenberg-Marquardt-Verfahren

Input: $x_0, F, F', \mu, \beta_0, \beta_1$

for $k = 0, 1, \dots$

i) Berechne $F(x_k), F'(x_k)$.

ii) Bestimme den Korrekturvektor s_k gemäß

$$[F'(x_k)^T F'(x_k) + \mu^2 I] s_k = -F'(x_k)^T F(x_k).$$

iii) Teste, ob die Korrektur s_k akzeptabel ist:

- $\varepsilon_\mu \leq \beta_0$: Setze $\mu = 2\mu$ und berechne s_k gemäß ii) neu.
- $\varepsilon_\mu \geq \beta_1$: Setze $\mu = \frac{\mu}{2}$ und behalte s_k .

iv) Setze $x_{k+1} = x_k + s_k$.

end

In Matlab sieht eine Realisierung dieses Verfahrens folgendermaßen aus.

MATLAB-Funktion: levenberg_marquardt.m

```

1  function x = levenberg_marquardt(F,DF,x0,mu0,beta0,beta1,maxit,tol)
2      n=length(x0);
3      k=0;
4      mu=mu0;
5      x=x0;
6      s=-(DF(x0)'*DF(x0)+mu^2*eye(n))\ (DF(x0)'*F(x0));
7      while norm(s)>tol && k<maxit
8          [s,mu] = korrektur(F,DF,x,mu,beta0,beta1);
9          x=x+s;
10         k=k+1;
11     end
12 end

```

```
13
14 function [s,mu] = korrektur(F,DF,x,mu,beta0,beta1)
15     n=length(x);
16     s=-(DF(x)'*DF(x)+mu^2*eye(n))\ (DF(x)'*F(x));
17     eps_mu=(F(x)'*F(x)-F(x+s)'*F(x+s))/(F(x)'*F(x)-(F(x)+DF(x)*s)
18         '* (F(x)+DF(x)*s));
19     if eps_mu <= beta0
20         [s,mu]=korrektur(F,DF,x,2*mu,beta0,beta1);
21     elseif eps_mu >= beta1
22         mu=mu/2;
23     end
24 end
```