Vorlesung

Mathematische Methoden der Geophysik und Meteorologie 1
Numerische Mathematik

Sommersemester 2011
Priv.-Doz. Dr. H. Elbern
mit
M. Sc. Math. Ketevan Kasradze
M. Sc. (Technomath.) Patricia Schmid

File: mathmet_I_v3.tex
Version: 3.1
Date: SS 2012, 27. Juni 2012

Inhaltsverzeichnis

Lehrbücher

Lehrbücher zur Orientierung bei der Lösung praktischer numerischer Probleme:

Köckler, N. , Numerische Algorithmen in Softwaresystemen, Teubner, 1990.

Meis, Th., und U. Marcowitz, Numerische Behandlung partieller Differentialgleichungen, Springer, 1981.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipies, Cambridge University Press

Shewchuk, J.S., An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, School of Computer Science, Carnegie Mellon University, 1994. (Verfgbar auf web)

J. Stoer, Einführung in die Numerische Mathematik I, Springer Verlag, 1983 (ggf. neuere Auflage)

J. Stoer, R. Bulirsch, Einführung in die Numerische Mathematik II, Springer Verlag, 1978 (ggf. neuere Auflage)

Grundlagen aus der Analysis findet man in

Courant, Hilbert, Methoden der mathematischen Physik I und II, Springer Verlag.

Kapitel 1

Einleitung

Das Modul “Mathematische Methoden der Geophysik und Meteorologie 1” vermittelt die für diese Studiengänge elementaren Grundlagen aus der klassischen numerischen Mathematik. Zusammen mit den Übungen sollen grundlegende Fertigkeiten erworben werden, numerische Verfahren

- unter Gesichtspunkten wie Effizienz, Komplexität, Robustheit, Stabilität und Konvergenz für eigene Anwendung zu bewerten, und

- diese nach den gleichen Gesichtspunkten für eigene Softwareentwicklungen und -nutzungen zu entwickeln und anzuwenden.

Die Stofffülle kann nur einen Überblick erlauben und für die spätere wissenschaftliche Praxis in der Geophysik und Meteorologie eine Einstiegshilfe sein, da in der Regel hier effizientere Spezialversionen der behandelten Algorithmen oder gar andere, komplexere Verfahren verwendet werden. Gleichwohl wird das Ziel angestrebt, nach erfolgreichem Abschluss dieses Moduls eine begründete Auswahl von numerischen Algorithmen treffen und bewerten zu können. Bei der Auswahl der Themen und numerischen Verfahren werden jene bevorzugt, die in der Geophysik und Meteorologie, anders als etwa in der allgemeinen Numerik von besonderer Bedeutung sind. Hier sind insbesondere Verfahren zu nennen, die bei Inversionsproblemen auftreten (etwa Minimierung, Optimierung und Integralgleichungen), wie sie in der Geophysik und Meteorologie bei Fernerkundungsverfahren besonders häufig vorkommen. Ferner gilt ein Augenmerk den Grundlagen zeitlicher Integration prognostischer Modelle, mit ihrer Stabilitätsproblematik besondere Aufmerksamkeit erfordern.

Das Modul ist ein nicht kompensierbares Pflichtmodul. Es ist bestanden, wenn

1. erfolgreich und regelmäßig an den Vorlesungen und Übungen teilgenommen wurde (es müssen mindestens 50% der in den Übungen zu erreichenden Punkte erworben worden sein), und

2. die Abschlussklausur bestanden wurde.

Bei nicht bestandener Abschlussklausur wird zum Ende der Sommersemesterferien oder zu Beginn des Wintersemesters die Gelegenheit einer Wiederholungsprüfung (Klausur oder mündliche Prüfung) gegeben. Bei nicht bestandener Wiederholungsprüfung wird die Wiederholung der Lehrveranstaltung des Moduls empfohlen. Wird die zweite Wiederholungsprüfung nicht bestanden, ist das Modul entgültig nicht bestanden. Die Modulnote ist die Note der Abschlussklausur (bzw. der Wiederholungsprüfung).

Teilnahmevoraussetzung für diese Vorlesung ist, dass die Module

- Mathematik für Physiker I
- Mathematik für Physiker II
- Datenverarbeitung und Programmieren

bestanden sind. Kenntnisse und einfache Programmierfertigkeiten in FORTRAN90 werden vorausgesetzt und sollen weiter entwickelt werden. Darüber hinaus werden MATLAB und vereinzelt auch Computeralgebra nur zur Unterstützung angewandt werden.

Die Verwendbarkeit des im Modul Erlernten ist allen exakten naturwissenschaftlichen Fächer gegeben. Die Anrechnung in Endnote wird gewichtet mit dem Faktor von 6/180 vorgenommen.

MATHMET II behandet die Datenanalyse. Soweit geeignet, wird das Modul MATHMET I vorzugsweise und vorbereitend numerische Verfahren behandeln, die im zweiten Teil genutzt werden können.

Frage:
Nach welchen allgemeinen Gesichtpunkten werden numerische Algorithmen bewertet?

Kapitel 2

Rundungsfehler

2.1 Problemstellung und Motivation

Die numerische Lösung (geo)physikalischer, wie auch anderer, nicht völlig trivialer Probleme, beispielsweise a) die Ausbreitung von Wellen im Erdinneren, oder b) die Berechnung atmosphärischer Strömungen zur Wettervorhersage, wird in mehreren Schritten vollzogen. Dabei ist in gewisser Weise jeder dieser Phasen problembehaftet. Man kann z.B. die folgenden Schritte unterscheiden:

1. Identifikation des geophysikalischen Modells, in der Regel ausgedrückt duch eine Differential- oder Integralgleichung. (Fehlerquelle: Unvollständige Modellbeschreibung).

   Beispiel zu a) Maxwellgleichungen (Fehlerquelle z. B. vernachlässigte Modellierung der Verschiebungsströme)
   Beispiel zu b) Navier-Stokes-Theorie und Thermodynamik (Fehlerquelle vernachlässigte Modellierung der Phasenübergänge beim Wasser in der Atmosphäre).

2. Formulierung eines mathematisches Modells (Fehlerquelle: unvollständige Formulierung der Differentialgleichungen oder unzulängliche Parametrisierungen.)

3. **Auswahl von numerisches Verfahren** zur Lösung, beispielsweise finite Differenzen (Fehlerquelle: falsche oder ineffiziente Methodenauswahl, instabile Algorithmen, Abbruchfehler durch Beendigung einer Iteration, unzureichende räumliche und zeitliche Auflösung/Abschneidewerte)

4. **Auswahl von Algorithmen** zur Organisation des Rechenablaufes (Fehlerquelle: Rechenungenauigkeiten)

5. **Erstellung von Software** (Fehlerquelle: Programmierfehler (bugs) oder Effizienzfehler (performance bugs))

6. Beschaffung von Inputdaten und Rechnung (Fehlerquelle: ungenaue Inputdaten)

Diese Vorlesung soll in den Stand versetzen, zu den Punkten 3, 4 und 5 die angemessenen Bewertungen vorzunehmen und die geeigneten Verfahren und Algorithmen auszuwählen, zu entwickeln, zu installieren und zu betreiben. Gegenstand dieses Kapitels ist zunächst das Kennenlernen der Grenzen der Rechengenauigkeit bei gegebenen Rechenablauf, also Punkt 4. Gleichwohl werden im Abschnitt 2.5 (Fehlerfortpflanzung) Ausdrücke vorgestellt, die in analoger Form auch für andere Fehlerquellen, wie insbesondere auch der Fehlerfortpflanzung bei zeitlicher Integration gelten.

2.2 Beispiele zur Problemstellung

Die Dringlichkeit der Problematik der Rechengenauigkeit kann an folgenden Beispielen erfasst werden, auch wenn sie nicht im engsten Sinne der Rundungsfehlerproblematik entspringen (entnommen aus *http://mathworld.wolfram.com/RoundoffError.html*):

Beispiel 2.1 *Ein bekanntes Beispiel ist der Fehlschlag der ersten Ariane 5 am 4. Juni 1996 (European Space Agency 1996), infolge fehlerhafter Software, die von der Ariane 4 adaptiert wurde. In der 37. Flugsekunde versuchte das Trägheitsnavigationssystem eine 64 Bit Gleitkommazahl in eine 16-Bit-Zahl zu verwandeln, aber verursachte ein Registerüberlauffehler. Dieser wurde von Steuersystem als Zahl interpretiert, wodurch die Rakete vom Kurs fortgelenkt wurde und gesprengt werden musste.*

Beispiel 2.2 *Das amerikanische Patriot-Flugabwehrsystem erwies sich im 2. Golfkrieg infolge eines Rundungsfehlers als ineffizient (Skeel 1992, U.S. GAO 1992). Das System nutzte ein ganzzahliges Zeitregister, welches in Schritten von 0.1 Sekunden hochgezählt wurde. Allerdings wurden die natürlichen Zahlen durch Multiplikation mit der binären Näherung von 0.1 in Dezimalzahlen ungewandelt,*

$$0.00011001100110011001100_2 = \frac{209715}{2097152}\,.$$

Nach 100 Stunden Einsatzzeit (3.6×10^6 ticks) wurde als Ergebnis ein Fehler von

$$\left(\frac{1}{10} - \frac{209715}{2097152}\right)(3600 \cdot 10 \cdot 100) \approx 0.3433$$

Sekunden akkumuliert. Diese Abweichung beeinträchtigte die Zielsteuerung. Im Ergebnis wurde eine irakische Scud-Rakete verfehlt, die in eine amerikanische Kaserne einschlug und 28 Soldaten tötete.

Beispiel 2.3 *Ein bekanntes Beispiel aus der meteorologischen Vorhersagbarkeit liefert die Entdeckung des Lorenz-Modells (E. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci., 1963). Lorenz ging von der gerechtfertigten Annahme aus, dass die prognostischen Variablen eines Atmosphärensystems mit 3 signifikanten Ziffern ausreichend genau beschrieben sind, und druckte sie*

entsprechend aus. Beim Versuch, Modelläufe zu wiederholen, setzte er entsprechend genau ausgedruckte Werte wieder ein. Er musste feststellen, dass nach einigen simulierten Tagen die letzten Ziffern sich unterschieden. Nach 2 Monaten Integrationszeit unterschied sich der mit 3 signifikanten Ziffern wiederholte simulierte Zustand völlig von der zuerst vorgenommenen Simulation.

Bemerkung: Das letztgenannte Beispiel dient nur der Illustration der Wirkung begrenzter Zahlendarstellung. Es soll ***nicht*** *zu der aussichtslosen Forderung verleiten, dass der Atmosphärenzustand mit einem Fehler von einem Promille und besser vorab ermittelt werden sollte.*

2.3 Rundungsfehlerarithmetik

Die nachfolgende Diskussion baut auf die Behandlung der Zahlendarstellung auf Rechnern des Moduls *Datenverarbeitung und Programmieren* auf. Rechner können zwar eine sehr hohe, aber gleichwohl nur endliche Anzahl von Zahlen darstellen. Daher müssen wir annehmen, dass Eingabedaten und Lösungsdaten eine endliche Menge bilden. Bei endlicher Stellenzahl t ist naturgemäß die Menge $\mathcal{A}$ der auf Computern darstellbaren Zahlen endlich. Die Elemente der Menge $\mathcal{A}$ dieser Zahlen bezeichnet man oft auch als *Maschinenzahlen*. Im Rahmen der verfügbaren Stellen sind **ganze Zahlen immer exakt** darstellbar. Unter *elementaren Operationen* versteht man zunächst die elementaren arithmetischen Operationen $+, -, \times, /$, also Addition, Substraktion, Multiplikation und Division.

Definition 2.1 *Als* ***Algorithmus*** *wird eine der Reihenfolge nach eindeutig festgelegte Sequenz von endlich vielen elementaren Rechenoperationen bezeichnet, die vorschreibt, wie aus Eingabedaten $x_1, \ldots, x_n$ die Lösung $y_1, \ldots, y_m$ berechnet werden soll.*

Im Weiteren interessieren uns nur reelle Zahlen, die als *Fließkommazahlen (floating point number)* dargestellt werden. Alle folgenden Ergebnisse lassen sich sinngemäß auch auf komplexe Zahlen übertragen. Siehe hierzu auch die entsprechenden Teile der Vorlesung *Datenverarbeitung und Programmieren*.

Definition 2.2 *Unter* ***Rundung*** *einer Zahl $x \notin \mathcal{A}$ versteht man eine Abbildung* $\mathrm{rd}(x)$*, die die Zahl x so approximiert, dass*

$$\mathrm{rd}(x) \in \mathcal{A}, \quad \textit{wobei} \quad |x - rd(x)| \leq |x - g| \quad \forall g \in \mathcal{A}. \tag{2.1}$$

Bei *t-stelliger normalisierter Dezimaldarstellung* $x = a \cdot 10^b, |a| \geq 10^{-1}$ mit $|a| = 0.a_1a_2 \ldots a_t a_{t+1} \ldots, \quad 0 \leq a_i \leq 9, \quad a_1 \neq 0$, kann man die Rundung rd definieren als $\mathrm{rd}(x) := \mathrm{sign}(x) \cdot a' \cdot 10^b$ mit

$$a' := \begin{cases} 0.a_1a_2 \ldots a_t & \text{falls } 0 \leq a_{t+1} \leq 4 \\ 0.a_1a_2 \ldots a_t + 10^{-t} & \text{falls } a_{t+1} \geq 5 \end{cases} \tag{2.2}$$

Hiermit werden Rechnungen also mit t *wesentlichen Stellen* vorgenommen.

Der relative Fehler ϵ lässt sich dann, wegen der Normalisierung mit $|a| \geq 1/10$ durch die *Maschinengenauigkeit* eps beschränken.

$$\left|\frac{\mathrm{rd}(x) - x}{x}\right| \leq \frac{5 \cdot 10^{-t-1}}{|a|} \leq 5 \cdot 10^{-t} =: \mathrm{eps} \tag{2.3}$$

Somit

$$\mathrm{rd}(x) = x(1+\epsilon), \quad \text{mit} \quad |\epsilon| \leq \mathrm{eps}\,. \tag{2.4}$$

2.4 Gleitpunktrechnung

Auch wenn für Zahlen $x, y \in \mathcal{A}$ gilt, ist nicht gewährleistet, dass das Ergebnis der elementaren arithmetischen Verknüpfungen $\pm, \times, /$ ebenfalls exakt auf der Maschine darstellbar ist. Statt der exakten Operationen sind hier nur die sogennannten *Gleitpunktoperationen* $\tilde{+}, \tilde{-}, \tilde{\times}, \tilde{/}$ verfügbar. Man schreibt häufig mit dem Ausdruck $\mathrm{gl}(E)$ den Wert der mit Gleitpunktrechnung ermittelten elementaren Operation

$$\left.\begin{array}{rrr} x\tilde{+}y := & \mathrm{gl}(x+y) := & (x+y)(1+\epsilon_1) \\ x\tilde{-}y := & \mathrm{gl}(x-y) := & (x-y)(1+\epsilon_2) \\ x\tilde{\times}y := & \mathrm{gl}(x\times y) := & (x\times y)(1+\epsilon_3) \\ x\tilde{/}y := & \mathrm{gl}(x/y) := & (x/y)(1+\epsilon_4) \end{array}\right\} \; |\epsilon_i| \leq \mathrm{eps}\,. \tag{2.5}$$

So gilt z.B. für die Gleitpunktsummierung $\tilde{+}$ wegen (2.3)

$$x\tilde{+}y = x, \quad \text{falls} \quad |y| < \frac{eps}{B}|x|\;, \tag{2.6}$$

mit B als Basis des Zahlensystems. Die im Vergleich zu x kleine Zahl y ist damit verloren und kann ggf. nur durch Umstellen des Algorithmus erhalten werden.

Ein weiteres Problem ist die *Auslöschung*: Wenn zwei fast gleich große Zahlen voneinander abgezogen werden, so geht von den t wesentlichen Stellen der führende Teil verloren.

Beispiel 2.4 *Für die Berechnung von* $99 - 70\sqrt{2}$ *gibt es zwei weitere, mathematisch äquivalente Ausdrücke:*

$$99 - 70\sqrt{2} = \sqrt{9801} - \sqrt{9800} = \frac{1}{\sqrt{9801} + \sqrt{9800}}$$

Je nach Wahl der wesentlichen Stellen t *und Berechnungsmethode, gehen bei 2, 4 oder 6 wesentlichen Stellen alle, 4 oder 3 Stellen verloren (siehe Übungen). Hier ist im ersten Fall die numerische Instabilität durch die Subtraktion von fast gleich großen Zahlen ausgelöst.*

2.5 Fehlerfortpflanzung

Wir konkretisieren nun den Begriff des Algorithmus auf die Bestimmung des Wertes $y = \phi(x)$ einer Funktion $\phi : D \to \mathbb{R}^m,\ D \subseteq \mathbb{R}^n$, bestehend aus den Komponenten $y_i = \phi_i(x_1, \ldots, x_n),\ i = 1, \ldots, m$. Ein **Algorithmus** ist in dieser Formulierung nun eine eindeutige, endliche Sequenz von elementaren Rechenvorschriften zur Berechnung von $y = \phi(x)$.

Sei $\tilde{x}$ ein Näherungswert für x. So ist $\Delta x_j := \tilde{x}_j - x_j$ der **absolute Fehler** und $\epsilon_{x_j} := \frac{\tilde{x}_j - x_j}{x_j}, x_j \neq 0$ der **relative Fehler** der j-ten Komponente von $\tilde{x}$. Bei der *differentiellen Fehleranalyse* vernachlässigt man Terme höherer als 1. Ordung. Dann gilt für den Fehler der Ausgabedaten Δy:

$$\Delta y = \begin{pmatrix} \Delta y_1 \\ \vdots \\ \Delta y_m \end{pmatrix} \cong \begin{pmatrix} \frac{\partial \phi_1}{\partial x_1} & \cdots & \frac{\partial \phi_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \phi_m}{\partial x_1} & \cdots & \frac{\partial \phi_m}{\partial x_n} \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{pmatrix} = D\phi(x) \cdot \Delta x \tag{2.7}$$

wobei $D\phi(x)$ die Funktionalmatrix (Jacobi-Matrix, Ableitungsmatrix) von ϕ an der Stelle x ist.

Für den relativen Fehler 1. Ordnung kann man nach komponentenweiser Division von (2.7) durch $\phi_i(x) \neq 0$ schreiben

$$\epsilon_{y_i} := \sum_{j=1}^{n} \frac{x_j}{\phi_i(x)} \cdot \frac{\partial \phi_i(x)}{\partial x_j} \cdot \epsilon_{x_j} \tag{2.8}$$

Der Verstärkungsfaktor $\boxed{\frac{x_j}{\phi_i(x)} \cdot \frac{\partial \phi_i(x)}{\partial x_j}}$ beschreibt die verstärkende oder abschwächende Wirkung der Funktion ϕ auf das Ergebnis, ausgelöst durch einen Eingangsfehler ϵ_{x_j} in x_j. Er wird oft auch *Konditionszahl* genannt[1]. Je nachdem ob ϵ_{y_i} groß oder klein ist, wird das Problem *schlecht* oder *gut konditioniert* genannt (*well-conditioned* oder *ill-conditioned*).

Für die elementaren arithmetischen Operationen findet man nun mittels (2.8):

$$\begin{aligned} \phi_{x \pm y}(x, y) &:= x \pm y & \epsilon_{x \pm y} &\approx \frac{x}{x \pm y}\epsilon_x \pm \frac{y}{x \pm y}\epsilon_y \\ \phi_{x \cdot y}(x, y) &:= x \cdot y & \epsilon_{x \cdot y} &\approx \epsilon_x + \epsilon_y \\ \phi_{x/y}(x, y) &:= x/y & \epsilon_{x/y} &\approx \epsilon_x - \epsilon_y \end{aligned} \tag{2.9}$$

Es zeigt sich also, dass nur die Subtraktion und die damit einhergehende Auslöschung *Fehlerverstärkungen* hervorrufen kann, und damit als *gefährliche Operation* gelten muß. Die anderen Fälle wirken neutral oder sind gar *fehlerdämpfend*.

[1] In einem späteren Kapitel wird eine alternative und häufiger gebrauchte Form der Konditionszahl eingeführt werden.

Wir sind nun an der Fortpflanzung von Rundungsfehlern in gegebenen Algorithmen, d.h. allgemein bei Sequenzen von elementaren Operationen interessiert. Die folgende formalisierte Beschreibung der differentiellen Fehleranalyse kann auch zur allgemeinen Fehlerrechnung und Stabilitätsrechnung komplexer geophysikalischer und meteorologischer Modelle verwandt werden, die Teil der Inversen Modellierung und Vorhersagbarkeitsproblematik ist. Der Algorithmus ϕ setze sich nun aus r Zwischenergebnissen zusammen:

$$\phi^{(i)} : D_i \to D_{i+1}, \quad i = 0, \ldots, r \quad D_j \subseteq \mathbb{R}^{n_j} \tag{2.10}$$

$$y = \phi(x) = \phi^{(r)} \circ \phi^{(r-1)} \circ \cdots \circ \phi^{(0)}(x^{(0)}) \quad D_0 = D \quad D_{r+1} \subseteq \mathbb{R}^{n_{r+1}} = \mathbb{R}^m . \tag{2.11}$$

Nach der Kettenregel gilt $D(f \circ g)(x) = Df(g(x)) \cdot Dg(x)$. Wir sind nun am Fehler $\Delta x^{(i+1)}$ des Näherungswertes $\tilde{x}^{(i+1)} = \mathrm{gl}(\phi^{(i)}(\tilde{x}^{(i)}))$ interessiert. Für diesen Fehler gilt:

$$\Delta x^{(i+1)} = \underbrace{\left(\mathrm{gl}(\phi^{(i)}(\tilde{x}^{(i)})) - \phi^{(i)}(\tilde{x}^{(i)})\right)}_{\text{Gleitpunktrechnungsfehler}} + \underbrace{\left(\phi^{(i)}(\tilde{x}^{(i)}) - \phi^{(i)}(x^{(i)})\right)}_{\text{Rundungsfehler}} \tag{2.12}$$

Nimmt man für die erste Klammer an, dass die Gleitpunktauswertung gl das gerundete exakte Ergebnis liefert, so kann man setzen

$$\tilde{x}^{(i+1)} = \mathrm{gl}(\phi^{(i)}(\tilde{x}^{(i)})) = \mathrm{rd}(\phi^{(i)}(\tilde{x}^{(i)})) = (I + E_{i+1}) \cdot \phi^{(i)}(\tilde{x}^{(i)}), \tag{2.13}$$

wobei I die Einheitsmatrix ist, und die Fehlermatrix E_{i+1} definiert ist als

$$E_{i+1} := \mathrm{diag}(\epsilon_j), \quad |\epsilon_j| \leq eps, \quad j = 1, \ldots, n_{i+1} .$$

Dabei ist ϵ_j der in der j-ten Komponente von $\phi^{(i)}$ auftretende relative Rundungsfehler. Für eine Abschätzung des entstehenden Fehlers ist es zulässig, $\tilde{x}^{(i)} \approx x^{(i)}$ anzunehmen, so dass

$$\mathrm{gl}(\phi^{(i)}(\tilde{x}^{(i)})) - \phi^{(i)}(\tilde{x}^{(i)}) \approx E_{i+1} \cdot \phi^{(i)}(x) = E_{i+1} \cdot x^{i+1} =: \alpha_{i+1} \tag{2.14}$$

α_{i+1} ist der bei der Berechnung des $i+1$. Zwischenergebnisses in Gleitpunktarithmetik entstehende absolute Rundungsfehler.

Ferner gilt für die 2. Klammer in (2.12) nach Gleichung (2.7)

$$\phi^{(i)}(\tilde{x}^{(i)}) - \phi^{(i)}(x^{(i)}) =: D\phi^{(i)}(x^{(i)}) \cdot \Delta x^{(i)} \tag{2.15}$$

und damit für den Gesamtfehler (Gleitpunkt- und Rundungsfehler), der bei der Berechnung von $\tilde{x}^{(i+1)}$ aus $\tilde{x}^{(i)}$ entsteht:

$$\Delta x^{(i+1)} \cong \alpha_{i+1} + D\phi^{(i)}(x^{(i)}) \cdot \Delta x^{(i)} = E_{i+1} \cdot x^{i+1} + D\phi^{(i)}(x^{(i)}) \cdot \Delta x^{(i)}. \tag{2.16}$$

Damit gilt für den Gesamtalgorithmus ϕ die Fehlerabschätzung, wenn man mit $\psi^{(i)} := D\phi^{(r)} \ldots D\phi^{(i)}$ den aus dem Rundungsfehler der i-tem Zwischenrechnung folgenden "Restfehler" bezeichnet

$$\begin{aligned} \Delta y &= \Delta x^{(r+1)} \\ &\cong D\phi^{(r)} \ldots D\phi^{(0)} \cdot \Delta x + D\phi^{(r)} \ldots D\phi^{(1)} \cdot \alpha_1 \ldots + \alpha_{r+1} \\ &= D\phi \cdot \Delta x + D\psi^{(1)} \cdot E_1 \cdot x^{(1)} + \ldots + E_{r+1} \cdot y.+ \end{aligned} \tag{2.17}$$

Definition 2.3 *Man nennt einen Algorithmus* ***numerisch stabiler*** *als einen zweiten Algorithmus zur Berechnung von $\phi(x)$, falls der Gesamteinfluß der Rundungsfehler bei diesem kleiner ist als bei jenem.*

Beispiel 2.5 *Wegen der Identität $a^2 - b^2 = (a+b)(a-b)$ kann man den Wert dieses Ausdruckes auf beiderlei Weise berechnen. Welcher Algorithmus ist stabiler?*
Für Algorithmus $\phi(a,b) = a^2 - b^2$ folgt

$$x^{(0)} = \begin{bmatrix} a \\ b \end{bmatrix}, \quad x^{(1)} = \begin{bmatrix} a^2 \\ b \end{bmatrix}, \quad x^{(2)} = \begin{bmatrix} a^2 \\ b^2 \end{bmatrix}, \quad y = x^{(3)} = a^2 - b^2$$

$$\begin{aligned} D\phi &= (2a, -2b) \\ \alpha_1 &= E_1 x^{(1)} \overset{(2.14)}{=} \begin{pmatrix} \epsilon_1 a^2 \\ 0 \end{pmatrix} \\ \alpha_2 &= E_2 x^{(2)} = \begin{pmatrix} 0 \\ \epsilon_2 b^2 \end{pmatrix} \\ \alpha_3 &= E_3 x^{(3)} = \epsilon_3 (a^2 - b^2), \qquad \epsilon_i \leq \text{eps} \quad \textit{für} \quad i = 1,2,3 \end{aligned}$$

Mit $\Delta x = (\Delta a, \Delta b)^T \quad \Rightarrow$

$$\Delta y \tilde{=} \underbrace{2a\Delta a - 2b\Delta b}_{Rundungsfehler} + \underbrace{a^2\epsilon_1 - b^2\epsilon_2 + (a^2 - b^2)\epsilon_3}_{Gleitpunktrechnungsfehler} \tag{2.18}$$

Für den Algorithmus $\phi(a,b) = (a+b)(a-b)$ erhält man entsprechend

$$x^{(0)} = \begin{bmatrix} a \\ b \end{bmatrix}, \quad x^{(1)} = \begin{bmatrix} a+b \\ a-b \end{bmatrix}, \quad y = x^{(2)} = a^2 - b^2 \tag{2.19}$$

$$D\phi = (2a, -2b) \tag{2.20}$$

$$\alpha_1 = E_1 x^{(1)} \overset{(2.14)}{=} \begin{pmatrix} \epsilon_1(a+b) \\ \epsilon_2(a-b) \end{pmatrix} = \begin{pmatrix} \epsilon_1 & 0 \\ 0 & \epsilon_2 \end{pmatrix} \begin{pmatrix} (a+b) \\ (a-b) \end{pmatrix} \tag{2.21}$$

$$\alpha_2 = \epsilon_3 (a^2 - b^2), \qquad \epsilon_i \leq \text{eps} \quad \textit{für} \quad i = 1,2,3 \tag{2.22}$$

Mit $\Delta x = (\Delta a, \Delta b)^T \quad \Rightarrow$

$$\Delta y \tilde{=} 2a\Delta a - 2b\Delta b + (a^2 - b^2)(\epsilon_1 + \epsilon_2 + \epsilon_3) \tag{2.23}$$

Für $3|a^2 - b^2| \leq a^2 + b^2 + |a^2 - b^2|$ ist Algorithmus 2 im Falle von $1/3 < |a/b|^2 < 3$ numerisch stabiler als Algorithmus 1, andernfalls ist Algorithmus 1 vorzuziehen.

Es bleibt insgesamt aber festzuhalten, dass unabhängig von der Algorithmenauswahl in (2.18) die Abschätzung $|E_{r+1}y| \leq \text{eps}\,|y|$ gilt. Ferner bleibt vom Eingangsfehler $|\Delta^{(0)}x| \leq \text{eps}\,|x|$, sofern nicht x mit t Stellen exakt dargestellt werden kann. Damit hat man als *unvermeidbaren Fehler* immer

$$\boxed{\Delta^{(0)}y := \text{eps} \cdot (|D\phi(x)| \cdot |x| + |y|)\,.} \tag{2.24}$$

Da diese Grenze nicht unterschritten werden kann, braucht man von keinem Algorithmus verlangen, kleinere Fehler zu liefern. Daher bezeichnet man einen absoluten oder relativen Rundungsfehler α_i, E_i als *harmlos*, wenn gilt

$$\left|D\psi^{(i)}(x^{(i)}) \cdot \alpha_i\right| = \left|D\psi^{(i)}(x^{(i)}) \cdot E_i x^{(i)}\right| \approx \Delta^{(0)}y \tag{2.25}$$

Algorithmen, die dies leisten, heißen *gutartig*.

2.5.1 Beispiel

(Entnommen Stoer I) Berechnung der Lösung einer quadratischen Gleichung $y = \phi(p,q). = -p + \sqrt{p^2+q}$. Zwei Problemfälle sind unmittelbar erkennbar: 1. $p^2 \approx -q$ und 2. $p < 0, q > 0, p \gg q$.
1. Fall:
Es sei also erstmal $p > 0$. Berechne (nach 2.18):

$$\frac{\partial\phi}{\partial p} = -1 + \frac{p}{\sqrt{p^2+q}} = \frac{-y}{\sqrt{p^2+q}}, \qquad \frac{\partial\phi}{\partial q} = \frac{1}{2\sqrt{p^2+q}}. \tag{2.26}$$

Mit $q = y^2 + 2yp$

$$\begin{aligned} \epsilon_y \approx \frac{p}{\sqrt{p^2+q}} &= \frac{-y}{\sqrt{p^2+q}}\epsilon_p + \frac{q}{2y\sqrt{p^2+q}}\epsilon_q \\ &= -\frac{p}{\sqrt{p^2+q}} = \frac{-y}{\sqrt{p^2+q}}\epsilon_p + \frac{p+\sqrt{p^2+q}}{2\sqrt{p^2+q}}\epsilon_q \end{aligned}$$

Damit ist ϕ für $q > 0$ gut konditioniert, und für $q \approx -p^2$ schlecht konditioniert, denn

$$\left|\frac{p}{\sqrt{p^2+q}} = \frac{-y}{\sqrt{p^2+q}}\right| \leq 1, \qquad \frac{p+\sqrt{p^2+q}}{2\sqrt{p^2+q}} \leq 1 \quad \text{für} \quad q > 0.$$

2. Fall:
Wir erhalten aus Fall 1 und der Abschätzung des unvermeidbaren Fehlers mit (2.24)

$$\text{eps} \leq \epsilon_y^{(0)} := \frac{\Delta^{(0)}y}{y} \leq 3\ \text{eps}.$$

Wegen $P < 0, q > 0, p \gg q$ kann man nun 2 Algorithmen prüfen:

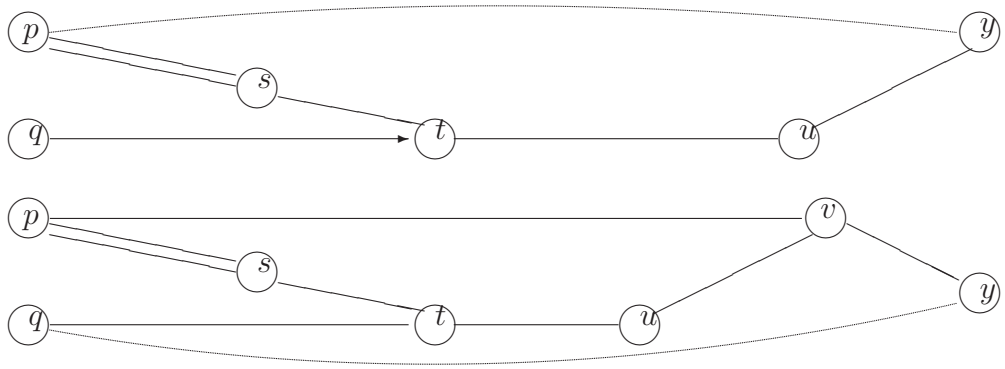


Abbildung 2.1: Graphen zu den Beispielalgorithmen 1 (oben) und 2 (unten).

Algorithmus 1 Wir setzen: $s := p^2,\ t := s+q,\ u := \sqrt{t},\ y := -p+u$. Auf Grund der Voraussetzungen ist bei $y := -p+u$ Auslöschung zu erwarten. Die Fehlerfortpflanzung in der Restabbildung, die aus der Gleitpunktberechnung der Quadratwurzel $\mathrm{gl}(\sqrt{t}) = \sqrt{t}\cdot(1+\epsilon), \quad |\epsilon| \le \mathrm{eps}$ folgt, ist mit $y = \psi(u) = -p+u$ nach (2.18)

$$\epsilon_y \approx \frac{u}{y}\frac{\partial\psi(u)}{\partial u}\epsilon_u = \frac{\sqrt{p^2+q}}{-p+\sqrt{p^2+q}}\epsilon_u = \frac{1}{q}(p\sqrt{p^2+q}+p^2+q)\epsilon =: k\epsilon$$

mit dem Verstärkungsfaktor $k > 2p^2/q > 0$ ist f"r $p \gg u$ deutlich größer als der unvermeidbare Fehler $\epsilon_y^{(0)}$.

Algorithmus 2 Hier wird der obige letzte Schritt $y := -p+u$ durch die Schritte $v := p+u, \quad y := q/v$ ersetzt. Die Fehlerfortpflanzung in der Restabbildung, die aus der Gleitpunktberechnung der Quadratwurzel folgt, ist mit $y = \psi(u) = \frac{q}{p+u}$ nach (2.18)

$$\begin{aligned}\epsilon_y &\approx \frac{u}{y}\frac{\partial\psi(u)}{\partial u}\epsilon_u = \frac{-qu}{y(p+u)^2}\epsilon_u \\ &= \frac{-q\sqrt{p^2+q}}{(-p+\sqrt{p^2+q})(p+\sqrt{p^2+q})^2}\epsilon_u = -\frac{\sqrt{p^2+q}}{p+\sqrt{p^2+q}}\epsilon_u =: k\epsilon_u\end{aligned}$$

Hier ist der Verstärkungsfaktor wegen $|k| < 1$ klein und der Algorithmus damit gutartig.

2.6 Neue Begriffe

Maschinenzahl, Maschinengenauigkeit, Algorithmus, Rundung, Fließkommazahl, t-stelliger normalisierter Dezimaldarstellung, wesentlichen Stellen, absoluter und relativer Fehler, elementare Operationen, Gleitpunktoperation,

Auslöschung, Verstärkungsfaktor, numerische (In)stabilität, Konditionszahl, schlecht - gut konditioniert, gefährliche Operation, Fehlerdämpfung, differentiellen Fehleranalyse, unvermeidbarer Fehler, harmloser Rundungsfehler, gutartiger Algorithmus.

2.7 Fragen

:

1. Welche Eigenschaften kennzeichnen einen Algorithmus?
2. Wann wird ein Problem als gut oder schlecht konditioniert bezeichnet?
3. Welcher der Gleitpunktoperatoren kann besonders fehlerverstärkend wirken?
4. Was kennzeichnet einen gutartigen Algorithmus?

Kapitel 3

Interpolation

Die Interpolation berechnet bei geophysikalischen Anwendungen oftmals die Werte für ein vorgegebenes Gitter, wobei Werte auf einem nicht unmittelbar nutzbaren Orten zur Verfügung stehen. Oftmals sind verfügbare Daten Messungen, während man die Werte auf einem anderen Gitter benötigt. Die gegebenen Werte, zwischen denen interpoliert werden soll, sind massgeblich; und dieses Kapitel stellt verschiedene mathematische Verfahren vor, wie dies am sinnvollsten erreicht wird. Gleichwohl muss darauf verwiesen werden, dass bekannte geophysikalische Sachverhalte nutzbringend verwendet werden sollten. Als Beispiel dient der Druck p in der Atmosphäre, der mit der Höhe der relativen Topographie bei Isothermie mit $Z_T := Z_2 - Z_1 = R/g_0 \int_{p_2}^{p_1} T d \ln p$ in einem logarithmischen Verhältnis steht,

$$Z_T = \frac{RT}{g_0} \ln(p_1/p_2),$$

wobei Temperatur T und g_0 mittlere Schwerebeschleunigung. So ist es sinnvoll, zur Interpolation für Zwischenwerte der relativen Topographie den Logarithmus anzuwenden.

3.1 Aufgabenstellung

Gegeben seien $n+1$ Tupel oder *Stützstellen* $(x_i, f_i),\ i = 0, \ldots, n,\ x_i \neq x_k$. Soll eine gegebene Funktion $\Phi(x; a_0, \ldots, a_n)$ durch Anpassung ihrer Parameter $a_0, \ldots, a_n$ so bestimmt werden, dass bis auf Rundungsfehlergenauigkeit gilt

$$\Phi(x_i; a_0, \ldots, a_n) = f_i,\ x_i \neq x_k,\ \text{falls } i \neq k, \tag{3.1}$$

so liegt ein *Interpolationsproblem* vor. Auch wenn im Grunde ein ähnlicher Zweck verfolgt wird, so ist diese Aufgabe in geophysikalischen Anwendungen scharf zu trennen von Ausgleichsproblemen (siehe Kapitel 6.1), in denen üblicherweise fehlerbefaftete Messwerte so angepasst werden, wie es die statistisch geschätzte Güte der Messgenauigkeit zuläßt. Eine Grundlage der Ausgleichsprobleme bildet die Approximation, bei der mit möglichst wenigen

Parameter eine in der Regel größere Tupelmenge angenähert werden soll. Im Gegensatz dazu, sollen sich bei der Interpolation die Stützstellen $(x_i, f_i),\ i = 0, \ldots, n,\ x_i \neq x_k$ nur durch Rechenungenauigkeiten von den vorgegebenen Werten unterscheiden dürfen.

Im Folgenden wird nur die *lineare* Interpolation behandelt, in der als Ansatz die Linearkombination der Φ_i gilt

$$\Phi(x; a_0, \ldots, a_n) := a_0\Phi_0(x) + a_1\Phi_1(x) + \ldots + a_n\Phi_n(x), \tag{3.2}$$

wobei die $\Phi_i(x)$ gleichwohl völlig nichtlinear sein können. Ferner werden hier weitgehend nur eindimensionale Beispiele vorgestellt. Zweidiemnsionale (Qubaturen) oder höher dimensionale Probleme werden nur erwähnt. Hier ist ein Beispiel die in der Geophysik und Meteorologie häufig benutzte Spektraldarstellung von Werten durch Kugelflächenfunktionen.

Die Generalisierung zu höherdimensionalen Problemen ist ohne Weiteres möglich.

3.2 Interpolation durch algebraische Polynome

Mit dem Polynomansatz

$$P(x) \equiv a_0 + a_1x + \ldots + a_nx^n \tag{3.3}$$

der Menge der Polynome Π_n mit Grad $P \leq n$ gilt der folgende

Existenz- und Eindeutigkeitssatz: Zu beliebigen $n + 1$ Stützstellen

$$(x_i, f_i),\ i = 0, \ldots, n,\ x_i \neq x_k \text{ für } i \neq k \tag{3.4}$$

gibt es genau ein Polynom $P \in \Pi_n$ mit

$$P(x_i) = f_i \text{ für } i = 0, \ldots, n. \tag{3.5}$$

Beweis: *Eindeutigkeit*: Mit $P_1, P_2 \in \Pi_n$ gilt nach Behauptung $P_1(x_i) = P_2(x_i) = f_i,$ für $i = 0, \ldots, n,$. Dann hat $P := P_1 - P_2 \in \Pi_n$ mit Grad $P \leq n$ mindestens $n + 1$ verschiedene Nullstellen (anschaulich: $n + 1$ Schnittpunkte der Graphen von P_1, P_2, und nicht wie vom Grad des Polynoms beschränkt nur n). Dies geht aber nur, falls $P_1 \equiv P_2$. Daher gilt die Behauptung.

Existenz: Das Polynom kann nach Lagrange konstruktiv ermittelt werden. Suche Polynome $L_i, \text{grad}(L_i) = n$ mit der Eigenschaft

$$L_i(x_k) := \delta_{ik} = \begin{cases} 1 & \text{falls } i = k \\ 0 & \text{falls } i \neq k. \end{cases} \tag{3.6}$$

Dies leisten aber für diese Stützstellen nur

$$L_i(x) := \frac{(x - x_0) \ldots\ (x - x_{i-1})\ (x - x_{i+1}) \ldots\ (x - x_n)}{(x_i - x_0) \ldots (x_i - x_{i-1})(x_i - x_{i+1}) \ldots (x_i - x_n)}, \tag{3.7}$$

wegen der oben gezeigten Eindeutigkeit. Das Lösungspolynom lautet daher (**Lagrangesche Interpolationsformel**)

$$P(x) \equiv \sum_{i=0}^{n} f_i \cdot L_i(x) = \sum_{i=0}^{n} f_i \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{(x - x_k)}{(x_i - x_k)} \tag{3.8}$$

3.3 Newtonsche Interpolationsformel und Dividierte Differenzen

Bei höheren n, insbesondere wenn man an mehreren Stellen ein bestimmtes Interpolationspolynom auswerten will, empfielt sich die Anwendung des Newtonschen Algorithmus oder das Verfahren der *dividierten Differenzen*. Das Ansatzpolynom vom Grade n lautet hier

$$P(x_{01\ldots n}) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_n(x - x_0)\ldots(x - x_{n-1}) \tag{3.9}$$

(Beachte: "x_n"fehlt in dieser Formel. Es wäre dann ein Polynom $n+1$. Grades.) Die a_i sind so zu bestimmen, dass $P(x_i) = f_i, i = 0, \ldots, n$.

Das Newtonverfahren wählt jedoch nicht die Form (3.9), sondern baut aus Effizienzgründen auf die Hornerdarstellung des interpolierenden Polynoms $P \in \Pi_n$ auf, mit $P(x_i) = f_i,\ i = 0, 1, \ldots, n$

$$P(x) = (\ldots(a_n(x - x_{n-1}) + a_{n-1})(x - x_{n-2}) + \ldots + a_1)(x - x_0) + a_0 \tag{3.10}$$

Die Koeffizienten können rekursiv in der folgenden Weise berechnet werden

$$\begin{array}{lll} f_0 = & P(x_0) = & a_0 \\ f_1 = & P(x_1) = & a_0 + a_1(x_1 - x_0) \\ f_2 = & P(x_2) = & a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ \vdots = & \vdots = & \vdots \end{array} \tag{3.11}$$

Mit weniger Rechnung und einem einfacheren Programmcode lassen sich die a_k nach dem folgenden Schema berechnen

$$\begin{array}{lllll} & k = 0 & 1 & 2 & \\ x_0: & f_0 = f[x_0] & & & \\ & & f[x_0, x_1] & & \\ x_1: & f_1 = f[x_1] & & f[x_0, x_1, x_2] & \\ & & f[x_1, x_2] & & \\ x_2: & f_1 = f[x_2] & & & \\ \vdots & \vdots & \vdots & \ddots & \end{array} \tag{3.12}$$

Die Größen $f[x_i, x_{i+1}, \ldots, x_{i+k}]$ werden dabei mit folgender Rekursion berechnet

$$f_i := f[x_i], \quad f[x_i, x_{i+1}, \ldots, x_{i+k}] := \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i} \tag{3.13}$$

Dabei ist $f[x_i, x_{i+1}, \ldots, x_{i+k}]$ die k-te dividierte Differenz, und $a_k = f[x_0, x_1, \ldots, x_k]$. Sie sind also der oberen Schrägreihe direkt zu entnehmen.

3.4 Trigonometrische Interpolation

3.4.1 Orthogonalitätsrelation

Wir betrachten ein in der Regel periodisches Interpolationsproblem auf dem Intervall $I = [0, 2\pi]$ mit den äquidistanten Stützstellen $(x_k, f_k), k = 0, \ldots, N-1$, N, gegeben, $x_k = k \cdot \frac{2\pi}{N}$, f_k komplex. Im weiteren ist i die imaginäre Einheit. I ist üblicherweise leicht zu reskalieren und damit keine Beschränkung der Allgemeinheit. Bei geophysikalischen Anwendungen findet man oft Interpolationsaufgaben längs eines Breitenkreises[1]. Das trigonometrische Polynom lautet $P(x) = a_0 + a_1 \exp(ix) + a_2 \exp(2ix) + \ldots + a_{N-1} \exp((N-1)ix)$, mit $a_i \in \mathbb{C}$. Wegen der Periodizität von $\exp(ix)$ gilt $P(x_k) = f_k, \quad k = \ldots -2, -1, 0, 1, 2, \ldots$ und $f_k = f_{k+nN}$, n ganz.

Wegen seiner Bedeutung zeigen wir den folgenden
Existenz- und Eindeutigkeitssatz: Zu komplexen Zahlen $f_k, k = 0, \ldots, N-1$ gibt es genau ein trigonometrisches Polynom

$$P(x) = a_0 + a_1 \exp(ix) + a_2 \exp(2ix) + \ldots + a_{N-1} \exp((N-1)ix) \tag{3.14}$$

mit $P(x_k) = f_k, k = 0, \ldots, N-1$. Dabei gilt

$$\sum_{k=0}^{N-1} \exp(ijx_k) \exp(-ihx_k) = \begin{cases} N & \text{falls } j = h \\ 0 & \text{falls } j \neq h, 0 \leq j, h \leq N-1. \end{cases} \tag{3.15}$$

Beweis: $(\exp(2ik\pi/N))^N = (\exp(ix_k))^N$ ist für alle $k = 0, \pm 1, \pm 2, \ldots$ eine Wurzel, d.h. eine Nullstelle, des Polynoms

$$0 = (\exp(ix))^N - 1 = \underbrace{(\exp(ix) - 1)}_{\neq 0 \ \forall k \neq 0, N, 2N, \ldots} \underbrace{\left(\exp(ix)^{N-1} + \exp(ix)^{N-2} + \ldots + \exp(ix) + 1\right)}_{\Rightarrow = 0 \ \forall \ k = 1, \ldots, N-1, \text{ sonst } = N} \tag{3.16}$$

[1]Bei Interpolationen von Pol zu Pol sind allerdings in der Regel keine trigonometrischen Polynome angebracht. Hier werden oft Linearkombinationen von Legendrepolynome genommen, die bei zweidimensionalen Problemen der Kugeloberfläche mit trigonometrischen Polynomen zu Kugelflächenfunktionen faktorisiert werden.

Daher gilt mit der rechten Klammer

$$\sum_{l=0}^{N-1} (\exp(2\pi ik/N))^l = \begin{cases} N & \text{falls } k = 0, \pm N, \pm 2N, \ldots \\ 0 & \text{sonst.} \end{cases} \tag{3.17}$$

Damit folgt die Behauptung. $\diamondsuit$

Man führt nun im komplexen Vektorraum $\mathbb{C}^N$ der N-Tupel $f = (f_0, f_1, \ldots, f_{N-1}) \in \mathbb{C}^N$ ein Skalarprodukt ein

$$\langle f, g \rangle := \frac{1}{N} \sum_{k=0}^{N-1} f_k \bar{g}_k, \tag{3.18}$$

$\bar{g}$ komplex konjugiert. Wählt man speziell für f, g,
$w_j := ((\exp(ix_0))^j, (\exp(ix_1))^j, \ldots, (\exp(ix_{N-1}))^j)\,,\ j = 0, \ldots, N-1$ so erhält man damit eine **Orthogonalitätsrelation**

$$\langle w_j, w_k \rangle = \delta_{jk} = \begin{cases} 1 & \text{falls } j = k \\ 0 & \text{falls } j \neq k, 0 \leq j, k \leq N-1. \end{cases} \tag{3.19}$$

Satz: (Diskrete Fouriertransformation)
Das trigonometrische Polynom $P(x) = \sum_{k=0}^{N-1} a_k \exp(ikx)$ erfülle $P(x_k) = f_k,\ k = 0, \ldots, N-1$, so gilt

$$a_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k \exp(-\frac{2\pi jki}{N}),\ j = 0, \ldots, N-1 \tag{3.20}$$

Beweis: Da $P(x_k) = f_k$, folgt mit der Orthogonalitätsrelation (3.19)

$$\frac{1}{N} \sum_{k=0}^{N-1} f_k w_k^{-j} = \langle f_k, w_j \rangle = \langle a_0 w_0 + a_1 w_1 + \ldots + a_j w_j + \ldots + a_{N-1} w_{N-1}, w_j \rangle = a_j. \tag{3.21}$$

$\diamondsuit$

Man erhält auf diese Weise die Koeffizienten a_j einer diskreten Fourieranalyse der Stützstellenwerte f_j. Zur Reproduktion hat man die Reihenentwicklung bis zum Abschneidewert (truncation number) $N-1$ fortzuführen. Allerdings haben die Polynome mit geringerem Abschneidewert $0 \leq s \leq N-1$ die folgende, in der Praxis äußerst wichtige Minimaleigenschaft:

Satz: Von allen trigonometrischen Polynomen $Q_s(x) = b_0 + b_1 \exp(ix) + \ldots + b_s \exp(six),\ s < N-1$ minimiert das s-te Abschnittspolynom P_s von P den quadratischen Fehler

$$S(Q_s) := \sum_{k=0}^{N-1} |f_k - Q_s(x_k)|^2\,. \tag{3.22}$$

Dabei ist $S(Q_{N-1}) = S(P) = 0$

3.4.2 Schnelle Fouriertransformation

Bei der Berechnung der N Fourierkoeffizienten in (3.20) werden $N \cdot N$ Operationen[2] vorgenommen. Der Algorithmus ist also von quadratischer Ordnung $\mathcal{O}(N^2)$. Insbesondere bei großen N ist dies ein zu teurer Algorithmus. So hat seit Januar 2010 das Modell des Europäischen Zentrums für mittelfristige Wettervorhersage (ECMWF) eine Abschneidezahl von $T = N - 1 = 1279, \quad \approx$ 16 km horizontaler Gitterabstand für die pro Zeitschritt 2-fach auszuführende Fouriertransformation. Eine Abhilfe schafft die *Schnelle Fouriertransformation (Fast Fourier Transformation, FFT)*. Es gibt mehrere Varianten (Cooley and Tukey, 1965; Gentleman and Sande, 1966). Die Verfahren beruhen allerdings auf spezielle Möglichkeiten, Berechnungen zu sparen, wenn N sich in möglichst kleine und wenige unterschiedliche Primfaktoren faktorisieren läßt. So gilt für das oben genannte Beispiel $T + 1 = N = 1280 = 2^8 \cdot 5$ (frühere Version: $T+1 = N = 800 = 2^5 \cdot 5^2$). Die beste Effizienz wird mit 2-er-Potenzen $N = 2^n$ erzielt. Wegen seiner großen Bedeutung wird im folgenden der Algorithmus von Sande und Turkey skizziert (Gentleman and Sande, 1966).

Mit der Abkürzung $\varepsilon_m := \exp(-\frac{2\pi i}{2^m})$ folgt unmittelbar $\varepsilon_n^2 = \varepsilon_{n-1}$. Gilt ferner $M = N/2$, so folgt $\varepsilon_n^M = \exp(-\pi i) = -1$ für $f_k \in \mathbb{C}, \quad h = 0, \ldots, N-1$

$$a_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k \exp(-\frac{2\pi jki}{N}) = \frac{1}{N} \sum_{k=0}^{N-1} f_k \varepsilon_n^{j \cdot k}. \tag{3.23}$$

Mit $h = 0, \ldots, M-1$ kann man in Koeffizienten gerader und ungerader Indices aufteilen

$$\begin{array}{lcllcl} N \cdot a_{2k} & = & \sum_{k=0}^{N-1} f_k \varepsilon_n^{2hk} & = & \sum_{k=0}^{M-1} (f_k + f_{k+M}) \varepsilon_{n-1}^{hk} & =: \sum_{k=0}^{M-1} f'_k \varepsilon_{n-1}^{hk} \\ N \cdot a_{2k+1} & = & \sum_{k=0}^{N-1} f_k \varepsilon_n^{(2h+1)k} & = & \sum_{k=0}^{M-1} ((f_k - f_{k+M}) \varepsilon_n^k) \varepsilon_{n-1}^{hk} & =: \sum_{k=0}^{M-1} f''_k \varepsilon_{n-1}^{hk} \end{array} \tag{3.24}$$

Man erhält damit 2 Fouriertransformationen wie (3.23), aber von jeweils halber Länge. Dieser erste Schritt reduziert die Komplexität auf $\mathcal{O}(2(N/2)^2)$. Führt man diese Halbierung mit gleichem Schema weiter fort, so erhält man mit $M := 2^{m-1}$ und $R := 2^{n-m}$

$$N \cdot a_{jR-r} = \sum_{k=0}^{2M-1} f_{rk}^{(m)} \varepsilon_m^{jk} \quad r = 0, \ldots, R-1, \; j = 0, \ldots, 2M-1, \tag{3.25}$$

wobei die $f_{rk}^{(m)}$ mit folgender Rekursion berechnet werden

$$\begin{array}{lcll} f_{0k}^{(n)} & = & f_k, & k = 0, \ldots, N-1 \\ f_{rk}^{(m-1)} & = & f_{rk}^{(m)} + f_{r,k+M}^{(m)} & \\ f_{r+R,k}^{(m-1)} & = & (f_{rk}^{(m)} - f_{r,k+M}^{(m)}) \varepsilon_m^k & . \end{array} \tag{3.26}$$

Dabei ist $m = n, n-1, \ldots, 1, \quad r = 0, 1, \ldots, R-1, \quad R = 2^{n-m}, \quad k = 0, 1, \ldots, M-1, \quad M = 2^{m-1}$

Es läßt sich zeigen, dass die Komplexität des Algoritmus von der wesentlich günstigeren Ordnung $\mathcal{O}(N \cdot (\ln N))$ ist.

[2]Meist wird unter einer *Operation* eine Multiplikation und eine Addition verstanden.

3.5 Spline-Interpolation

Zwar kann man bei algebraischen Polynomen beliebig viele Stützstellen einführen, sofern hierzu die Wertepaare gegeben sind. Allerdings nimmt die Glätte der Interpolationsfunktion ab und Überschwinger lassen sich nicht unterdrücken. Präziser ausgedrückt, im allgemeinen konvergieren Interpolationspolynome mit wachsendem Grad nicht gegen eine gegebene zu interpolierende Funktion. Dank der Minimaleigenschaften und der damit verbundenen Orthogonalitätsrelation leisten dies sie trigonometrischen Polynome. Die Voraussetztungen zur Anwendung der trigonometrischen Interpolation sind oben bereits genannt worden. Die Koeffizienten müssen in einem gesonderten Rechengang ermittelt werden. Ferner sind sie auf periodische Randbedingungen und äquidistante Stützstellen eingeschränkt. Wünschenswert wäre eine Verbindung der Eigenschaften beider Interpolationsverfahren. Diese Eigenschaft erfüllt im gewissen Sinne die Interpolation mit Splines[3].

3.5.1 Kubische Splines

Die folgende Darstellung wird nur kursorisch sein.

Ein Interpolationsintervall $[a, b]$ sei durch die nicht notwendigerweise äquidistante Lage der Stützstellen $\Delta := \{a = x_0 < x_1 < \ldots < x_n = b\}$ unterteilt.
Def.: (*Spline-Funktion*)
Eine zur Aufteilung Δ gehörende Spline-Funktion S_Δ ist eine relle Funktion $S_\Delta : [a, b] \to \mathbb{R}$ mit folgenden Eigenschaften

a) S_Δ ist auf $[a, b]$ 2-mal stetig differenzierbar, d.h. $S_\Delta \in \mathcal{C}^2[a, b]$

b) Auf allen Teilintervallen $[x_i, x_{i+1}]$ $i = 0, \ldots, n-1$ stimmt S_Δ mit einem Polynom 3. Grades überein.

Wie eingangs angedeutet, bewerten wir S_Δ mittels einer durch die 2. Ableitung definierte zu minimierende Krümmung

$$\|f\|^2 := \int_a^b |f''(x)|^2 \, dx. \tag{3.27}$$

Es gilt die folgende Minimum-Norm-Eigenschaft für Splines:
Satz: (*Minimum-Norm*)
Gegeben seien $\Delta := \{a = x_0 < x_1 < \ldots < x_n = b\}$, $Y = \{y_0, \ldots, y_n\}$ und eine stetige Funktion mit $f(x_i), i = 0, 1, \ldots, n$. Dann gilt

$$\|f - S_\Delta(Y)\|^2 = \|f\|^2 - \|S_\Delta(Y)\|^2 \geq 0. \tag{3.28}$$

Wird zusätzlich eine der folgenden Bedingungen erfüllt

[3]Spline (engl.) Straklatte, d.i. ein biegsames Lineal, welches an gegebenen Stützstellen angelegt eine Kurve mit angenähert geringster Krümmung bezeichnet.

a) ("natürlicher Spline")$S''_\Delta(Y,a) = S''_\Delta(Y,b) = 0$ (Spline-Funktion hat an den Endpunkten a, b keine Krümmung), oder

b) ("periodischer Spline") f, S_Δ periodisch, also $S'_\Delta(Y,a) = S'_\Delta(Y,b), \quad S''_\Delta(Y,a) = S''_\Delta(Y,b)$, oder

c) ("vollständiger Spline") $f'(a) = S'_\Delta(Y,a), f'(b) = S'_\Delta(Y,b)$ (vorgegebene Anfangs- und Endsteigungen),

so ist die Splinefunktion $S_\Delta(Y)$ eindeutig bestimmt.

Die oben gesetzte Forderung $S_\Delta \in \mathcal{C}^2[a,b]$ liefert die lineare Bedingung für die 2. Ableitung

$$S''_\Delta(Y;x) = M_j \frac{(x_{j+1}-x)}{h_{j+1}} + M_{j+1}\frac{(x-x_j)}{h_{j+1}}, \qquad x \in [x_j, x_{j+1}],$$

wobei $h_{j+1} := x_{j+1} - x_j$ gesetzt wird. Mit dieser Definition der **Momente** $M_j := S''_\Delta(Y,x_j)$ lässt sich durch zweimalige Integration die Formel zur Berechnung von kubischen Splines herleiten. Diese lautet für $x \in [x_j, x_{j+1}]$, $j = 0, ..., n-1$ ($n+1$: Anzahl der Stützstellen):

$$\begin{aligned} S'_\Delta(Y;x) &= -M_j\frac{(x_{j+1}-x)^2}{2h_{j+1}} + M_{j+1}\frac{(x-x_j)^2}{2h_{j+1}} + A_j \\ S_\Delta(Y;x) &= M_j\frac{(x_{j+1}-x)^3}{6h_{j+1}} + M_{j+1}\frac{(x-x_j)^3}{6h_{j+1}} + A_j(x-x_j) + B_j \end{aligned} \tag{3.29}$$

mit den Integrationskonstanten

$$A_j = \frac{y_{j+1}-y_j}{h_{j+1}} - \frac{h_{j+1}}{6}(M_{j+1}-M_j) \tag{3.30}$$

$$B_j = y_j - M_j\frac{h_{j+1}^2}{6}. \tag{3.31}$$

Setzt man (3.31) in (3.29) ein, und beachtet man die Gleichheit der 1. Ableitungen an den Stützstellen $S'_\Delta(Y;x-) = S'_\Delta(Y;x+)$, so lautet nun die Bestimmungsgleichung für die Momente $M_j := S''_\Delta(Y,x_j), \quad j = 1,\ldots,n-1$

$$\frac{h_j}{6}M_{j-1} + \frac{h_j+h_{j+1}}{3}M_j + \frac{h_{j+1}}{6}M_{j+1} = \frac{y_{j+1}-y_j}{h_{j+1}} + \frac{y_{j-1}-y_j}{h_j}, \qquad j = 1,\ldots,n-1. \tag{3.32}$$

Zusätzlich zu diesen $n-1$ Gleichungen erhält man abhängig von den 3 Fällen
Fall a): $S''_\Delta(Y;a) = M_0 = 0 = M_n = S''_\Delta(Y;x+)$
Fall b): $S''_\Delta(Y;a) = M_0 = M_n = S''_\Delta(Y;b)$ und $S'_\Delta(Y;a) = S'_\Delta(Y;b)$
also $\frac{h_n}{6}M_{n-1} + \frac{h_n+h_1}{3}M_n + \frac{h_1}{6}M_1 = \frac{y_1-y_n}{h_1} + \frac{y_{n-1}-y_n}{h_n}$,
Fall c) $S'_\Delta(Y;a) = y'_0, \Rightarrow \frac{h_1}{3}M_0 + \frac{h_1}{6}M_1 = \frac{y_1-y_0}{h_1} - y'_0$
$S'_\Delta(Y;b) = y'_n, \Rightarrow \frac{h_n}{6}M_{n-1} + \frac{h_n}{3}M_n = \frac{y_n-y_{n-1}}{h_n} - y'_0$.

Das Gleichungssystem lautet nun für die Fälle a und c in Matrixschreibweise

$$\begin{pmatrix} 2 & \lambda_0 & & & & c \\ \mu_1 & 2 & \lambda_1 & & & \\ & \mu_2 & \ddots & \cdot & & \\ & \cdot & \cdot & \ddots & & \\ & & \cdot & \cdot & 2 & \lambda_{n-1} \\ c & & & & \mu_n & 2 \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \vdots \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ \vdots \\ \vdots \\ d_n \end{pmatrix}, \tag{3.33}$$

$$\lambda_j := \frac{h_{j+1}}{h_j + h_{j+1}}, \qquad \mu_j := 1 - \lambda_j \tag{3.34}$$

und

$$d_j := \frac{6}{h_j + h_{j+1}} \left(\frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j} \right), \qquad j = 1, 2, \ldots n-1. \tag{3.35}$$

Im Falle **a)** gilt

$$\lambda_0 = 0,\ d_0 = 0,\ \mu_n = 0,\ d_n = 0 \text{ und } c = 0. \tag{3.36}$$

Im Falle **c)** gilt

$$\lambda_0 = 1,\ d_0 = \frac{6}{h_1} \left(\frac{y_1 - y_0}{h_1} - y_0' \right). \tag{3.37}$$

$$\mu_n = 1,\ d_n = \frac{6}{h_n} \left(y_n' - \frac{y_n - y_{n-1}}{h_n} \right), c = 0\ . \tag{3.38}$$

Im periodischen Falle **b)** gilt

$$\lambda_n = \frac{h_1}{h_n + h_1}, \qquad \mu_n = 1 - \lambda_n \tag{3.39}$$

$$d_n = \frac{6}{h_n + h_1} \left(\frac{y_1 - y_n}{h_1} - \frac{y_n - y_{n-1}}{h_n} \right), c = \mu. \tag{3.40}$$

3.5.2 B-Splines

Eine Möglichkeit, spezielle Randwertunterscheidungen zu vermeiden, bieten Basis-Splines, kurz *B-Splines*. Dazu wird ein Funktionsraum von 2-mal stetig differenzierbaren Splinefunktionen 3-ten Grades konstruiert, die als Linearkombination die gegebenen Stützstellen interpolieren (und bei anderen Anwendungen als in diesem Kapitel auch nur approximieren) können. Die Konstruktion gelingt folgendermaßen: Definiere abgeschnittene Potenzfunktionen

$$F_x(t) := \begin{cases} (t-x)^3 & \text{falls } x \le t \\ 0 & \text{falls } x > t\ . \end{cases} \tag{3.41}$$

Ferner wird eine weitere Stützstellenmenge definiert durch

$$\{t_0, t_1, t_2, t_3, t_4, t_5, \ldots, t_n, t_{n+1}, t_{n+2}, t_{n+3}, t_{n+4}\} := \{x_0, x_0, x_0, x_0, x_2, x_3, \ldots, x_{n-2}, x_n, x_n, x_n, x_n \tag{3.42}$$

und $f_i := f_x(t_i)$ mit den dividierten Differenzen $[f_i, \ldots, f_{i+k}]$. Damit hat man für den i-ten B-Spline

$$B_i(x) := (t_{i+4} - t_i)\,[f_i, \ldots, f_{i+4}]\,, \quad i = 0, \ldots, n \tag{3.43}$$

Man kann zeigen, dass diese B-Splines linear unabhängige Basisfunktionen im Raum der kubischen Splinefunktionen sind. Diese erfüllen die folgenden Eigenschaften:

1.

$$\begin{aligned}
B_0 > 0 \text{ falls} \quad & x \in [x_0, x_2) \\
B_1 > 0 \text{ falls} \quad & x \in (x_0, x_3) \\
B_2 > 0 \text{ falls} \quad & x \in (x_0, x_4) \\
B_3 > 0 \text{ falls} \quad & x \in (x_0, x_5) \\
B_i > 0 \text{ falls} \quad & x \in (x_{i-1}, x_{i+2}) \quad i = 4, \ldots, n-4 \\
B_{n-3} > 0 \text{ falls} \quad & x \in (x_{n-5}, x_n) \\
B_{n-2} > 0 \text{ falls} \quad & x \in (x_{n-4}, x_n) \\
B_{n-1} > 0 \text{ falls} \quad & x \in (x_{n-3}, x_n) \\
B_n > 0 \text{ falls} \quad & x \in (x_{n-2}, x_n] \\
B_i(x) = 0 \text{ falls} \quad & \text{außerhalb dieser Intervalle}
\end{aligned} \tag{3.44}$$

2.

$$\sum_{i=0}^{n} B_i(x) = 1 \quad \forall x \in [x_0, x_n] \tag{3.45}$$

3. Die Matrix B mit den Koeffizienten $b_{ik} := B_k(x_i)$ hat Siebenbandgestalt und ist positiv definit.

Eine B-Splinefunktion im inneren Gebiet $i = 4, 5, \ldots, n-4$ setzt sich über seine Teilintervalle wie folgt zusammen

$$B_i(x) = \frac{1}{6h^3} \begin{cases} (x - x_{i-2})^3 & x \in [x_{i-2}, x_{i-1}] \\ h^3 + 3h^2(x - x_{i-1}) + 3h(x - x_{i-1})^2 - 3(x - x_{i-1})^3, & x \in [x_{i-1}, x_i] \\ h^3 + 3h^2(x_{i+1} - x) + 3h(x_{i+1} - x)^2 - 3(x_{i+1} - x)^3, & x \in [x_i, x_{i+1}] \\ (x_{i+2} - x)^3 & x \in [x_{i+1}, x_{i+2}] \\ 0 & \text{sonst} \end{cases} \tag{3.46}$$

Die gesuchte B-Splinefunktion $s(x)$ kann nunmehr als Linearkombination dieser Basissplines dargestellt werden

$$s(x) = \sum_{k=0}^{n} \alpha_k B_k(x), \tag{3.47}$$

also eine Aufgabe mit $n+1$ Bedingungen. Das zugehörige Gleichungssystem ist dann gegeben durch

$$f_i = \sum_{k=0}^{n} \alpha_k B_k(x_i), \quad i = 0, 1, \ldots, n \tag{3.48}$$

3.5.3 Neue Begriffe

Newtonsche Interpolationsformel, Dividierte Differenzen, Trigonometrische Interpolation, Orthogonalitätsrelation, Schnelle Fouriertransformation, Spline-Interpolation, Kubische Splines, B-Splines, lineare Polynominterpolation, Lagrangesche Interpolationsformel, Dividierte Differenzen, Trigonometrische Interpolation, Orthogonalitätsrelationen, FFT, Spline-Funktion, kontinuierliche und diskrete Gaußapproximation, Tschebyscheffpproximation

3.5.4 Fragen

1. Warum sind bei der "einfachen" Polynominterpolation die Anzahl der Stützstellen, und damit der Grad des Interpolationspolynoms begrenzt?

2. Wann eignen sich trigonometrische Interpolationsverfahren?

3. Wie reduziert sich die Komplexität der Berechnung der Koeffizienten bei trigonometrischer Interpolation durch FFT, und welche Beschränkung erzwingt dies?

4. Welchen Vor- und Nachteile haben kubische Splines gegenüber der einfachen Polynominterpolation?

5. Durch welche Bedingungen können kubische Splines eindeutig bestimmt werden?

3.5.5 Weitere Interpolationsverfahren

Weitere wichtige Interpolationsverfahren sind die rationale Interpolation durch einen Quotienten von Polynomen, die Waveletinterpolation, die mit einer Orthogonalitätsrelation beruht, ohne, dass eine periodische Form erwartet wird und eine Wirkung nur in einem bergrenzten Bereich um eine Stützstelle erwünscht ist.

Im Bereich mehrdimensionaler Interpolation gibt es in der Geostatistik Verfahren wie Kriging (Geologie zumeist), Optimale Interpolation (zumeist Meteotrologie), die eng verwandt sind, aber wegen ihrer approximativen Methodik nicht zur Interpolation im engeren Sinne gehören.

3.6 Approximation

3.6.1 Gaußapproximation

Wir behandeln nun den Fall, dass durch eine Vielzahl von gegebenen Tupeln (x_i, f_i), oder gar eine stückweise glatte Funktion f, eine auf Rundungsfehlergenauigkeit genaue Anpassung nicht möglich oder sinnvoll ist. Der letztgenannte Fall ist typisch für eine Anpassung an üblicherweise fehlerbehaftete Messungen.

Die vielfältigen Aufgabenstellungen der Gaußapproximation führten zu diversen weiteren Bezeichnungen: Datenanpassung, Ausgleichsrechnung oder Regressionsanalyse. Man unterscheidet zwischen einer kontinuiertliche und einer diskreten Gaußapproximation.

Kontinuiertliche Gaußapproximation Das folgende Problem liegt im kontinuierlichen Fall vor:
Bei einer gegebenen, auf dem Intervall (a, b) stückweise stetigen, quadratintegrablen Funktion $f : (a, b) \to \mathcal{C}^2$ und einer auf gleichem Intervall positiven Gewichtsfunktion $w(x) > 0, a \leq x \leq b$, sowie einem Funktionenansatz

$$g(x) := g(x; a_0, a_1, \ldots, a_m) := \sum_{i=0}^{m} a_i \phi_i(x) \tag{3.49}$$

mit ebenfalls gegebener Ansatzfunktionenfamilie $\phi_i, i = 0, \ldots, m$ mit m linear unabhängigen Basisfunktionen, sind die Koeffizienten $a_0, a_1, \ldots, a_m$ so zu bestimmen, dass

$$F(a_0, a_1, \ldots, a_m) := \int_a^b (f(x) - g(x))^2 w(x) dx \tag{3.50}$$

minimal ist.

Zur Berechnung dieses Minimums leitet man Gleichung (3.50) jeweils nach den a_i ab und setzt jede der Komponentengleichung $= 0$. Führt man das Skalarprodukt

$$(f, g) := \int_a^b f(x) g(x) w(x) dx \tag{3.51}$$

ein, erhält man das *Normalgleichungssystem*

$$\begin{pmatrix} (\phi_0, \phi_0) & (\phi_0, \phi_1) & \ldots & (\phi_0, \phi_m) \\ (\phi_1, \phi_0) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ (\phi_m, \phi_0) & \ldots & \ldots & (\phi_m, \phi_m) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} (f, \phi_0) \\ (f, \phi_1) \\ \vdots \\ (f, \phi_m) \end{pmatrix}, \tag{3.52}$$

Dank der angenommenen linearen Unabhängigkeit der ϕ_i ist das Gleichungssystem eindeutig lösbar. Die numerische Lösung derartiger Gleichungssysteme werden später behandelt.

Sind die ϕ_i darüber hinaus paarweise orthogonal, also

$$(\phi_i, \phi_j) = 0, \text{ falls } \quad i \neq j, \tag{3.53}$$

so erhält man direkt die Lösung

$$a_i = \frac{(f, \phi_i)}{(\phi_i, \phi_i)}, \quad i = 0, 1, \ldots, m \tag{3.54}$$

Mit der Interpolation mittels trigonometrischen Funktionen wurde ein diskretisiertes orthogonales Funktionensystem bereits vorgestellt. In diesem Zusammenhang wurde auch die approximierende Optimalitätseigenschaft benannt, falls weniger Funktionen als Stützstellen genutzt werden.

Diskrete Gaußapproximation Allgemein kann man nun für die diskrete Gaußapproximation die analoge Herleitung finden. Wählt man N Stützstellen, so findet Gleichung (3.50) seine diskrete Entsprechung in

$$F(a_0, a_1, \ldots, a_m) := \sum_{i=0}^{N} (f(x_i) - g(x_i))^2 w_i, \quad w_i > 0. \tag{3.55}$$

Das Skalarprodukt lautet entsprechend

$$(p, q) := \sum_{i=0}^{N} p(x_i) q(x_i) w_i. \tag{3.56}$$

Mit diesen Definitionen kann die Normalgleichung aufgestellt und bei linearer Unabhängigkeit der Vektoren $\phi_i(x_k), k = 0, \ldots, N, \quad i = 0, 1, \ldots, m$ eindeutig gelöst werden.

3.6.2 Approximation mit Tschebyscheffpolynomen

Als ein sehr Konstruktion Beispiel orthogonaler Basisfunktionen für die GA haben sich Tschebyscheffpolynome (TPe) erwiesen. Die effiziente Konstruktion fußt auf der Tatsache, dass die Funktionen $\cos(k\phi),\ k = 0, 1, \ldots$ durch die trigonometrischen Additionstheoreme als Polynom von $\cos(\phi)$ rekursiv, und damit einfach, darstellbar sind.

$$T_k(x) := T_k(\cos(\phi)) := \cos(k\phi), \quad \text{wobei} \quad x := \cos(\phi), \quad x \in [-1, 1] \tag{3.57}$$

Man findet mittels der Additionstheoreme der Trigonometrie als erste TP

$$\begin{aligned} \cos(0\phi) = 1 \Rightarrow \quad & T_0(x) = 1 \\ \cos(1\phi) = \cos(\phi) \Rightarrow \quad & T_1(x) = x \\ \cos(2\phi) = 2\cos^2(\phi) - 1 \Rightarrow \quad & T_2(x) = 2x^2 - 1 \\ \cos(3\phi) = 2\cos(\phi)\cos(2\phi) - \cos(\phi) = 4\cos^3(\phi) - 3\cos(\phi) \Rightarrow \quad & T_3(x) = 4x^3 - 3x \end{aligned} \tag{3.58}$$

Man ermittelt ferner

$$T_4(x) = 8x^4 - 8x^2 + 1, \qquad T_5(x) = 16x^5 - 20x^3 + 5x, \ldots \tag{3.59}$$

Die folgenden Eigenschaften lassen sich zeigen:

1. Beschränkung

$$|T_k(x)| \leq 1, \quad x \in [-1, 1], k = 0, 1, 2, \ldots \tag{3.60}$$

2. Rekursion

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x), \quad k \geq 1 \end{aligned} \tag{3.61}$$

3. Symmetrie

$$T_k(-x) = (-1)^k T_k(x) \tag{3.62}$$

4. Extremalstellen haben eine Drängung zu den Intervallrändern

$$x_l^{(e)} = \cos\left(\frac{l\pi}{k}\right), \quad l = 0, 1, \ldots, k, k \geq 1 \tag{3.63}$$

5. Nullstellen haben ebenfalls eine Drängung zu den Intervallrändern

$$x_l^{(e)} = \cos\left(\frac{(2l-1)\pi}{2k}\right), \quad l = 0, 1, \ldots, k, k \geq 1 \tag{3.64}$$

Orthogonalität bezüglich der Gewichtsfunktion $w(x) := \frac{1}{\sqrt{1-x^2}}$

$$\int_{-1}^{1} T_l(x)T_j(x)\frac{1}{\sqrt{1-x^2}}dx = \begin{cases} 0, & \text{falls} \quad l \neq j \\ \pi/2, & \text{falls} \quad l = j > 0 \\ \pi, & \text{falls} \quad l = j = 0 \end{cases} \tag{3.65}$$

3.6.3 Neue Begriffe

Gaußapproximation, Approximation mit Tschebyscheffpolynomen, Ansatzfunktionen, Normalgleichung

3.6.4 Fragen

1. Wann ist das Normalgleichungssystem eindeutig lösbar?
2. Warum gelten die Tschebyscheffpolynome als besonders effizient?

Kapitel 4

Integration

Dieser Abschnitt behandelt die numerische Berechnung bestimmter Integrale

$$\int_a^b f(x)dx, \qquad |a|, |b| < \infty. \tag{4.1}$$

4.1 Newton-Cotes-Formeln

Ein naheliegendes elementares Verfahren erhält man, indem man die Funktion $f(x)$ an geeigneten Stellen x_i auswertet und mittels der so gewonnenen Stützstellen ein interpolierendes Polynom $P(x)$ konstruiert und dann integriert, in der Erwartung $\int_a^b P(x)dx \approx \int_a^b f(x)dx$. Wir wählen für das Folgende eine äquidistante Intervalleinteilung $h := (b-a)/n, n > 0$ ganz, so dass $x_i := a + i \cdot h,\ i = 0, \ldots, n$. Mit s als geeignet gewählter neuer Variable lautet bei Transformation mittels $x = a + h \cdot s$ das Langrangesche Interpolationspolynom

$$L_i(x) = \varphi_i(s) := \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{(s-k)}{(i-k)}. \tag{4.2}$$

Damit erhält man nach Integration die *Newton-Cotes-Formeln*

$$\begin{aligned} \int_a^b P(x)dx &= \sum_{i=0}^{n} f_i \int_a^b L_i(x)dx \\ &= h \sum_{i=0}^{n} f_i \int_0^n \varphi_i(s)ds \\ &= h \sum_{i=0}^{n} f_i \alpha_i \\ &= \frac{b-a}{sn} \sum_{i=0}^{n} \sigma_i f_i, \end{aligned} \tag{4.3}$$

wobei $\sigma_i := s\alpha_i$. Dabei sind also die Gewichte (Koeffizienten)

$$\alpha_i := \int_0^n \varphi_i(s)ds \tag{4.4}$$

durch die Wahl der approximierenden Interpolationspolynome bestimmt, nicht aber durch Werte f_i oder den Intervallgrenzen a, b.

Mit $n = 1$ erhält man die bekannte Trapezregel. Die Simpsonregel erhält man mit $n = 2$, mit den Gewichten

$$\alpha_0 = \int_0^2 \frac{s-1}{0-1} \cdot \frac{s-2}{0-2} ds = \frac{1}{2}\int_0^2 (s^2 - 3s + 2)ds = \frac{1}{2}\left(\frac{8}{3} - \frac{12}{2} + 4\right) = \frac{1}{3} \tag{4.5}$$

$$\alpha_1 = \int_0^2 \frac{s-0}{1-0} \cdot \frac{s-2}{1-2} ds = \ldots = \frac{4}{3} \tag{4.6}$$

$$\alpha_2 = \int_0^2 \frac{s-0}{2-0} \cdot \frac{s-1}{2-1} ds = \ldots = \frac{1}{3} \tag{4.7}$$

Damit gewinnt man den Näherungswert

$$\int_a^b P_2(x)dx = \frac{h}{3}(f_0 + 4f_1 + f_2) \tag{4.8}$$

Die tabelliert vorliegenden Gewichte $\alpha_i,\ i = 0, 1, \ldots, n$ sind rationale Zahlen mit der Eigenschaft $\sum_{i=0}^n \alpha_i = n$. Die ersten 4 Newton–Cotes–Formeln erhält man mit

n	σ_i					$sn = \sum_{i=1}^n \sigma_i$	Fehler	Name
1	1		1			2	$\frac{h^3}{12} f^{(2)}(\xi)$	Trapezregel
2	1	4	1			6	$\frac{h^5}{90} f^{(4)}(\xi)$	Simpson-Regel
3	1	3		3	1	8	$\frac{3h^5}{80} f^{(4)}(\xi)$	3/8-Regel, (pulcherima)
4	7	32	12	32	7	90	$\frac{8h^7}{945} f^{(6)}(\xi)$	Milne–Regel

Dieses Verfahren kann man bis $n = 6$ anwenden. Bei mehr Unterteilungen treten negative Gewichte auf.

Eine allgemeine Fehlerabschätzung lautet

$$\int_a^b P_n(x)dx - \int_a^b f(x)dx = h^{p+1} \cdot K \cdot f^{(p)}(\xi),\ \xi \in (a, b) \tag{4.9}$$

Dabei hängen p und K von der Wahl von n ab.

4.2 Integration mit Intervallaufteilungen

Die Beschränkung des Polynomgrades ist jedoch nicht problematisch, da man in der Praxis ohnehin zumeist eine gewählte Formel in dem beliebig fein unterteilten Integrationsintervall *wiederholt* anwendet. Für den einfachsten Fall

der Trapezregel, d.h. $n = 1$ bei N–facher Unterteilung mit $[x_i, x_{i+1}]\,, \quad i = 0, 1, \ldots, N, \quad h := (b-a)/N$ erhält man die Teilnäherung $I_i := h/2(f(x_i) + f(x_{i+1})$. Das gesamte Integral über $[a, b]$ wird somit zu

$$T(h) := \sum_{i=0}^{N-1} \frac{h}{2} \left(f(x_i) + f(x_{i+1})\right) = h{\cdot}(f(a)/2 + f(a+h) + \ldots + f(b-h) + f(b)/2)\,. \tag{4.10}$$

Jedes Teilintervall i hat nun den Fehler

$$I_i - \int_{x_i}^{x_{i+1}} f(x)dx = \frac{h^3}{12} f^{(2)}(\xi_i), \quad \xi_i \in (x_i, x_{i+1}). \tag{4.11}$$

Nun gilt aber

$$\min_i f^{(2)}(\xi_i) \leq \frac{1}{N} \sum_{i=0}^{N-1} f^{(2)}(\xi_i) \leq \max_i f^{(2)}(\xi_i), \tag{4.12}$$

so dass aber für $f \in \mathcal{C}^2\,[a, b]$ ein $\xi \in (\min_i \xi_i, \max_i \xi_i)$ existiert mit

$$f^{(2)}(\xi) = \frac{1}{N} \sum_{i=0}^{N-1} f^{(2)}(\xi_i). \tag{4.13}$$

Daher läßt sich für das Gesamtintervall abschätzen

$$T(h) - \int_a^b f(x)dx = \frac{h^3}{12} \frac{(b-a)}{h} \sum_{i=0}^{N-1} \frac{1}{N} f^{(2)}(\xi_i) = \frac{h^2(b-a)}{12} f^{(2)}(\xi), \quad \xi \in (a, b). \tag{4.14}$$

Die wiederholte Anwendung der Trapezformel ist also ein Verfahren von quadratischer Ordnung in h, falls $f \in \mathcal{C}^2\,[a, b]$.

Ein weiteres Verfahren zur Verbesserung der Genauigkeit der Trapez– und Simpsonregel ist die **Romberg-Integration**. Die Taylorreihe des Fehlers läßt sich schreiben

$$T(h) - I = c_1 h^2 + c_2 h^4 + \ldots \tag{4.15}$$

mit c_i unabhängig von h. Die grundsätzliche Idee ist, wegen $\lim_{h\to 0} T(h) = \int_a^b f(x)dx$ dem exakten Integral möglichst nahe zu kommen, wobei Ergebnisse bereits durchgeführter Rechnungen mit größeren Schrittweiten h genutzt werden können. Mit einer Reihe von geeignet gewählten Abständen $h_0 = b{-}a, h_1 = h_0/n_1, \ldots, h_m = h_0/n_m$ werden nun die verschiedenen Trapezsummen
$T_{i0} = T(h_i),\ i = 1, \ldots, m$
wie oben beschrieben ermittelt. Man ermittelt nun rekursiv durch Interpolation das Polynom
$\tilde{T}_{mm}(h) := a_0 + a_1 h^2 + \ldots + a_n h^{2m}$ in h^2,
für welches gilt

$\tilde{T}_{mm}(h_i) = T_{mm}(h_i),\ i = 1, \ldots, m.$
Die gesuchte, verbesserte Näherung ermittelt man dann für $\tilde{T}_{mm}(0)$.

Man kann dann z.B. durch Halbierung der Schrittweiten h_i bei der Trapezregel folgende Rekursion anwenden

1. Schrittweitenwahl
$$h_k := \frac{b-a}{2^k}, \quad k = 0, 1, \ldots, m \tag{4.16}$$
2. Trapezregelauswertung
Berechne $T_{0,0} := T(h_0)$ und weiter
$$T_{k,0} := \frac{1}{2}T_{k-1,0} + h_k \left(f(a+h_k) + f(a+3h_k) + \ldots + f(b-3h_k) + f(b-h_k)\right) \tag{4.17}$$
3. Für $j = 1, 2, \ldots, m$ und $k = j, j+1, \ldots, m$
$$T_{k,j} := \frac{4^j T_{k,j-1} - T_{k-1,j-1}}{4^j - 1} \tag{4.18}$$

Für die Fehlerabschätzung gilt dann, nach längerem Nachweis

$$|T_{k,j} - I| \leq \frac{(b-a)^{2j+3} B_{j+1}}{4^{k-j}\, 2^{j(j+1)} (2j+2)!} \max_{x \in [a,b]} \left| f^{(2m+2)}(x) \right|, \tag{4.19}$$

falls $f \in \mathcal{C}^{2m+2}[a, b]$, mit B_i Bernoullizahlen.

4.3 Gaußintegration

Eine wesentliche Verbesserung der Genauigkeit erhält man mit einer auf Gauß zurückgehenden Methode. Man approximiert das zu integrierende Integral von Funktionen des Typs $\omega(x) \cdot f(x)$ mit stetigem $\omega(x) > 0$ für $x \in [a, b]$

$$I(f) := \int_a^b \omega(x) \cdot f(x) dx \approx \sum_{i=1}^{n} w_i \cdot f(x_i) =: \tilde{I}(f). \tag{4.20}$$

Trivialerweise kann $\omega(x) \equiv 1$ sein. In diesem, wie auch im allgemeinen Fall, kann man sich fragen, ob durch geschickte Wahl der w_i und x_i ein Polynom höheren Grades als n exakt integriert werden kann. Wir befreien uns also von der Festlegung äquidistanter Stützstellen mit dem Ziele, die gewonnenen $2n$ Freiheitsgrade zur Verbesserung der numerischen Integration zu nutzen. Formales Ziel ist, die Differenz $\tilde{I}(f) - I(f)$ für Polynome möglichst hohen Grades verschwinden zu lassen. Man kann nach längerem Beweisgang zeigen, dass

1. zu jedem $n = 1, 2, \ldots$ eindeutig bestimmte Zahlen $w_i, x_i, i = 1, \ldots, n$ gibt, so dass $\tilde{I}(f) = I(f)$ für alle Polynome p mit Grad $p \leq 2n - 1$.

2. für alle $i = 1, \ldots, n$ gilt $w_i > 0$ und $x_i \in (a, b)$.

Um eine Approximation an $I(f)$ durch Polynome wachsender Ordnung ohne Überschwingungen zu erhalten, werden geeignete, im Intervall beschränkte Polynome $p_j(x)$ gesucht, deren geeignete Linearkombination das Gewünschte liefern. Dies gelingt mittels geeigneter Familien orthonormaler Funktionen, deren zugehöriges Skalarprodukt auf das gegebene Integral aufbaut. Wir definieren das Skalarprodukt zweier Funktionen f, g bei gegebener Gewichtsfunktion $\omega(x)$

$$\langle f, g \rangle := \int_a^b \omega(x) f(x) g(x) dx \tag{4.21}$$

Ein Satz von orthonormalen Polynomen läßt sich nun mit dem folgenden Verfahren konstruieren.

$$p_{-1} \equiv 0 \tag{4.22}$$
$$p_0 \equiv 1 \tag{4.23}$$
$$p_{j+1} = (x - a_j) p_j(x) - b_j p_{j-1}(x), \quad j = 0, 1, \ldots \tag{4.24}$$

wobei

$$a_j = \frac{\langle x p_j, p_j \rangle}{\langle p_j, p_j \rangle}, \qquad j = 0, 1, \ldots \tag{4.25}$$
$$b_j = \frac{\langle p_j, p_j \rangle}{\langle p_{j-1}, p_{j-1} \rangle}, \qquad j = 0, 1, \ldots \tag{4.26}$$

Alle Polynome $p_j(x)$ haben im Intervall (a, b) n verschiedene Stützstellen (= Nullstellen). Ferner fallen die Stützstellen der verschiedenen Polynome nicht überein, sondern trennen einander bei einem Grad Unterschied p_{j-1}, p_j. Es kann nun gezeigt werden: Die N Nullstellen $x_i, i = 1, \ldots, N$ des zur Integration gewählten Polynoms $p_N(x)$ sind nun genau die optimalen Stützstellen, die in (4.20) die exakte Integration für alle Linearkombinationen der Polynome bis zum Grad $2N - 1$ gewährleisten. Methoden für die numerische Ermittlung von Nullstellen werden in einem folgenden Kapitel behandelt.

Desweiteren müssen noch die Gewichte w_i in (4.20) bestimmt werden, um das Geforderte zu leisten. Eine Möglichkeit ist durch die Lösung der Gleichung

$$\begin{pmatrix} p_0(x_1) & \ldots & p_0(x_N) \\ p_1(x_1) & \ldots & p_1(x_N) \\ \vdots & & \vdots \\ p_{N-1}(x_1) & \ldots & p_{N-1}(x_N) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} \int_a^b \omega(x) p_0(x) dx \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{4.27}$$

gegeben.

Die Wahl der Familie orthogonaler Polynome ist problemabhängig. So sind z.B. für die Lösung der Potentialgleichung auf der Sphäre die von Pol zu Pol

integrierenden Polynome vom Gauß-Legendre-Typ[1] erforderlich, nachdem der Integrationsbereich von Pol zu Pol nach Skalierung auf das Intervall $\sin[90S, 90N] = [-1, 1]$ transformiert wurde.

Wichtige Orthogonalsysteme für die Gaußquadratur lauten nun:

- Gauß-Legendre:
  $W(x) \equiv 1 \qquad [-1, 1]$
  $(j+1)P_{j+1} = (2j+1)xP_j - jP_{j-1}$
- Gauß-Tschebyscheff:
  $W(x) = 1/\sqrt{1-x^2} \qquad [-1, 1]$
  $T_{j+1} = 2xT_j - T_{j-1}$
- Gauß-Laguerre:
  $W(x) = x^\alpha \exp(-x) \qquad (0, \infty)$
  $(j+1)L^\alpha_{j+1} = (-x + 2j + \alpha + 1)L^\alpha_j - (j+\alpha)L^\alpha_{j-1}$
- Gauß-Hermite:
  $W(x) = \exp(-x^2) \qquad (-\infty, \infty)$
  $H_{j+1} = 2xH_j - 2jH_{j-1}$

Eingebettete Gaußregeln

Bei der schnellen Fouriertransformation wurde systematisch von der Eigenschaft der Exponentialfunktion Gebrauch gemacht, dass Stützstellenevaluationen an weiter distanzierten Stützstellen auch für nachfolgende Verfeinerungen genutzt werden können. Man bezeichnet die Stützstellen als *eingebettet*. Wie oben beschrieben, sind im Gegensatz dazu die Stützstellen/Nullstellen für die nach der Gaußintegration ermittelten Polynome disjunkt, und können somit nicht verwendet werden. Mittels der Gauß-Konrod-Quadratur können ausgehend von der n-Punkte-Gaußformel optimale $2n+1$-Punkte-Formel durch Ergänzung von $n+1$ Punkten zu gewinnen

$$\tilde{I}(f) := \sum_{i=1}^{n} w_i \cdot f(x_i) + \sum_{j=1}^{n+1} v_i \cdot f(y_j). \tag{4.28}$$

Dieses Verfahren integriert Polynome bis zum Grad $3n+1$ exakt. Allerdings arbeitet dieses Verfahren nicht für jedes n exakt.

4.4 Mehrdimensionale Integration

Die Gauß-Quadratur kann für 2 Dimensionen zur Gauß-Kubatur erweitert werden, wo entsprechente Stützstellen und Gewichte auf einer Fläche ermittelt werden müssen. Allgemein kann man bei niedrigdimensionalen Problemen

[1] Zusammen mit den im vorigen Kapitel behandelten und längs der Breitenkreisen integrierenden Exponentialfunktionen bilden sie nach summandenweisen Produkt die 2-dimensionalen Kugelflächenfunktionen (spherical harmonics) $Y^j_k = P_j(\sin(\varphi)) \exp(\frac{2\pi i k}{N})$.

durch Produktansatz das Integral

$$I := \int_a^b \int_c^d f(x,y)dxdy \approx \sum_{i=1}^{n} \sum_{j=1}^{n} w_i v_j f(x_i, y_j) \tag{4.29}$$

lösen. Allerdings wächst der Rechenaufwand mit der Anzahl der Dimensionen sehr schnell überproportional. Alternativ kann man Ansätze aus der Theorie der *Finiten Elemente* heranziehen. Bei sehr hohen Dimensionen werden Monte-Carlo-Verfahren angewandt (siehe Vorlesung MATHMET 2).

4.4.1 Neue Begriffe

Newton-Cotes-Formeln, Integration mit Intervallaufteilungen, Gaußintegration, Trapezregel, Simpsonregel, Romberg-Integration, Quadratur, Gauß-Konrod-Quadratur,

4.4.2 Fragen

- Wie kann man bei höherer Anzahl von Stützstellen vermeiden, Polynome höherer Ordnung heran zu ziehen?
- Welche zwei Vorteile liefert die Anwendung der Gau"integration, und welchen "Preis"muss man dafr akzeptieren?

Kapitel 5

Lineare Gleichungssysteme

Lineare Gleichungssystem können mittels Iterationen approximativ oder durch direkte Methoden bis auf Rundungsfehler exakt gelöst werden. In diesem Kapitel werden zunächst Verfahren der letztgenannten Methode behandelt, die in vorgegeben endlich vielen Schritten zur Lösung gelangen. Iterative Verfahren werden typischerweise bei sehr hoher Anzahl von Unbekannten angewandt, wie sie oft bei der numerischen Lösung partieller Differentialgleichungen auftreten. Dieser Bereich wird gegen Ende des Kapitels kurz behandelt.

Gegeben seien die quadratische Matrix $A \in \mathbb{R}^{n\times n}$ und der Vektor $b \in \mathbb{R}^n$, während Vektor $x \in \mathbb{R}^n$ die gesuchte Lösung ist.

$$Ax = b, \qquad \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}, . \tag{5.1}$$

5.1 Gauß-Elimination

Das lineare Gleichungsproblem (5.1) mit quadratischen, nichtsingulären Matrizen ist zu einem wesentlichen Teil gelöst, wenn man die Matrix A in eine untere Dreiecksmatrix L und obere Dreiecksmatrix R faktorisieren kann.

$$A = L \cdot R. \tag{5.2}$$

In diesem Falle hat man zwei gestaffelte Gleichungssysteme

$$L \cdot c = b \qquad \text{mit} \quad R \cdot x = c. \tag{5.3}$$

Gestaffelte Gleichungssysteme sind aber direkt lösbar. Hier ermittelt man zuerst den unbekannten Vektor c, danach die eigentlichen Unbekannten x. Im bereits faktorisierten Fall lautet der Algorithmus, falls die Diagonalelemente

von L,R, l_{ii},$r_{ii} \neq 0$ sind

$$c_i := \frac{1}{l_{ii}} \left(b_i - \sum_{k=1}^{i-1} l_{ik} c_k \right), \qquad i = 1, 2 \ldots, n,$$
$$x_i := \frac{1}{r_{ii}} \left(c_i - \sum_{k=i+1}^{n} r_{ik} x_k \right), \qquad i = n, n-1, \ldots, 1. \tag{5.4}$$

Nun läßt sich nicht jede reguläre Matrix LR-zerlegen, wie man an folgendem einfachen Beispiel rasch erkennt: Sei $A = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}\right)$, also $\det(A) = -1 \neq 0$. Eine LR-Zerlegung ergibt dann $a_{11} = l_{11} \cdot r_{11} = 0$. Da damit zumindest l_{11} oder $r_{11} = 0$ wäre, müßte A nicht-regulär, d.h. $\det(A) = 0$ sein, was im Widerspruch zur Regularität von A steht.

Das Problem läßt sich aber bekanntermaßen durch Zeilenvertauschung lösen, d.h. man multipliziert das Gleichungssystem (5.1) von links mit einer Permutationsmatrix P. Es gilt der

Satz 5.1 *Sei $A \in \mathbb{R}^{n \times n}$ eine reguläre Matrix. So gibt es immer eine Permutationsmatrix P, eine untere Dreiecksmatrix L und obere Dreiecksmatrix R, so dass gilt*

$$P \cdot A = L \cdot R. \tag{5.5}$$

Im oben genannten Beispiel löst $P = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ das Problem.

In der Praxis ermittelt und verwendet man nicht explizit die Matrizen P, L, R, sondern man wendet das Gaußsche Eliminationsverfahren an, welches, bei gleicher theoretischer Grundlage, effizienter ist. Aus Gründen der Anschaulichkeit schreibt man die Matrix

$$[A, b] = \left(\begin{array}{ccc|c} a_{11} & \ldots & a_{1n} & b_1 \\ \vdots & & \vdots & \vdots \\ \vdots & & \vdots & \vdots \\ a_{n1} & \ldots & a_{nn} & b_n \end{array}\right) \in \mathbb{R}^{n \times n+1}. \tag{5.6}$$

Nun werden von Spalte 1 bis n die Matrixelemente unter der Diagonalen eliminiert, also zunächst eine Matrix $[A', b']$ konstruiert, die nach Erledigung der 1. Spalte die folgende Form hat

$$[A, b]^{(1)} =: [A', b'] = \left(\begin{array}{cccc|c} a'_{11} & a'_{12} & \ldots & a'_{1n} & b'_1 \\ 0 & a'_{22} & & \vdots & b'_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a'_{n2} & \ldots & a'_{nn} & b'_n. \end{array}\right). \tag{5.7}$$

Der Formalismus lautet nun für jede Spalte $k = 1, \ldots, n$,

1. Spaltenpivotsuche: Ermittle ein *Pivotelement*[1] $|a_{ik}| = \max_m |a_{mk}| \neq 0$, $i, m = k, \ldots, n$. Falls nicht vorhanden, ist A singulär: Stop!

[1] pivot, *franz., engl.* Zapfen, Türangel

2. Bilde Matrix $\left[\bar{A}, \bar{b}\right]^{(k)}$ durch Vertauschen der Zeile i mit k.

3. Subtrahiere für $j = k+1, \ldots, n$ das $l_{jk} := \bar{a}_{jk}/\bar{a}_{kk}$-fache der k-ten Zeile von der j-ten Zeile. Mit Notation als Zeilenvektor also: $(a'_{j*}|b'_j)^{(k)} = (\bar{a}_{j*}|\bar{b}_j)^{(k)} - l_{jk} \cdot (\bar{a}_{k*}|\bar{b}_k)^{(k)}$, wobei für die ersten k Elemente $a'_{jl} = 0,\ l = 1, \ldots, k$.

Um ein möglichst gutartiges Verhalten zu erziehlen, wird also in der Regel als Pivotelement das betragmäßig größte Element unter der Diagonalen gesucht. Man kann dabei die Schritte 2 und 3 als Matrixmultiplikationen auffassen mit der Permutationsmatrix durch Vertauschen der Zeile i mit k der Einheitsmatrix

$$P^{(k)} := \begin{pmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & \mathbf{0} & & \\ & & 1 & & & & & & & \\ & & & 0 & \ldots & \ldots & 0 & 1 & & \\ & & & \vdots & 1 & & & 0 & & \\ & & & \vdots & & \ddots & & \vdots & & \\ & & & 0 & & & 1 & \vdots & & \\ & & & 1 & 0 & \ldots & \ldots & 0 & & \\ & & & & & & & & 1 & \\ & & \mathbf{0} & & & & & & \ddots & \\ & & & & & & & & & 1 \end{pmatrix}, \ , \tag{5.8}$$

mit der Inversen $(P^{(k)})^{-1} = P^{(k)}$, und einer unteren Dreiecksmatrix $G^{(k)}$ mit höchstens einer von der Einheitsmatrix verschiedenen Spalte (auch *Frobenius-matrix* genannt)

$$G^{(k)} := \begin{pmatrix} 1 & & & & & \\ & \ddots & & \mathbf{0} & & \\ & & 1 & & & \\ & & -l_{k+1,k} & \ddots & & \\ & \mathbf{0} & \vdots & \mathbf{0} & \ddots & \\ & & -l_{n,k} & & & 1 \end{pmatrix}, \ (G^{(k)})^{-1} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & \mathbf{0} & & \\ & & 1 & & & \\ & & l_{k+1,k} & \ddots & & \\ & \mathbf{0} & \vdots & \mathbf{0} & \ddots & \\ & & l_{n,k} & & & 1 \end{pmatrix}. \tag{5.9}$$

Die Annulierung der k-ten Subdiagonalelemente erreicht man also mittels

$$[A, b]^{(k)} = G^{(k)} P^{(k)} \, [A, b]^{(k-1)} \, . \tag{5.10}$$

Die vollständige obere Dreiecksmatrix ermittelt man daher mit

$$[R, c] = [A, b]^{(n-1)} = G^{(n-1)} P^{(n-1)} \ldots G^{(1)} P^{(1)} \, [A, b] \, . \tag{5.11}$$

Erledigt man alle Pivotsuchen und Zeilenaustauschungen vorab, so kann man die vollständige LR-Zerlegung mit $L := ((G^{(n-1)})^{-1} \ldots ((G^{(1)}))^{-1}$ schreiben, und man sieht sofort

$$LR = PA. \tag{5.12}$$

Das Verfahren kann durch eine zusätzliche Zeilenpivotwahl = Spaltenvertauschung erweitert werden, womit eine *vollständige* oder *totale Pivotsuche* erreicht wird. Dies entspricht der Multiplikation einer Permutationsmatrix Q von rechts, also

$$LR = AQ. \tag{5.13}$$

Bei der Zeilenpivotwahl werden damit aber auch die Positionen des Lösungsvektors x vertauscht, und müssen daher entsprechend registriert werden, um diese Vertauschung vor Ergebnissausgabe wieder rückgängig zu machen.

Da die Matrizen A oft sehr groß werden können, sind speichersparende Maßnahmen immer sinnvoll. Daher speichert man die Elemente von R und L an den entsprechenden Stellen von A durch Überschreiben. Die Diagonale von L, die durchgehend mit 1 besetzt ist, kann dabei gespart werden. Allerdings sind Buchhaltervektoren notwendig, die die Zeilen- und Spaltenvertauschungen speichern.

Dreieckszerlegungen sind von kubischer Komplexität: $\mathcal{O}(n^3)$; genauer, man benötigt $N^3/3$ Operationen. LR-Zerlegungen kann man daher auch effizient zur Berechnung der Determinante von A verwenden, verglichen mit dem in der Linearen Algebra gelehrten Entwicklungssatz von Laplace, welcher von der Ordnung $\mathcal{O}(n!)$ ist. Man erhält wegen $\det(P) = \pm 1,\ \det(L) = 1$

$$\det(PA) = \pm \det(A) = \det(R) = \prod_{i=1}^{n} r_{ii}. \tag{5.14}$$

Beispiel (aus Stoer)

$$\begin{pmatrix} 3 & 1 & 6 \\ 2 & 1 & 3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 7 \\ 4 \end{pmatrix}.$$

$$\left(\begin{array}{ccc|c} \underbrace{3} & 1 & 6 & 2 \\ 2 & 1 & 3 & 7 \\ 1 & 1 & 1 & 4 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 3 & 1 & 6 & 2 \\ \mathbf{2/3} & 1/3 & -1 & 17/3 \\ \mathbf{1/3} & \underbrace{2/3} & -1 & 10/3 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 3 & 1 & 6 & 2 \\ \mathbf{1/3} & 2/3 & -1 & 10/3 \\ \mathbf{2/3} & \mathbf{1/2} & -1/2 & 4 \end{array}\right),$$

wobei die Pivotelemente durch Unterklammerung und die Elemente von L durch Fettdruck markiert sind. Das gestaffelte Gleichungssystem lautet daher

$$\begin{pmatrix} 3 & 1 & 6 \\ 0 & 2/3 & -1 \\ 0 & 0 & -1/2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 10/3 \\ 4 \end{pmatrix}.$$

Als Lösung ermittelt man
$x_3 = -8, \quad , x_2 = 3/2(10/3 + x_3) = -7, \quad x_1 = 1/3(2 - x_2 - 6x_3) = 19.$

Das Gleichungssystem $PA = LR$ lautet somit

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 1 & 6 \\ 2 & 1 & 3 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ 2/3 & 1/2 & 1 \end{pmatrix} \begin{pmatrix} 3 & 1 & 6 \\ 0 & 2/3 & -1 \\ 0 & 0 & -1/2 \end{pmatrix}.$$

5.2 Choleskyverfahren

Das Problem (5.1) soll nun unter der Annahme des Spezialfalles gelöst werden, dass A eine symmetrische und positiv-definite Matrix ist. In geophysikalischen und meteorologischen Anwendungen tritt dieser Fall besonders häufig auf, wenn A Kovarianzmatrizen bezeichnen, die grundsätzlich diese Eigenschaften besitzen[2].

Definition 5.1 *Eine reelle (komplexe) $n \times n$-Matrix A heißt* positiv definit, *falls gilt*

a) $A = A^T$, $(A = A^H)$, *also A ist eine symmetrische (Hermitesche) Matrix,*

b) $x^T Ax > 0$ $(x^H Ax > 0)$ $\forall x \in \mathbb{R}^n$ $(x \in \mathbb{C}^n)$, $x \neq 0$

$A = A^T$ heißt positiv semidefinit, *falls $x^T Ax \geq 0$ gilt. (Analog für Hermitesche Matrizen.)*

Es gelten nun folgende zwei entscheidende Sätze

Satz 5.2 *Matrix $A \in \mathbb{R}^{n \times n}$ sei symmetrisch und positiv definit. Dann gilt:*

a) $a_{ik}^2 < a_{ii} \cdot a_{kk}$ $\forall i \neq k$.

b) *Das absolut größte Element der Matrix liegt auf der Diagonalen.*

Zum Beweis siehe Stoer I.

Satz 5.3 *Matrix $A \in \mathbb{R}^{n \times n}$ sei symmetrisch und positiv definit. Dann gibt es genau eine linke untere Dreiecksmatrix L mit positiven Diagonalelementen, so dass*

$$LL^T = A \tag{5.15}$$

Zum Beweis siehe Stoer I.

Man kann zeigen, dass der folgende numerisch stabile Algorithmus zur Bestimmung von L gilt, falls $|a_{11}| >$ eps:

1. Setze $l_{11} := \sqrt{a_{11}}$.
2. Für $k = 2, \ldots, n$ berechne
   $l_{k1} := a_{k1}/l_{11}$
3. Für $i = 2, \ldots, n$:
   setze $s := a_{ii} - \sum_{m=1}^{i-1} l_{im}^2$.
   Falls $s >$ eps
   $l_{ii} := \sqrt{(}s)$, sonst Abbruch.
   Für $k = i + 1, \ldots, n$
   $l_{ki} := \frac{1}{l_{ii}} \left(a_{ki} - \sum_{m=1}^{i-1} l_{im} l_{km} \right)$.

[2]Falls eine Kovarianzmatrix nur positiv semidefinit ist, ist das zu lösende System reduzierbar, bis die Bedingung erfüllt ist.

Die Komplexität des Verfahrens besteht aus $n^3/6$ Operationen, also halb so groß wie bei der Gaußelimination, sowie zuzüglich Berechnung von n Quadratwurzeln. Man kann die Zerlegung auch wurzelfrei aufteilen mit $A = LDL^T$, wobei die Diagonalelemente von L : $l_{ii} = 1$ und Diagonalmatrix D die mit s berechneten Werte aufnimmt.

5.3 Fehlerabschätzung

Wie bekannt dürfen numerische Lösungen $\tilde{x}$ des Gleichungssystems $Ax = b$ in der Regel nur als Näherungen der wahren Lösung x betrachtet werden. Zur Beurteilung der Lösungsqualität und des Fehlers $\Delta x = x - \tilde{x}$ wird ein skalares Maß eingeführt, d.h. eine *Norm* $\|\circ\|$ im $\mathbb{R}^n$ oder $\mathbb{C}^n$ mit (hier im allgemeineren komplexen Fall)

$$\|\circ\| : \mathbb{C}^n \to \mathbb{R}, \tag{5.16}$$

die also als reelles Maß eine Größe eines Fehlervektors bezeichnen soll.

Definition 5.2 *Eine Abbildung* $\|\circ\| : \mathbb{C}^n \to \mathbb{R}$ *ist eine* Norm*, wenn folgende Eigenschaften erfüllt sind*

a) $\|x\| > 0 \ \forall x \in \mathbb{C}^n, \ x \neq 0,$

b) $\|\alpha x\| = |\alpha| \, \|x\| \ \ \forall x \in \mathbb{C}^n, \ \alpha \in \mathbb{C}$ *(Homogenität),*

c) $\|x + y\| \leq \|x\| + \|y\| \ \ \forall x, y \in \mathbb{C}^n$ *(Dreiecksungleichung).*

Beispiel 5.1 *Beispiele für Normen sind*

a) *Euklidische Norm:* $\|x\|_2 := \sqrt{x^H x} = \sqrt{\sum_{i=1}^n |x_i|^2}$,

b) *Maximumnorm:* $\|x\|_\infty := \max_i |x_i|$.

Von einer Matrixnorm wird für Matrizen $A \in \mathbb{C}^{m \times n}$ analog verlangt

a) $\|A\| > 0 \ \forall A \neq 0, \ A \in \mathbb{C}^{m \times n},$

b) $\|\alpha A\| = |\alpha| \, \|A\|$ (Homogenität),

c) $\|A + B\| \leq \|A\| + \|B\|$ (Dreiecksungleichung).

Zwei weitere Begriffe:

Definition 5.3 *Eine Matrixnorm* $\|\circ\|$ *heißt*

a) *mit den Vektornormen* $\|\circ\|_\alpha$ *auf* $\mathbb{C}^n$ *und* $\|\circ\|_\beta$ *auf* $\mathbb{C}^m$ verträglich*, wenn* $\|Ax\|_\beta \leq \|A\| \, \|x\|_\alpha \ \ \forall x \in \mathbb{C}^n$,

b) *im Falle quadratischer Matrizen* $A \in \mathbb{C}^{n \times n}$ submultiplikativ*, wenn* $\|AB\| \leq \|A\| \, \|B\| \ \ \forall A, B \in \mathbb{C}^{n \times n}$.

Beispiele für Matrixnormen sind

Beispiel 5.2 a) *Schur–Norm:* $\|A\|_F = \sqrt{\sum_{i,k=1}^{n} |a_{ik}|^2}$ *(submultiplikativ),*

b) *Maximumnorm:* $\|A\| = \max_{ik} |a_{ik}|$ *(nicht submultiplikativ),*

c) *Zeilensummennorm* $\|A\|_\infty = \max_i \sum_{k=1}^{n} |a_{ik}|$ *(submultiplikativ).*

Für theoretische Betrachtungen ist noch die *Grenzennorm* interessant

$$\mathrm{lub}(A) := \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \tag{5.17}$$

Definition 5.4 *Unter der* Kondition *einer regulären Matrix A zu einer Matrixnorm $\|\circ\|$ versteht man das Produkt* $\mathrm{cond}(A) := \|A\| \, \|A^{-1}\|$.

Die Kondition von A ist ein Maß für die Empfindlichkeit der Lösung x auf Änderungen von b.

Wir nehmen für das Folgende an, dass $\|x\|$ eine beliebige Vektornorm, aber $\|A\|$ eine damit verträgliche und submultiplikative Matrixnorm ist. Wir betrachten zunächst den Einfluß von Änderungen von b bei der Lösung des Gleichungssystems $Ax = b$. Es gelte

$$A(x + \Delta x) = b + \Delta b. \tag{5.18}$$

Als Folge der Linearität und Regularität findet man $\Delta x = A^{-1}\Delta b$ mit der Abschätzung

$$\|\Delta x\| \leq \left\|A^{-1}\right\| \|\Delta b\| . \tag{5.19}$$

Im Kontext der numerisch nur genäherten Lösung $\tilde{x}$ kann man das *Residuum* r der Lösung als künstliche Abänderung von b infolge Lösungsfehlers Δx auffassen:

$$r(\tilde{x}) = b - A\tilde{x} = A(x - \tilde{x}) = A\Delta x = \Delta b. \tag{5.20}$$

Weil $A\tilde{x} = b - r(\tilde{x})$ exakt gilt, findet man

$$\|\Delta x\| \leq \left\|A^{-1}\right\| \|r(\tilde{x})\| . \tag{5.21}$$

Da ferner infolge Verträglichkeit gilt $\|A\| \, \|x\| \geq \|b\|$, folgt für die relative Änderung

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\| \left\|A^{-1}\right\| \frac{\|\Delta b\|}{\|b\|} = \mathrm{cond}(A) \frac{\|\Delta b\|}{\|b\|}. \tag{5.22}$$

Über das Problem der Stabilität numerischer Lösungen linearer Gleichungen kann man auf das analoge Problem bei noch zu behandelnden Ausgleichsrechnungen hinweisen, in denen b oftmals Messungen repräsentiert. Bei großer Kondition $\mathrm{cond}(A)$ hängt hier das Ergebnis kritisch von den Messfehlern ab! Eine Lösung dieses Problems steht im Zentrum der Ausgleichsrechnung.

Es besteht auch die Möglichkeit, den Einfluß von Änderungen an der Matrix A auf die Lösung x zu untersuchen.

5.3.1 Skalierungen

Bei der Pivotsuche zur Elimination wie in (5.7) wurde das absolut größte Subdiagonalelement $|a_{ik}| = \max_m |a_{mk}| \neq 0, \quad i, m = k, \ldots, n$ der k-ten Spalte ermittelt. Dies dient, wie oben dargelegt wurde, allgemein der Minimierung von Rundungsfehlern. Im strengen Sinne gewährleistet dies nur, dass bei regulären Matrizen kein vorzeitiger Abbruch des Algorithmus erfolgt. Dass die Rundungsfehlerproblematik gleichwohl weiterbestehen *kann*, sieht man an folgendem (Stör entnommenem) Beispiel. Gegeben sei die Gleichung

$$\begin{pmatrix} 0.005 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix} \tag{5.23}$$

welche die exakte Lösung $x = 5000/9950 = 0.503\ldots, y = 4950/9950 = 0.497\ldots$ besitzt. Wählt man ohne Pivotsuche $a_{11} = 0.005$ als Pivotelement,

$$\begin{pmatrix} 0.005 & 1 \\ 0 & -200 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.5 \\ -99 \end{pmatrix}, \tag{5.24}$$

so ergibt eine 2-stellige Gleitpunktrechnung das Ergebnis $y = 0.5, \ x = 0$. Wählt man dagegen $a_{21} = 1$ als Pivotelement, so erhält man mit ebenfalls 2-stelligen Rechnungen

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} \tag{5.25}$$

das Ergebnis $y = 0.50, \ x = 0.50$. Erwartungsgemäß ist die Genauigkeit deutlich größer.

Nun kann man aber das gleiche Geichungssystem (5.23) umskalieren oder umskaliert formuliert haben, etwa duch andere physikalische Einheiten. Dies ändert nichts an der Lösung. Formal geschieht dies durch Multiplikation von (5.23) durch eine Diagonalmatrix D von links $DAx =: \tilde{A}x = Db =: \tilde{b}$.

Sei hier $D := \begin{pmatrix} 200 & 0 \\ 0 & 1 \end{pmatrix}$. Zwar hat man nun mit

$$\begin{pmatrix} 1 & 200 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 100 \\ 1 \end{pmatrix} \tag{5.26}$$

ein $a_{11} = 1$ als Pivotelement, als Lösung erhält man jedoch wieder wie im Fall (5.24) das Ergebnis $y = 0.5, \ x = 0$! Das Kriterium $|a_{ik}| = \max_m |a_{mk}| \neq 0, \ i, m = k, \ldots, n$ kann also nur von heuristischer Natur sein. Gleichwohl ist die Nutzung von Skalierungsmatrizen sinnvoll, insbesondere wenn auch Spalten umnormiert werden. Im allgemeinen Fall lautet dann die Gleichung

$$D_1 A D_2 D_2^{-1} x = D_1 b = \hat{A} y = \tilde{b}, \tag{5.27}$$

wobei $\hat{A} := D_1 A D_2, \ y := D_2^{-1} x$. Insgesamt muss die Beobachtung festgehalten werden, dass Skalierung und Pivotsuchstrategie nicht unabhängig voneinander

angewandt werden sollen. Als praktische Regel gilt:
Die Wahl von D_1, D_2 soll so sein, dass für alle Zeilen und Spalten $i, l = 1, 2, \ldots, n$ von $\hat{A}$ angenähert gilt

$$\sum_{k=1}^{n} |\hat{a}_{ik}| \approx \sum_{j=1}^{n} |\hat{a}_{jl}|. \tag{5.28}$$

Matrizen mit dieser Eigenschaft heißen *equilibriert*. Allerdings ist eine Bestimmung der Matrizen D_1, D_2, die die Bedingung (5.28) erfüllen schwierig. Wählt man $D_2 = I, D_1 = \text{diag}(s_1, \ldots, s_n)$ mit

$$s_i := 1/\sum_{k=1}^{n} |\hat{a}_{ik}|, \tag{5.29}$$

so erreicht man für $\hat{A} := D_1 A D_2$

$$\sum_{k=1}^{n} |\hat{a}_{ik}| = 1, \ i = 1, \ldots, n. \tag{5.30}$$

In der Praxis wird dann nicht die Matrix $\hat{A}$ explizit aufgestellt, sondern wird für den Schritt $A^{(j-1)} \to A^{(j)}$ eine mit s_r gewichtete Pivotsuche durchgeführt, wobei Zeile $r \geq j$ so bestimmt wird, dass

$$|a_{rj}^{(j-1)}| \ s_r = \max_{i \geq j} |a_{ij}^{(j-1)}| \ s_i \neq 0. \tag{5.31}$$

5.4 Die Dreieckszerlegung nach Householder

Obwohl durch die Pivotsuche Rundungsfehler reduziert werden können, werden bei jeder Matrizenmultiplikation in (5.11) neue Fehler eingeführt. Die Empfindlichkeit der Lösung x von den Zwischenmatrizen $[A, b]^{(k)}$ ist durch $\text{cond}(A^{(k)}) := \left\|A^{(k)}\right\| \left\|(A^{(k)})^{-1}\right\|$ gegeben, denn der Rundungsfehler $\epsilon^{(k)}$, den man beim Übergang zum System $[A, b]^{(k)} = G^{(k)} P^{(k)} [A, b]^{(k-1)}$ begeht, wird dabei um den Faktor $\text{cond}(A^{(k)})$ verstärkt. Dabei gilt mit (5.22)

$$\frac{\|\Delta x\|}{\|x\|} \leq \sum_{k=0}^{n-1} \epsilon^k \, \text{cond}(A^{(k)}). \tag{5.32}$$

Indizieren wir die Ausgangsgleichung mit 0, so ist schon bei einem Schritt $[A, b]^{(k)}$ mit $\text{cond}(A^{(k)}) >> \text{cond}(A^{(0)})$ die Gutartigkeit des Lösungsverfahrens verloren.

Ziel ist es nun, als Ersatz für die Gaußzerlegung mittels $G^{(k)} P^{(k)}$ ein Konstruktionsverfahren für Transformationsmatrizen $Q^{(k)}$ zu finden, welches die Konditionszahl der $\text{cond}(A^{(k)})$ nicht weiter wachsen läßt. Dies gelingt mit der

Householdertransformation, wobei die Euklidische Norm und ihre verträgliche Grenzennorm

$$\mathrm{lub}(A) = \max_{x \neq 0} \sqrt{\frac{x^T A^T A x}{x^T x}} \tag{5.33}$$

herangezogen werden. Haben wir etwa eine unitäre Matrix U, also $U^T U = I$, so bleibt bei Multiplikation mit A wegen

$$\begin{aligned} \mathrm{lub}(A) = \quad \mathrm{lub}(U^T U A) &\leq \quad \mathrm{lub}(U^T)\,\mathrm{lub}(UA) = \quad \mathrm{lub}(UA) \\ &\leq \quad \mathrm{lub}(U)\,\mathrm{lub}(A) = \quad \mathrm{lub}(A) \end{aligned} \tag{5.34}$$

die Matrixnorm $\mathrm{lub}(A)$ erhalten.

Die Frage ist, ob wir mittels dieser Transformation Q zu einer Zerlegung

$$A = QR \tag{5.35}$$

gelangen, die gleichzeitig unserem Ziel entsprechend, die Lösung der Ausgangsgleichung $Ax = b$ auch noch erleichtert. In der Faktorisierung (5.35) können wieder die zeilenweisen Einträge einer Spalte von R als die Koeffizienten gedeutet werden, mit denen die Spalte i von A durch Linearkombinationen aller Spalten von Q dargestellt werden. Die erwünschte Erleichterung erhält man mittels Abbildung der Teilspalten von A auf "kanonische" Achsenvektoren (=Einheitsvektoren). Lineare Abbildungen, die die Matrizennorm invariant lassen, sind Drehungen und Spiegelungen. Die Idee der Householdertransformation ist es, Teilspalten von A so an einer Hyperebene W zu spiegeln, dass R eine obere Dreiecksmatrix ist, wobei die Transformationsmatrix ebenfalls leicht zu invertieren ist.

Sei w der Vektor der Länge 1, der senkrecht auf W steht. Eine Spiegelung an W erreichen wir mit der Transformationsmatrix

$$Q_w = I - 2ww^T, \qquad \text{wobei} \quad w^T w = 1,\ w \in \mathbb{C}^n. \tag{5.36}$$

Folgende Eigenschaften zeichnen diese Matrix Q aus: sie ist

- symmetrisch

$$Q^T = I^T - 2(ww^T)^T = I - 2ww^T = Q \tag{5.37}$$

- orthonormal

$$Q^T Q = Q^2 = (I - 2ww^T)(I - 2ww^T) = I - 4ww^T + 4ww^T ww^T = I, \tag{5.38}$$

- und damit also auch involutorisch, also selbstinvers $Q^2 = I$.

Der Spiegelungsvektor w ist nun so zu bestimmen, dass der erste Spaltenvektor a von A auf ein Vielfaches k des ersten Achsenvektors e längeninvariant gespiegelt wird:

$$Q_w a = a - 2(w^T a)w =: ke. \tag{5.39}$$

Wegen der Orthonormalität gilt $k = \pm \|a\|$. Damit erhält man als normierten Spiegelungsvektor

$$w = \frac{a + ke}{\|a + ke\|}, \tag{5.40}$$

wobei man aus Stabilitätsgründen das Vorzeichen so wählt, dass der Nenner am größten ist. Dies bestätigt die Probe

$$\begin{aligned} Q_w a = a - 2(w^T a)w &= a - 2\frac{(a+ke)(\|a\|^2 + kae)}{\|a\|^2 + 2kae + k^2\|e\|^2} \\ &= a - 2(a+ke)\frac{(\|a\|^2 + kae)}{\|a\|^2 + 2kae + k^2\|a\|^2} \\ &= -ke \end{aligned}$$

Wir wählen nun den ersten kanonischen Einheitsvektor $e_1 := e$. Spaltenvektor a wird nun zerlegt in seine erste Komponente und die restlichen Komponenten $a =: (a_1, \bar{a})^T$, also $\|a\| = \sqrt{|a_1|^2 + \|\bar{a}\|^2}$, wobei nun das Vorzeichen von k durch jenes von a_1 bestimmt wird. Der unnormierte Spiegelungsvektor u, also die Richtung von w ist dann

$$u := a + \text{sign}(a_1)\|a\| e_1 = (\text{sign}(a_1)(|a_1| + \|a\|), \bar{a})^T. \tag{5.41}$$

Die zugehörige Normierung errechnet man zu

$$\|a + ke_1\| = \sqrt{2\|a\|(\|a\| + |a_1|)} =: 1/\beta \tag{5.42}$$

Die Householdertransformation erfolgt nun in der folgenden Weise: Zu jeder, auch singulärer Matrix $A \in \mathbb{R}^{n\times n}$ existiert nun eine Zerlegung

$$A = QR \tag{5.43}$$

mit einer orthogonalen Matrix Q und einer rechten oberen Dreiecksmatrix R. Bei dieser wird die Matrix $A^{(1)} := A$ durch Multiplikation mit Matrizen $Q_j^T, j = 1, ..., n-1$ in einer Sequenz von $n-1$ Schritten in eine Matrix $R := A^{(n)}$ umgewandelt. Wir setzen hierzu Q_j^T, $j = 1, ..., n-1$

$$Q_j^T := \begin{pmatrix} I_{j-1} & 0 \\ 0 & \widehat{Q_j^T} \end{pmatrix} \quad \text{mit} \quad \widehat{Q_j^T} \in \mathbb{R}^{(n+1-j)\times(n+1-j)} \Longrightarrow Q_j^T \in \mathbb{R}^{n\times n}. \tag{5.44}$$

$$\widehat{Q_j^T} := I_{n+1-j} - \beta_j u_j u_j^T, \quad \text{mit} \quad \beta_j := \frac{1}{\|a^{(j)}\|_2 \cdot (|a_1^{(j)}| + \|a^{(j)}\|_2)}, \tag{5.45}$$

$$u_j := \begin{pmatrix} \text{sgn}(a_1^{(j)}) \cdot (|a_1^{(j)}| + \|a^{(j)}\|_2) \\ a_2^{(j)} \\ \vdots \\ a_{n+1-j}^{(j)} \end{pmatrix} \quad a^{(j)} := \begin{pmatrix} a_{j,j}^{(j)} \\ \vdots \\ a_{n,j}^{(j)} \end{pmatrix}. \tag{5.46}$$

Dann fährt man fort mit

$$\begin{aligned} A^{(2)} &= Q_1^T A^{(1)} \\ A^{(3)} &= Q_2^T C^{(2)} = Q_2^T Q_1^T A^{(1)} \\ &\vdots \quad\quad \vdots \\ \underbrace{A^{(n)}}_{R} &= Q_{n-1}^T A^{(n-1)} = ... = \underbrace{Q_{n-1}^T Q_{n-2}^T \cdot ... \cdot Q_2^T Q_1^T}_{Q^T} \underbrace{A^{(1)}}_{A} \end{aligned}$$

Komplexität Es kann gezeigt werden, dass die Anzahl der Operationen des Verfahrens für $A \in \mathbb{C}^{n\times n}$ von der Ordnung $\frac{4}{3}n^3$ beträgt. Handelt es sich um eine $m \times n$-Matrix, mit $m >> n$, so ist die Komplexität $2n^2 \cdot m$.

5.5 Große lineare Gleichungssysteme

5.5.1 Allgemeine Iterationsverfahren

Große lineare Gleichungssysteme (etwa $\mathcal{O}(n) > 10^7$ bei meteorologischen Modellen) entstehen in der Regel aus der Diskretisierung partieller Differntialgleichungen. Die hierbei aufgestellten Matrizen sind üblicherweise *dünn besetzt* (engl. *sparse*), d.h. die Anzahl der Elemente $\neq 0$ wächst etwa linear mit der Anzahl der Zeilen/Spalten, und nicht quadratisch. Oft liegt eine Bandstruktur der Matrizen vor, wenngleich die Bandbreite erheblich sein kann. Auch innerhalb des Bandes sind oft viele Matrizeneinträge $= 0$. Die Verwendung bisher behandelter Verfahren verbietet sich aus 2 Gründen:

- Die enstehenden "Zwischenrechnungsmatrizen" zerstören die dünne Besetzung und die dadurch entstehende hohe Anzahl der Operationen ist nicht mehr effizient behandelbar.
- Der benötigte Speicherplatz steht (oft im Kontext mit weiteren Rechnungen) nicht zur Verfügung.

DerLösung großer linearer Gleichungssysteme liegt zumeist ein Iterationsverfahren zu Grunde, bei dem, ausgehend von einem Startvektor $x^{(0)}$, eine Folge von Vektoren

$$x^{(0)} \to x^{(1)} \to x^{(2)} \to x^{(3)} \ldots \tag{5.47}$$

erzeugt wird, die gegen die Lösung x konvergiert.

Wir beginnen wieder mit dem Gleichungssystem (5.1), dessen exakte Lösung im nichtsingulären Fall formal $x = A^{-1}b$ lautet. Mit Blick auf das allgemeine Iterationsschema

$$x^{(i+1)} = \Phi(x^{(i)}), \quad i = 0, 1, \ldots \tag{5.48}$$

wählt man eine geeignete nichtsingulären Matrix B zu der Umformung von (5.1) nach

$$Bx + (A - B)x = b, \tag{5.49}$$

um die Iterationsgleichung

$$Bx^{(i+1)} + (A - B)x^{(i)} = b \tag{5.50}$$

zu konstruieren. Explizit lautet nun die Iterationsvorschrift

$$x^{(i+1)} = x^{(i)} - B^{-1}(Ax^{(i)} - b) = (I - B^{-1}A)x^{(i)} + B^{-1}b. \tag{5.51}$$

Die Leistungsfähigkeit des Verfahrens (5.51) hängt davon ab, ob

- das Gleichungssystem (5.50) leicht nach (5.51) auflösbar ist, also die Inverse von B leicht zu ermitteln ist, und
- zur schnellen Konvergenz die Eigenwerte von $I - B^{-1}A$ möglichst kleine Beträge haben.

Von Bedeutung sind die folgenden Beispiele: Zunächst wählen wir als Zerlegung von A

$$A = D - E - F, \tag{5.52}$$

wobei

$$D := \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ & & \ddots \\ 0 & & a_{nn} \end{pmatrix}, \quad E := -\begin{pmatrix} 0 & & & 0 \\ a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \dots & a_{n,n-1} & 0 \end{pmatrix} \tag{5.53}$$

$$F := -\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & a_{n-1,n} \\ 0 & \dots & \dots & 0 \end{pmatrix}. \tag{5.54}$$

Ferner sei

$$L := D^{-1}E, \qquad U := D^{-1}F, \qquad J := L + U, \qquad H := (I - L)^{-1}U. \tag{5.55}$$

Man erhält dann folgende Verfahren:

1. Gesamtschritt- oder Jacobi-Verfahren
   Setze
   $$B := D, \Rightarrow I - B^{-1}A = B^{-1}(E + F) = J. \tag{5.56}$$
   Als Iterationsvorschrift erhält man somit für $a_{jj} \neq 0$
   $$x_j^{(i+1)} = \left(b_j - \sum_{k \neq j} a_{jk} x_k^{(i)} \right) / a_{jj}, \quad j = 1, 2, \dots, n, \quad i = 0, 1, \dots \tag{5.57}$$

2. Einzelschritt- oder Gauß-Seidel-Verfahren
   Man setzt

$$B := D - E, \Rightarrow \tag{5.58}$$

$$\begin{aligned} I - B^{-1}A &= I - (D-E)^{-1}(D-E-F) = (D-E)^{-1}F \\ &= (D - D\underbrace{D^{-1}E}_{L})^{-1}F = (I-L)^{-1}\underbrace{D^{-1}F}_{U} = H, \end{aligned}$$

   so dass man für (5.50) erhält

$$\sum_{k<j} a_{jk}x_k^{(i+1)} + a_{jj}x_j^{(i+1)} + \sum_{k>j} a_{jk}x_k^{(i)} = b_j, \quad j = 1,2,\ldots,n, \quad i = 0,1,\ldots \tag{5.59}$$

3. Das Iterationsverfahren kann man auch nutzen, um die angesammelten Fehler in Falle von Gaußelimination zu entfernen. Dabei ist dessen Ergebnis $x^{(0)}$ bereits eine gute Näherung. Damit gilt auch die angenäherte Dreieckszerlegung $B = \bar{L} \cdot \bar{R} \approx A$, wobei diese Zerlegung uns eine gute, leicht zu invertierende Matrix$B$ liefert.

   Für (5.50) erhält man dann

$$B(x^{(i+1)} - x^{(i)}) = r^{(i)} := b - Ax^{(i)}. \tag{5.60}$$

   Mit $u^{(i)} := \bar{R}^{-1}\bar{L}^{-1}r^{(i)}$ erhält man

$$x^{(i+1)} = x^{(i)} + u^{(i)}, \tag{5.61}$$

   wobei $u^{(i)}$ mittels

$$\bar{L}z = r^{(i)}, \qquad \bar{R}u^{(i)} = z \tag{5.62}$$

   in bekannter Weise berechnet werden kann. In der Regel genügen wegen der guten Anfangsnäherung $x^{(0)}$ nur sehr wenige Iterationen. Allerdings erhält man aber aus dem gleichen Grunde sehr kleine, und durch Auslöschung in ihrer Genauigkeit gefährdete Residuen $r^{(i)} := b - Ax^{(i)}$. Diese berechnet man daher zweckmäßigerweise mit doppelter Genauigkeit.

Konvergenz Die Frage der Konvergenz ist bisher offen gelassen worden. Hierzu definieren wir den Spektralradius einer Matrix.

Definition 5.5 *(Spektralradius) Sind λ_i die Eigenwerte von A, so bezeichnet man mit*

$$\rho(A) := \max_{1 \le i \le n} |\lambda_i| \tag{5.63}$$

den Spektralradius von A.

Es gilt der folgende Satz

Satz 5.4 *Der Algorithmus (5.50) ist genau dann konvergent, wenn*

$$\rho(I - B^{-1}A) < 1. \tag{5.64}$$

Dabei ist die Bedingung

$$\mathrm{lub}(I - B^{-1}A) < 1 \tag{5.65}$$

hinreichend.

Dieses Ergebnis stellt die Verbindung zum Fixpunktsatz der Analysis her, setzt aber zur Anwendung die aufwändige Berechnung der Eigenwerte oder Grenzennorm voraus. Ein einfacher Nachweis der Konvergenz ist aber möglich, denn es gilt:

Satz 5.5 *1. (Starkes Zeilensummenkriterium) Das Gesamtschritt- und Einzelschrittverfahren ist konvergent für alle Matrizen A mit*

$$|a_{ii}| > \sum_{k \neq i} |a_{ik}| \quad i = 1, 2, \ldots, n. \tag{5.66}$$

2. (Starkes Spaltensummenkriterium) Das Gesamtschritt- und Einzelschrittverfahren ist konvergent für alle Matrizen A mit

$$|a_{ii}| > \sum_{i \neq k} |a_{ik}| \quad k = 1, 2, \ldots, n. \tag{5.67}$$

5.5.2 Das SOR-Verfahren

Obwohl schärfere Voraussetzungen an die Matrix A erfüllt werden müssen, ist in der Praxis jedoch das Succesive Over-Relaxation-Relaxationsverfahren (SOR-Verfahren) bedeutender als das Jacobi- oder Gauß-Seidelverfahren, da in vielen typischen Anwendungen diese Voraussetzungen erfüllt werden. Die Grundidee ist, die Matrix B derart von einem skalaren Parameter ω abhängig zu machen, dass der Spektralradius $\rho(I - B^{-1}(\omega)A)$ möglichst klein ist. Relaxationsverfahren verwenden folgenden Ansatz

$$B(\omega) := \frac{1}{\omega} D(I - \omega L), \tag{5.68}$$

also im Ansatz das Gauß-Seidelverfahren für $\omega = 1$. Wir haben dann

$$\begin{aligned} B(\omega) - A =& \frac{1}{\omega} D(I - \omega \underbrace{D^{-1}E}_{L} - \omega I + \omega D^{-1}E + \omega \underbrace{D^{-1}F}_{U}) \\ =& \frac{1}{\omega} D((1-\omega)I + \omega U). \end{aligned} \tag{5.69}$$

Damit läßt sich nun für den Spektralradius der Matrix mit (5.68)

$$H(\omega) := (I - B^{-1}(\omega)A) = (I - \omega L)^{-1}\left((1-\omega)I + \omega U\right) \tag{5.70}$$

zeigen, dass für beliebige Matrizen gilt

$$\rho(H(\omega)) \geq |\omega - 1|\,, \tag{5.71}$$

sowie für positiv definite Matrizen A

$$\rho(H(\omega)) \leq 1 \quad \forall\, 0 < \omega < 2. \tag{5.72}$$

Die Angabe eines optimalen ω_b ist allgemein nicht möglich. Für eine bestimmte Klasse von Matrizen kann man jedoch genauere Angaben machen. Matrizen, bei deren Zerlegung die Eigenwerte der Matrizen

$$J(\alpha) = \alpha L + \alpha^{-1} U, \quad \alpha \neq 0 \tag{5.73}$$

von α unabhängig sind, heißen *konsistent geordnet*. Es gilt nun der Satz

Satz 5.6 *(Young, Varga): Sei A konsistent geordnet, die Eigenwerte von $J \in \mathbb{R}^{n\times n}$ und $\rho(J) < 1$. Dann gilt*

$$\omega_b = \frac{1}{1 - \sqrt{1 - \rho(J)^2}} \tag{5.74}$$

und

$$\rho(H(\omega_b)) = \omega_b - 1 = \left(\frac{\rho(J)}{1 - \sqrt{1 - \rho(J)^2}} \right)^2. \tag{5.75}$$

Dabei gilt für ω

$$\rho(H(\omega)) = \begin{cases} \omega - 1 & \textit{für} \quad \omega_b \leq \omega \leq 2 \\ 1 - \omega + 1/2\omega^2\rho(J)^2 + \omega\rho(J)\sqrt{1 - \omega + 1/4\omega^2\rho(J)^2} & \text{für} \quad 0 \leq \omega \leq \omega_b \end{cases} \tag{5.76}$$

Wegen des Verlaufes der Kurve sollte man einen eher etwas zu großen als zu kleinen Parameter wählen.

5.6 Mehrgitterverfahren

Für spezielle Anwendungen gibt es Löser, die wesentlich leistungsfähiger sind. Gleichungssysteme, die bei der Lösung elliptischer partieller Differentialgleichungen, die wir im letzten Kapitel kennen lernen werden, auftreten, sind ein häufig genanntes Beispiel. Da Potentialgleichungen zu diesem Typ gehören, treten sie in geophysikalischen Anwendungen, aber auch bei impliziten Lösern als Teilaufgaben meteorologischer Anwendungen immer wieder auf. Mehrgitterverfahren bieten hier in der Regel die besten Methoden, das numerische Problem zu lösen.

Um klarzustellen, das wir hier keine allgemeine lineare Gleichung (5.1) lösen, sondern ein Gleichungssystem, welches einen linearen elliptischen Operator $\mathcal{L}$ diskretisiert, schreiben wir

$$\mathcal{L}u = f \tag{5.77}$$

Als praktische Anwendung kann man etwa unter $\mathcal{L}$ den Laplaceoperator $\nabla^2 = \Delta$ verstehen, u ist das Potentialfeld und f eine Ladung oder Masse. Wie später noch eingehender zu beschreiben ist, lösen wir dies Problem auf einem Gitter, welches einer Diskretisierung eines physikalischen, anschaulichen (nicht mathematischen) Raumes entspricht. Wir wählen die Maschenweite h. Dann kann man die Gleichung (5.77) "diskretisiert schreiben

$$\mathcal{L}_h u_h = f_h \tag{5.78}$$

Eine Näherungslösung $\tilde{u}$ von (5.78) habe dann den Fehler

$$v_h = u_h - \tilde{u}_h, \tag{5.79}$$

mit dem Residuum (oder auch Defekt)

$$r_h = \mathcal{L}_h \tilde{u}_h - f_h. \tag{5.80}$$

Wegen der Linearität kann man auch schreiben

$$\mathcal{L}_h v_h = -r_h \tag{5.81}$$

Wollen wir nun tatsächlich diese Gleichung lösen, so müssen wir uns ein Verfahren auswählen, welches dies zumindest als Näherungslösung leistet. Wir kennen bereits das Jacobi- oder Gauß-Seidelverfahren. Eine Näherung $\hat{\mathcal{L}}$ von $\mathcal{L}$ ist etwa der Diagonalteil von $\mathcal{L}$ mittels des Jacobiverfahrens oder die untere Dreiecksmatrix im Falle des Gauß-Seidelverfahrens

$$\hat{\mathcal{L}}_h \hat{v}_h = -r_h. \tag{5.82}$$

Wir erhalten damit den Korrekturterm

$$\tilde{u}_h^{neu} = \tilde{u}_h + \hat{v}_h. \tag{5.83}$$

Dies kann man nun iterativ lösen, ohne bisher einen Gewinn gegenüber anderen Verfahren zu erzielen. Betrachtet man allerdings nun, etwa durch eine Fourieranalyse, wie die Zwischenlösungen, oder Zwischenfehler $\hat{v}$ sich zur exakten Lösung hin verändern, so beobachtet man, dass die "glatte", also langwelligen Anteile sich nur sehr langsam verbessern, während die "unruhigen" oder rauhen, kurzwelligen Anteile sich schnell anpassen. Was "kurz" oder "lang" ist, wird im Bezug zur Maschenweite gesetzt. Daher ist es naheliegend, die auf dem Gitter mit der Maschenweite h als langwellig geltenden Komponenten durch ein gröberes Gitter als kurzwellige Anteile effizient zu verbessern. Man setzt daher $H = 2h$ und löst das Problem zwischenzeitlich auf dem gröberen Gitter. Man erhält somit die Gleichung

$$\mathcal{L}_H v_H = -r_H \tag{5.84}$$

Hierzu muss durch einen "Restriktions"- oder "Injektionsoperator" (fein-zu-grob-Operator) $\mathcal{R}$ ein r_H berechnet werden.

$$r_H = \mathcal{R} r_h \tag{5.85}$$

Haben wir so die Lösung $\tilde{v}_H$ auf einem groben Gitter ermittelt, so muss dies durch einen "Prolongationsoperator"$\mathcal{P}$ wieder zurück auf ein feines Gitter gebracht werden.

$$\tilde{v}_h = \mathcal{P}\tilde{v}_H. \tag{5.86}$$

Damit kann man dann ein neues $\tilde{u}_h^{neu} = \tilde{u}_h + \hat{v}_h$ mit (5.83) berechnen. Ein Algorithmus mit 2 Gittern lautet dann

1. Berechne ein $\bar{u}_h$ durch einige Schritte mit dem Einzelschritt- oder Gesammtschrittverfahren mit (5.78 – 5.83),

2. berechne das restringierte Residuum r_H auf dem groben Gitter mit (5.85),

3. lse (5.84) auf dem groben Gitter,

4. interpoliere die Grobgitterkorrektur mit (5.86) auf die Fehlerfunktion des feinen Gitters,

5. berechne die neue Lösung $\tilde{u}_h^{neu}$ mit (5.83), und

6. berechne erneut $\bar{u}_h^{neu}$ durch einige Schritte mit dem Einzelschritt- oder Gesammtschrittverfahren.

Diese Prozedur wird nun rekursiv ber mehrere Gitterebenen angewandt. Es gibt hierbei Varianten, in dem vom feinsten zum gröbsten Gitter und direkt wieder zurück fortgeschritten wird (V-cycle), oder in dem vor der Rückkehr zum feinsten Gitter erst wiederholt zwischen groben und mittleren Gitter gewechselt wird (W-cycle). Die Praxis zeigt das effizientere Verfahren. Eine Vielzahl von Varianten der Mehrgittermethode existiert. Hier wird auf die Spezialliteratur verwiesen.

5.7 Neue Begriffe

Pivotelement, Kondition, Equilibrierung,

5.8 Fragen

1. Welche grundsätzliche Zweiteilung läßt sich bei den Lösungsverfahren von Gleichungssystemen vornehmen?

2. Wie berechnet man numerisch effizient die Determinante einer Matrix?

3. Was erfährt man aus der Konditionszahl einer Matrix über die Lösung des Gleichungssystems?

4. Warum strebt man equilibrierte Matrizen an?

5. Für welche Gleichungssysteme sind die beschriebenen Lösungsverfahren effizient?

6. Welchen nachteilige Wirkung verhindert man mit der Housholdertransformation und was sind die Folgen für die Effizienz?

Kapitel 6

Ausgleichsrechnung

6.1 Vorbemerkungen

Anders als bei wissenschaftlichen Disziplinen, in denen wesentliche Ergebnisse aus Laborexperimenten folgen, nehmen in Geophysik und Meteorologie Feldexperimente eine zentrale Rolle ein. Als Folge gibt es oft Bedingungen, wo interessierende Parameter $x_1, \ldots, x_n$ nicht direkt gemessen werden können, weil sie wie in der festen Erde oder in der freien Atmosphäre direkten Beobachtungen nicht zugänglich sind. Seismische Experimente oder Fernerkundungen liefern, etwa mit Wellenlaufzeiten und Strahldichten, Messungen $y_1, \ldots, y_m$, die indirekt Rückschlüsse auf den Erdaufbau oder Atmosphärenzustände x_i schließen lassen.

Beobachtungen und Parameter seien nun unter m verschiedenen Bedingungen $z_k, k = 1, \ldots, m$ über theoretisch vorausgesetzte oder empirisch ermittelte Beziehungen f_k verknüpft

$$y_k = f(z_k; x_1, \ldots, x_n) =: f_k(x_1, \ldots, x_n), \ k = 1, \ldots, m. \tag{6.1}$$

Es müssen nun $m \geq n$ verschiedene Experimente oder unterschiedliche Beobachtungen y_i vorliegen, um in gewissen Sinne optimale Abschätzungen der Parameter $x_1, \ldots, x_n$ zu gewinnen. Das am häufigsten gewählte Optimalitätsmaß ist der auf Laplace und Gauß zurückgehende mittlere quadratische Abstand

$$\min_{x_i} \sum_{k=1}^{m} \left(y_k - f_k(x_1, \ldots, x_n)\right)^2. \tag{6.2}$$

Alternativ kann auch eine Maximumsnorm gewählt werden, mit der Aufgabe, den Ausdruck

$$\max_{1 \leq k \leq m} |y_k - f_k(x_1, \ldots, x_n)|$$

zu minimieren. Ist (6.2) überall stetig differenzierbar, so erhält man ein optimales x mit der Lösung der *Normalgleichung*

$$\frac{\partial}{\partial x_i} \sum_{k=1}^{m} \left(y_k - f_k(x_1, \ldots, x_n)\right)^2 = 0, \quad i = 1, \ldots, n. \tag{6.3}$$

6.2 Lineares Ausgleichsproblem

Sind die f_k lineare Abbildungen von x, so kann man mit einer Matrix $A \in \mathbb{R}^{m\times n}$ schreiben

$$\begin{pmatrix} f_1(x_1,\ldots,x_n) \\ \vdots \\ f_m(x_1,\ldots,x_n) \end{pmatrix} = Ax. \tag{6.4}$$

Im Kontext der linearen Gleichungssysteme geben wir also die Voraussetzung auf, dass A notwendigerweise eine quadratische und reguläre Matrix ist. Es wird zwischen den Fällen unterschieden:

Homogene Systeme mit

$$Ax = 0,\ A \in \mathbb{R}^{m\times n}, \tag{6.5}$$

wobei $x = 0$ immer eine Lösung ist, jedoch nur im Fall $m = n$ und bei regulärem A eindeutig. Andernfalls, mit $m < n$, gibt es einen Lösungsraum mit unendlich vielen Lösungen, oder nur der trivialen Lösung.

Rechteckige inhomogene Systeme

$$Ax = y,\ A \in \mathbb{R}^{m\times n},\ y \in \mathbb{R}^m,\ x \in \mathbb{R}^n. \tag{6.6}$$

Das Gleichungssystem kann unterbestimmt sein mit der Folge unendlich vieler Lösungen, eindeutig bestimmt sein, oder überbestimmt sein und gar keine Lösung besitzen. Die Lösung des Problems (6.4) als *Ausgleichsproblem* bietet sich hier als Ersatz an. Es gibt hier immer eine Lösung. Man wählt den kürzesten Lösungsvektor, der bei gegebener Norm eindeutig ist. Idee: Die durch das unterbestimmte Gleichungssystem gegebene Information wird als Zusatzinformation zu einem a-priori-Wissen herangezogen.

Nur im unterbestimmten homogenen Fall, also $\mathrm{Rang}(A) = p < n$, wo man in der Regel nicht an der trivialen Lösung interessiert ist, sondern an dem Lösungsraum der Dimension $n - p$, sucht man entsprechend viele linear unabhängige Vektoren $x_i, i = 1,\ldots,n-p$, die in der Regel auf $\|x\|_2 = 1$ normiert werden.

Bei der Wahl des mittleren quadratischen Abstandes sind folgende Begriffe von Bedeutung:

Definition 6.1 *1. Die Lösung kleinster Quadrate (least square solution, lss) von $y = Ax$ ist jeder Vektor $\hat{x} \in \mathbb{R}^n$ mit*

$$\|r\|_2 := \|y - A\hat{x}\|_2 = \min_{x\in\mathbb{R}^n} \|y - Ax\|_2\,, \tag{6.7}$$

. (minimales Residuum r)

2. Die Lösung kleinster Quadrate mit kleinster Norm $\hat{x}$ ist die Lösung mit (Lösung kleinster Euklidischer Länge)

$$\check{x} := \|\hat{x}\| = \min_{\hat{x}} \left\{ \|\hat{x}\| \,\middle|\, \|y - A\hat{x}\|_2 = \min_{x\in\mathbb{R}^n} \|y - Ax\|_2 \right\} \tag{6.8}$$

$\diamondsuit$

Zur Ermittlung der optimalen Lösung für den linearen Fall, ausgehend von der allgemeinen Bedingung (6.3), erhält man

$$\nabla_x \left((y - Ax)^T (y - Ax)\right) = 2(A^T Ax - A^T y) = 0. \tag{6.9}$$

Also erhält man

$$A^T Ax = A^T y. \tag{6.10}$$

Wir wollen zeigen, dass x genau dann die Lösung von (6.10) ist, wenn x auch die optimale Lösung von

$$\|y - Ax\|^2 = (y - Ax)^T (y - Ax) \tag{6.11}$$

ist. Hierzu beweisen wir folgenden

Satz 6.1 *Für das lineare Ausgleichsproblem*

$$\min_{x \in \mathbb{R}^n} \|y - Ax\| \tag{6.12}$$

a) *existiert mindestens eine Lösung x_0. Falls eine weitere Lösung x_1 existiert, so gilt $Ax_0 = Ax_1$.*

b) *Das Residuum $r := y - Ax_0$ ist eindeutig und es gilt $A^T r = 0$.*

c) *x_0 ist genau dann Lösung von $\min_{x \in \mathbb{R}^n} \|y - Ax\|$, falls x_0 Lösung von $A^T Ax = A^T y$ ist.*

Wegen der besonderen Bedeutung der Ausgleichsrechnung erfolgt hier der

Beweis: Zu $A \in \mathbb{R}^{m \times n}$ gibt es einen linearen Unterraum (Teilraum) $L := \{Ax | x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m$, der also von den Spalten von A aufgespannt ist. Senkrecht dazu steht der Teilraum $L^\perp := \left\{r | r^T z = r^T A = 0,\ \forall z \in L\right\}$. Wegen $y \in \mathbb{R}^m = L \oplus L^\perp$ gibt es die eindeutige Zerlegung

$$y = s + r,\ s \in L,\ r \in L^\perp. \tag{6.13}$$

Es gibt mindestens ein $x_0 \in \mathbb{R}^n$ mit $s = Ax_0$. Wegen $r^T A = A^T r = 0$, folgt unmittelbar

$$A^T y = A^T s = A^T Ax_0, \tag{6.14}$$

also ist x_0 die Lösung der Normalgleichung.
Umgekehrt gilt für jede weitere Lösung x_1 der Normalgleichung die eindeutige Zerlegung (6.13).

$$y = s + r,\ s = Ax_1 \in L,\ r = y - Ax_1 \in L^\perp. \tag{6.15}$$

daher gilt wegen der Eindeutigkeit (6.13) $Ax_0 = Ax_1$.

Optimalität:
Nimmt man ein beliebiges $x \in \mathbb{R}^n$ und definiert man $z = Ax - Ax_0$, und $r := y - Ax_0$, so folgt

$$\|y - Ax\|^2 = \left\| \overbrace{r}^{\in L^\perp} - \underbrace{z}_{\in L} \right\|^2 = \|r\|^2 + \|z\|^2 \geq \|r\|^2 = \|y - Ax_0\|^2 , \quad (6.16)$$

also ist das Optimum mit x_0 gegeben. $\diamondsuit$
Für den Fall dass die Spalten von A linear unabhängig sind, erhält man mit $A^T A \in \mathbb{R}^{n\times n}$ eine symmetrische und positiv definite Matrix. Die Lösung der Normalgleichung kann nun mit dem Choleskyverfahren erzielt werden. Man beachte, dass die Kondition dieser Matrix = $\mathrm{cond}(A^2)$, also quadratisch ungünstiger als die Matrix A selbst. Daher sucht man in der Regel stabilere Verfahren.

Formal erhält man jedoch zunächst

$$x = (A^T A)^{-1} A^T y. \quad (6.17)$$

6.3 Lösung des linearen Ausgleichsproblems

Eines der üblichen Lösungsverfahren und seine theoretische Grundlage sollen hier skizziert werden. Es gilt der Satz

Satz 6.2 *Sei Matrix $A \in \mathbb{R}^{m\times n}$ vom Rang p. Dann gibt es zwei orthogonale Matrizen $U \in \mathbb{R}^{m\times m}$ und $V \in \mathbb{R}^{n\times n}$, und eine Diagonalmatrix $S \in \mathbb{R}^{m\times n}$ mit*

$$U^T AV = S \text{ oder } A = USV^T.$$

Dabei besteht S aus p Diagonalelementen $\sigma_i > 0, i = 1, \ldots, p$, alle weiteren sind gleich Null.

$$S = \mathrm{diag}(\sigma_1, \ldots, \sigma_p, 0, \ldots, 0) = \begin{pmatrix} \hat{S} & 0 \\ 0 & 0 \end{pmatrix}, \ \hat{S} \in \mathbb{R}^{p\times p} \quad (6.18)$$

Die Diagonalelemente von S heißen Singulärwerte *von A.* $\diamondsuit$

Die Singulärwerte sind eindeutig bestimmt, jedoch nicht die orthogonalen Matrizen U und V. Zur Lösung gibt es folgenden

Satz 6.3 *Sei Matrix $A \in \mathbb{R}^{m\times n}$ vom Rang p und $A = USV^T$ eine Singulärwertzerlegung von A. Es sei ferner $d \in \mathbb{R}^m$*

$$U^T y =: d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \ d_1 \in \mathbb{R}^p, \ d_2 \in \mathbb{R}^{m-p},$$

und $z \in \mathbb{R}^n$ mit

$$V^T x =: z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}, \ z_1 \in \mathbb{R}^p, \ z_2 \in \mathbb{R}^{n-p}.$$

Ferner sei $\hat{z}_1 := \hat{S}^{-1} d_1$. So gilt für die Lösung kleinster Quadrate

1. *Alle Lösungen sind bestimmt durch*

$$\hat{x} = V \begin{pmatrix} \check{z}_1 \\ z_2 \end{pmatrix}, \; z_2 \text{ beliebig.}$$

2. *Jede der Lösungen liefert den selben Residuenvektor*

$$\hat{r} = y - A\hat{x} = U \begin{pmatrix} 0 \\ d_2 \end{pmatrix}, \; \|\hat{r}\|_2 = \|d_2\|_2 .$$

3. *Die eindeutige Lösung kleinster Quadrate mit kleinster Norm ist gegeben durch*

$$\check{x} = V \begin{pmatrix} \check{z}_1 \\ 0 \end{pmatrix}.$$

Zur praktischen Lösung kann man einen Algorithmus von Golub und Reinsch (1971) verwenden. Das Verfahren besteht aus 3 Schritten. Für die Matrix A wird der Fall $m > n$ gewählt. (Der anderslautende Fall wird durch entsprechende Transformation erhalten.) Die Singulärwertdarstellung lautet nun

$$A = U \begin{pmatrix} \tilde{S} \\ 0 \end{pmatrix} V^T, \; A \in \mathbb{R}^{m\times n}, \; U \in \mathbb{R}^{m\times m}, \; \tilde{S} \in \mathbb{R}^{n\times n}, \; V \in \mathbb{R}^{n\times n}. \tag{6.19}$$

Bidiagonalisierung von A: Hierzu wird mit der Householdertransformation H eine besondere Form der Transformation gewählt, deren Transformationsmatrizen so konstruiert sind, dass sie mit ihrer Kondition 1 die durch A gegebene Kondition nicht weiter verschlechtern. Mit geeignet gewählten $w \in \mathbb{R}^n$ und $\|w\|_2 = 1$ heißt die symmetrische orthogonale Matrix $H = I - 2ww^T$ *Householdermatrix*. Dabei erzielt man mit $2n-1$ Householdertransformationen ein

$$\begin{pmatrix} B \\ 0 \end{pmatrix} = Q_n(\ldots((Q_1 A)H_2)\ldots H_n) =: Q^T AH \; \in \mathbb{R}^{m\times n}, \tag{6.20}$$

mit

$$B = \begin{pmatrix} q_1 & e_2 & & & \\ & q_2 & \ddots & & \\ & & \ddots & \ddots & \\ & & & & e_n \\ & & & & q_n \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

Die Transformationsmatrizen Q_i eliminieren in der i-ten Spalte alle Elemente der i+1-ten bis zur m-ten Zeile. Die Transformationsmatrizen H_i werden so bestimmt, dass in der $i-1$ -ten Zeile alle Elemente der i+1-ten bis zur n-ten Spalte verschwinden.

Singulärwertzerlegung: Hier soll die Zerlegung

$$B = \check{U}\check{S}\check{V}^T$$

ermittelt werden Diese Teilaufgabe wird im Rahmen der Eigenwertberechnung behandelt.

Zusammensetzung der Singulärwertzerlegung: Die im vorangestellten Schritt gewonnenen Singulärwerte der Matrix $\check{S}$ müssen im negativen Falle mit -1 multipliziert werden (Multiplikation mit einer Diagonalmatrix). Danach werden die Elemente durch eine Permutationsmatrix geordnet. Das Zusammenfassen dieser Schritte führt zur Singulärwertzerlegung

$$A = USV^T, \tag{6.21}$$

wobei die umformenden Elementarmatrizen zusammengefasst wurden.

6.4 Statistische Interpretation

Das Ausgleichsproblem als Aufgabe zur Parameterbestimmung aus einer Vielzahl von Messwerten, wie in Gepophysik und Meteorologie üblich, folgt aus einer statistischen Formulierung. Typischerweise sind die y_i messfehlerbehaftete Beobachtungen, mit dem Mittelwert $\mathcal{E}[y_i] = \mu_i$ und der einheitlichen, aber nicht korrelierten Standardabweichung (Streuung) σ. Statistisch gesehen ist y damit ein Zufallsvektor, der durch ein Modell in Form von Matrix A nur in statistischen Sinne angepasst werden kann. Es muss daher eine Verbindung zu den benötigten statistischen Größen und den Größen des Ausgleichsproblems aufgezeigt werden können. Wichtig sind hier im gegebenen Kontext nur Mittelwert und Kovarianzmatrix. In Vektornotation erhält man Mittelwert und Kovarianzmatrix

$$\mathcal{E}[y_i] = 1/k_i \sum_{l=1}^{k_i} y_i^l = \mu_i, \qquad \mathcal{E}\left[(y-\mu)(y-\mu)^T\right] = \sigma^2 I. \tag{6.22}$$

Man erhält damit

$$\mathcal{E}[x] = \mathcal{E}\left[(A^TA)^{-1}A^Ty\right] = (A^TA)^{-1}A^T\mathcal{E}[y] = (A^TA)^{-1}A^T\mu. \tag{6.23}$$

Für die Kovarianzmatrix erhält man

$$\begin{aligned}\mathcal{E}\left[(x-\mathcal{E}(x))(x-\mathcal{E}(x))^T\right] &= \mathcal{E}\left[(A^TA)^{-1}A^T(y-\mu)(y-\mu)^TA(A^TA)^{-1}\right] \\ &= (A^TA)^{-1}A^T\mathcal{E}\left[(y-\mu)(y-\mu)^T\right]A(A^TA)^{-1} \\ &= \sigma^2(A^TA)^{-1}\end{aligned} \tag{6.24}$$

6.5 Nichtlineare Ausgleichsprobleme

In den meisten wissenschaftlich interessanten Fällen sind in Gepophysik und Meteorologie die Verknüpfungen zwischen gesuchten Parametern und Messungen nichtlinear. Es ist daher wichtig, sich von der Einschränkung der Linearität zu lösen. Die Annahme, dass die f_k in (6.3) ein lineares System ist, sei nun reduziert auf die Voraussetzung stetiger Differenzierbarkeit. Wir suchen ein $x^* = (x_1^*, \ldots, x_n^*)$, welches

$$\|y - f(x)\|^2 = \sum_{k=1}^{m} (y_k - f_k(x_1, \ldots, x_n))^2 \tag{6.25}$$

minimiert. Sei nun $x = (x_1, \ldots, x_n)^T$ eine Näherung des gesuchten Optimums x^* im geophysikalischen und meteorologischen Kontext oft auch *a priori*, *first guess*, oder *Hintergrundwert* (background value), ggf auch *Vorhersagewert* (forecasted value) genannt. Dann kann man mittels der Funktionalmatrix oder Jacobimatrix

$$Df(\xi) := \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}_{x=\xi} \tag{6.26}$$

mittels Taylornäherung eine verbesserte Stelle $\bar{x}$ finden mit

$$\min_{\bar{x} \in \mathbb{R}^n} \|y - f(x) - Df(x)(\bar{x} - x)\|^2 = \|r(x) - Df(x)(\bar{x} - x)\|^2 \leq \|y - f(x)\|^2 , \tag{6.27}$$

mit dem *Residuum* $r(x) := y - f(x)$.

Satz 6.4 *Gegeben sei $s(x) := \bar{x} - x$. So gibt es ein $\lambda > 0$, so dass die Funktion*

$$\phi(\tau) := \|y - f(x + \tau s)\|^2$$

für alle $0 \leq \tau \leq \lambda$ streng monoton fällt und es gilt

$$\phi(\lambda) := \|y - f(x + \lambda s)\|^2 < \phi(0) = \|y - f(x)\|^2$$

Beweis: Aus $f(x)$ stetig differenzierbar folgt auch $\phi(\tau)$ stetig differenzierbar. Daher

$$\begin{aligned} \phi'(0) &= \frac{d}{d\tau} \Big((y - f(x + \tau s))^T (y - f(x + \tau s)) \Big)_{\tau=0} \\ &= -2 \left(Df(x)s \right) (y - f(x)) \\ &= -2 \left(Df(x)s \right) r(x) \end{aligned} \tag{6.28}$$

Das linearisierte Minimierungsproblem (6.27) entspricht nach Satz (6.1) der Normalgleichung

$$Df(x)^T Df(x)s = Df(x)^T r(x). \tag{6.29}$$

Multiplikation dieser Gleichung von links mit s^T liefert direkt

$$s^T Df(x)^T r(x) = s^T Df(x)^T Df(x)s = \|Df(x)s\|^2$$

Daher ist

$$\phi'(0) = -2\,\|Df(x)s\| < 0.$$

falls Df regulär und $s \neq 0$. Wegen der geforderten Stetigkeit von f gibt es ein $\lambda > 0$ mit $\phi'(\tau) < 0$ für $0 \leq \tau \leq \lambda$. $\diamondsuit$

Als Lösungsverfahren bietet sich nun an, $\phi(\tau)$ längs des Vektors $s(x)$ zu minimieren, um alsdann von diesem Linienminimum erneut die Funktionalmatrix zu berechnen, und eine neue Liniensuche durchzuführen. Also,

1. für einem first guess oder Startvektor $x^{(0)}$ berechnet man die Lösung des linearisierten Ausgleichsproblems

$$\min_{s\in\mathbb{R}^n} \left\|r^{(i)}(x) - Df(x^{(i)})s\right\|^2$$

2. Längs des Lösungsvektors s berechnet man nun für $\phi(\tau) := \left\|y - f(x + \tau s^{(i)})\right\|^2$ nach einer geeigneten Folge die Werte $\phi(\tau) < \phi(0)$. Eine geeignete Weise ist ausgehend von hohen k abwärts das kleinste k zu wählen bei dem

$$\phi(2^{-k}) < \phi(0) = \left\|r(x^{(i)})\right\|^2 \tag{6.30}$$

3. Setze $x^{(i+1)} := x^{(i)} + 2^{-k}s^{(i)}$

6.6 Pseudoinverse

Wir kehren zur linearen Situation zurück. Im Falle linear unabhängiger Spalten von A konnte die Lösung mit (6.17) eindeutig beschrieben werden. Dieser Ansatz soll nun verallgemeinert werden für den Fall des linearen Ausgleichproblems (6.12)

$$x = (A^T A)^{-1} A^T y. \tag{6.17}$$

Definition 6.2 *Eine $n \times m$-Matrix A^+ heißt Pseudo-Inverse oder Moore-Penrose-Inverse einer $m \times n$-Matrix A, falls A^+ folgende Eigenschaften hat*

a) $A^+A = (A^+A)^T$

b) $AA^+ = (AA^+)^T$

c) $AA^+A = A,\ A^+AA^+ = A^+$

Es gilt der

Satz 6.5 *Jede Matrix A besitzt genau eine Pseudoinverse A^+.*

Insbesondere gilt das

Korollar 6.1 *Für alle Matrizen A gilt $A^{++} = A$ sowie $(A^+)^T = (A^T)^+$.*

Es gilt ferner der

Satz 6.6 *Sei Matrix $A \in \mathbb{R}^{m\times n}$ und $e_j, j = 1, \ldots, m$ j-te Spalte der Einheitsmatrix I_m, dann ist $A^+ \in \mathbb{R}^{n\times m}$ genau dann Pseudoinverse, wenn ihre j-te Spalte u_j den eindeutig kleinsten quadratischen Abstand kleinster Norm von $Au_j = e_j$ besitzt.*

Ferner gilt

Satz 6.7 *1. Der kleinste quadratische Abstand kleinster Norm von $y = Ax$ ist gegeben durch*

$$\check{x} = A^+ y. \tag{6.31}$$

2. Sei Matrix $B \in \mathbb{R}^{n\times n}$ und Rang$(B) = n$. Dann ist

$$B^+ = B^{-1}. \tag{6.32}$$

3. Hat $A \in \mathbb{R}^{m\times n}$ die Singulärwertzerlegung $A = USV^T$, dann ist die Pseudoinverse von A gegeben durch

$$A^+ = VS^+U^T, \tag{6.33}$$

wobei $S^+ = \mathrm{diag}(s_i^+)$ mit

$$s_i^+ := \begin{cases} s_i^{-1} & \text{falls } s_i > 0,\ i = 1, \ldots, p \\ 0 & \text{falls } s_i = 0. \end{cases} \tag{6.34}$$

Kapitel 7

Nullstellenbestimmung und Minimierung

7.1 Problemstellung

Nullstellenbestimmungen und Minimierungsprobleme sind Teil des größeren Problembereiches der Optimierung. Die im vorangestellten Kapitel behandelten Ausgleichsprobleme sind ein Spezialbereich. Es gibt viele Gebiet, die geophysikalische und meteorologische Studiengebiete direkt betreffen. Hier sind die für Fernerkundungsprobleme typischen Inversionsaufgaben eine offensichtliche Anwendung. Daneben tauchen diese Problembereiche oftmals auch als Teilaufgaben, etwa bei der Lösung partieller Differentialgleichungen auf. Aufteilungen der Problemklassen kann nach vielerlei Gesichtspunkten erfolgen. Falls möglich, wird die Optimierungsaufgabe als *unbeschränkte Optimierung* im reellen oder komplexen Zahlenraum vorgenommen, etwa bei der Parameteroptimierung, weil hier mit weniger Aufwand bei großer Methodenauswahl gearbeitet werden kann. Ist allerdings mindestens ein Parameter beschränkt bei der Annahme von Wertebereichen, etwa kann er als Konzentration oder Masse keine negativen Werte annehmen, so liegt ein *beschränktes Optimierungsproblem* vor.

Darüber hinaus gibt es noch diskrete oder ganzzahlige Optimierungsaufgaben. Diese Probleme findet man etwa bei der Frage nach der besten Konfiguration von Beobachtungsstationen. Eine weitere Klasse von Optimierungsproblemen betrifft den Fall diskontinuierlicher oder nicht-glatter Systeme, etwa bei Phasenübergängen von Zuständen. Im Folgenden werden nur Verfahren zur unbeschränkten Optimierung im Reellen behandelt.

Auch in dieser Problemklasse gibt es keinen idealen Algorithmus. Bei der Auswahl eines geeigneten Verfahrens sind einige Kriterien zu berücksichtigen:

Funktionsauswertung Ist eine Funktionsauswertung teuer, und kann nur wenige Male erfolgen? Und ist die Berechnung der Ableitung möglich?

Stetigkeit Ist die zu minimierende Funktion stetig oder stetig differenzierbar?

Speicherbedarf Ist die Dimension N des vorliegenden Problems so groß, so dass Zwischenspeicherungen nur von der Ordnung N behandelbar sind, oder können meist rechnerisch effizientere Verfahren angewandt werden, die einen Speicherbedarf der Ordnung N^2 beanspruchen?

Gegenstand dieses Kapitels ist die eingehendere Behandlung von Verfahren zur Lösung der Gleichungen (6.1, 6.3), oder Minimierungsprobleme wie (6.2), durch iterative Methoden. Ein Vorgriff war der Ansatz (6.30) zur Lösung des nichtlinearen Optimierungsproblems.

Zwei Algorithmengruppen sind geeignet, im unbeschränkten und hochdimensionalen Fällen diese Aufgabe nach Berechnungen der partiellen Ableitungen und Linienminimierungen zu lösen: Dies sind das *Konjugierte Gradientenverfahren* mit Speicherbedarf der Ordnung N, sowie quasi-Newton-Methoden mit Speicherbedarf der Ordnung N^2, wenn keine Varianten gewählt werden, die mit weniger Speicherungen auskommen.

Zumeist läßt sich die Lösung von Gleichungen in der Regel auf ein Problem der Suche nach Nullstellen reduzieren. Dies läßt sich allgemein beschreiben:

Sei $f : \mathbb{R}^n \to \mathbb{R}^n$ eine Abbildung von n reellen Funktionen $f_i(x_1, \ldots, x_n)$ mit

$$\begin{pmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_n(x_1, \ldots, x_n) \end{pmatrix} = \mathbf{0} \qquad x = (x_1, \ldots, x_n)^T. \tag{7.1}$$

Explizite Lösungen sind hier in der Regel nicht möglich. Der Ansatz lautet daher, mittels iterativer Verfahren eine *Iterationsfunktion* Φ zu lösen, deren Fixpunkt ξ Lösung von (7.1) ist, also $\Phi(\xi) = \xi$. Hierzu muss Φ in einer Umgebung von ξ stetig sein. Ziel ist es daher, eine in einer möglichst großen Umgebung vom Minimum möglichst schnell konvergierende Iterationsfunktion zu finden.

Auch wenn man nur ein Minimum einer stetig differenzierbaren Funktion $\min_{x \in \mathbb{R}^n} h(x)$ mit $h : \mathbb{R}^n \to \mathbb{R}$ sucht, so liefert die Suche nach den Nullstellen des Gradienten

$$g(x) := \begin{pmatrix} \frac{\partial h}{\partial x_1} \\ \vdots \\ \frac{\partial h}{\partial x_n} \end{pmatrix} = \mathbf{0} \tag{7.2}$$

das gesuchte Minimum. Die gestellte Aufgabe ist ein *Minimierungsproblem ohne Nebenbedingungen*. Im allgemeinen können jedoch m Nebenbedingungen der Art gefordert werden, dass für das Minimum etwa gelte: $h(x) \leq 0,\ i = 1, 2, \ldots, m < n$. Bei geophysikalischen und meteorologischen Anwendungen sind häufig positiv definite Größen zu bestimmen, wie etwa Dichten oder Konzentrationen. Die hierdurch geforderten Lösungen gehören streng genommen zur letztgenannten Klasse, auch wenn häufig diese Nebenbedingungen nicht berücksichtigt werden.

Eine systematische Methode zur Entwicklung von Iterationsfunktionen kann man aus der Taylorentwicklung gewinnen. Für eine Funktion $f : \mathbb{R} \to \mathbb{R}$ die

in einer Umgebung $U(\xi)$ der Nullstelle ξ mindestens k-mal differenzierbar ist, lautet mit der Taylorentwicklung die Nullstellenbedingung

$$f(\xi) = 0 = f(x_0) + f'(x_0)(\xi - x_0) + \frac{f''(x_0)}{2}(\xi - x_0)^2 + \dots \tag{7.3}$$

$$+ \frac{f^{(k)}(x_0 + \theta(\xi - x_0))}{k!}(\xi - x_0)^k \quad 0 < \theta < 1. \tag{7.4}$$

Damit erhält man mit der Iterationsfunktion

$$x^{(i+1)} = \Phi(x^{(i)}), \qquad \Phi(x^{(i)}) := x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}, \tag{7.5}$$

ein Iterationsverfahren 1. Grades, wobei die $x^{(i)}$ die gegen die Nullstelle konvergierenden Zwischenergebnisse sind. Dieser Algorithmus wird *Newton–Raphson–Verfahren* 1. Grades genannt.

Im Falle von (7.1) kann man mit der Schreibweise (6.26) analog das *allgemeine Newton-Verfahren* zur Lösung eines Systems von Gleichungen notieren

$$x^{(i+1)} = x^{(i)} - \left(Df(x^{(i)})\right)^{-1} f(x^{(i)}),\ i = 0, 1, 2, \dots \tag{7.6}$$

7.2 Quasi-Newtonverfahren

In diesem Abschnitt sollen weiterführende Methoden vorgestellt werden, die bei vielfältigen praktischen Problemgebieten direkt eingesetzt werden können. Gegeben sei eine reelle, zweimal stetig differenzierbare Funktion $h : \mathbb{R}^n \to \mathbb{R}$. Zu bestimmen sei $x \in \mathbb{R}^n$ mit

$$\min_{x \in \mathbb{R}^n} h(x) \tag{7.7}$$

Bei geophysikalischen und meteorologischen Anwendungen erwachsen diese Optimierungsaufgaben typischerweise bei Ausgleichrechnungen zu Inversionsproblemen, Fernerkundungen und der Datenassimilation, wobei der Abstand zu Messungen durch Parametervariationen vermindert werden sollen. Üblicherweise sind diese Aufgaben "schlecht gestellt" (ill posed), d.h. die Konditionszahl des linearisierten Problems ist groß, und ein Gradientverfahren wie im Falle von (7.6) wird kaum effizient zum Minimum iterieren. Die Funktion $h \in \mathcal{C}^2(\mathbb{R}^n)$ wird hier dann oft Kostenfunktion (cost function, objective function, penalty function) genannt. Wir bezeichen ferner den Gradienten von h mit

$$\operatorname{grad}(h) = \operatorname{grad}_x h = g(x)^T = Dh(x) = \left(\frac{\partial h}{\partial x_1}, \dots, \frac{\partial h}{\partial x_n}\right), \tag{7.8}$$

der am Minimum verschwinden soll (analog zu $f(x)$ beim Newtonverfahren (7.6)) und die infolge der Vertauschbarkeit der Ableitungen symmetrischen

Hessematrix (Matrix der 2. Ableitungen)

$$H(x) := DDh(x) = Dg(x)^T = \begin{pmatrix} \frac{\partial^2 h}{\partial x_1^2} & \cdots & \frac{\partial^2 h}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 h}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 h}{\partial x_n^2} \end{pmatrix}, \tag{7.9}$$

(analog zu $Df(x)$ beim Newtonverfahren (7.6)). In ähnlicher Weise wie beim Ansatz (6.30) versucht man mit einer Folge $x^{(0)}, x^{(1)}, \ldots$ eine lineare Minimisierung längs einer verfahrensabhängig bestimmten Suchrichtung $s^{(k)}$, wobei

$$x^{(k+1)} = x^{(k)} + \ s^{(k)}, \tag{7.10}$$

so dass

$$h(x^{(k+1)}) \approx \min_{\alpha} \phi_k(\alpha), \qquad \phi_k(\alpha) := h(x^{(k)} + \alpha s^{(k)}). \tag{7.11}$$

Vom Newtonverfahren (7.6) könnte man somit die Newtonsche Suchrichtung $-\left(Df(x^{(k)})\right)^{-1} f(x^{(k)}) \cong -H(x^{(k)})^{-1} g_k =: s^{(k)}$ mit $\alpha = 1$ übernehmen. In der geophysikalischen oder meteorologischen Praxis, in der auch $n = 10^7$ sein kann, ist es völlig illusorisch und ineffizient, bei jedem Iterationsschritt eine Matrix $H \in \mathbb{R}^{10^7 \times 10^7}$ zu berechnen, und dann im symmetrischen Fall etwa mittels Choleskyzerlegung zu invertieren. Gelingt es allerdings, die Matrizen $(H(x^{(k)}))^{-1}$ durch leichter zu berechnende $H^{(k)}$ zu ersetzen, so kann der Ansatz weiterhin sinnvoll sein[1]. Mit der Taylorzerlegung $g^{(k+1)} = g^{(k)} + H(x^{(k)})(x^{(k+1)} - x^{(k)})$ und $q^{(k)} := g^{(k+1)} - g^{(k)}$, $p^{(k+1)} := x^{(k+1)} - x^{(k)}$ erhält man eine Quasi-Newton-Gleichung,

$$H^{(k+1)} \left(g^{(k+1)} - g^{(k)}\right) = x^{(k+1)} - x^{(k)} \text{ oder } H^{(k+1)} q^{(k)} = p^{(k)}, \tag{7.12}$$

falls die gesuchte Matrix $H^{(k+1)}$ eine leicht zu berechnende, angenäherte Inverse der Hessematrix $H(x^{(k+1)})$ ist. Nun kann man insbesondere bei den oben genannten geophysikalischen Problemkreisen mit typischen Kostenfunktionen von quadratischer Form $h(x) = 1/2 x^T A x + b^T x + c$ mit A als positiv definiter Matrix, (im Allgemeinen die Inverse einer Kovarianzmatrix) so konstruieren, dass alle $H^{(k)}$ wiederum positiv definit sind. Dieser Vorgang kann nun auf verschiedene Weisen erfolgen, etwa indem man die schrittweisen Verbesserungen durch Matrizen vom Rang 1 vornimmt, und in diesen die Information aufnimmt, die man mit jeder Iteration gewinnt. Eine konkrete Möglichkeit gelingt in der folgenden Weise: Man konstruiert eine "update"-Funktion $H^{(k+1)} = H^{(k)} - auu^T$ mit $u \in \mathbb{R}^n$ zu bestimmender Vektor, a Skalar. Die Schreibweise uu^T bezeichnet eine Matrix des Ranges 1, da alle Zeilen (Spalten) als Linearkombination einer einzigen der Zeilen (Spalten) dargestellt werden können. Man setzt nun mit (7.12)

$$au \underbrace{u^T q^{(k)}}_{\in \mathbb{R}} = p^{(k)} - H^{(k)} q^{(k)},$$

[1] Achtung: Die Notation der Matrizen $(H(x^{(k)}))^{-1} \approx H^{(k)}$ ist etwas verwirrend, aber entspricht der Konvention

wobei $au^T q^{(k)} = 1$. Hieraus folgt dann für die gesuchte Rang-1-update-Funktion

$$H^{(k+1)} = H^{(k)} + \frac{\left(p^{(k)} - H^{(k)}q^{(k)}\right)\left(p^{(k)} - H^{(k)}q^{(k)}\right)^T}{\left(p^{(k)} - H^{(k)}q^{(k)}\right)^T q^{(k)}}. \tag{7.13}$$

Als Weiterentwicklung gibt es mit dem DPF-Verfahren einen Rang-2-Ansatz von (Davidon, Fletcher und Powell), in dem ein weiterer Matrixterm eingeführt wird $H^{(k+1)} = H^{(k)} - auu^T + bvv^T$: Hierbei werden mit den Ansätzen $auu^T q^{(k)} = p^{(k)}$ und $bvv^T q^{(k)} = -H^{(k)}q^{(k)}$ für jede Seite der Quasi-Newton-Gleichung eine eigene Rang-1-Matrix formuliert. Aus dem ersten Term folgt $u = p^{(k)} \Rightarrow au^T q^{(k)} = 1$. Aus dem zweiten Term folgt $v = H^{(k)}q^{(k)} \Rightarrow bvq^{(k)} = -1$. Damit erhält man für das DPF-Verfahren

$$H^{(k+1)}_{DPF} = H^{(k)} + \frac{p^{(k)}(p^{(k)})^T}{(p^{(k)})^T(q^{(k)})} - \frac{H^{(k)}q^{(k)}(H^{(k)}q^{(k)})^T}{(q^{(k)})^T H^{(k)}q^{(k)}}. \tag{7.14}$$

Das Verfahren, welches in der Praxis wohl die weitestgehende Bewährung gefunden hat, ist das Rang-2-Verfahren von Broyden-Fletcher-Goldfarb-Shanno (BFGS). Bei der Herleitung wird zunächst keine Rang-2-Näherung der Matrix $H^{(k+1)}$ gesucht wie in den beiden vorgenannten Ansätzen, sondern vielmehr eine Rang-2-Näherung $B^{(k+1)}$ der ursprünglichen Hessematrix $H(x^{(k+1)})$, welche dann exakt invertiert werden kann: $B^{(k+1)}H^{(k+1)} = I$. Diese Matrix erhält man mit

$$B^{(k+1)}_{BFGS} = B^{(k)} + \frac{q^{(k)}(q^{(k)})^T}{(q^{(k)})^T(p^{(k)})} - \frac{B^{(k)}p^{(k)}(B^{(k)}p^{(k)})^T}{(p^{(k)})^T B^{(k)}p^{(k)}}. \tag{7.15}$$

Man kann zeigen, dass man die hierzu exakt inverse Matrix erhält mit

$$H^{(k+1)}_{BFGS} = H^{(k)} + \left(1 + \frac{(q^{(k)})^T H^{(k)}q^{(k)}}{(p^{(k)})^T q^{(k)}}\right)\frac{p^{(k)}(p^{(k)})^T}{(p^{(k)})^T q^{(k)}} - \left(\frac{(p^{(k)})^T q^{(k)} H^{(k)} + H^{(k)}q^{(k)}(p^{(k)})^T}{(p^{(k)})^T q^{(k)}}\right). \tag{7.16}$$

7.3 Liniensuche

7.3.1 Vorbemerkungen

Im vorangegangenen Kapitel wurden bei der Aufstellung der Gleichung (7.11) zwei Probleme angesprochen: die Bestimmung einer Suchrichtung $s^{(k)}$ mittels Inversion der Hessematrix H, bzw. ihre Näherung, und die Minimierung längs der Suchrichtung $s^{(k)}$. Behandelt wurde bisher nur das erstgenannte Problem. Dieser Abschnitt gilt nun der Suche eines optimalen α mit $\min_\alpha h(x^{(k)} + \alpha s^{(k)})$, um $x^{(k+1)}$ gemäß (7.10) zu bestimmen. Eine gute Wahl des Liniensuchalgorithmus ist wichtig, weil damit die Anzahl von Funktionenberechnungen $h(\alpha)$, der Richtungsableitungen längs der Minimierungslinie $h'(\alpha)$, und und insgesamt dann auch Iterationsschritte im quasi-Newton-Verfahren oder bei anderen iterativen Verfahren, wie das der Kongugierten Gradienten, gespart werden können. Die Strategie der Bestimmung des nächsten Evaluationspunktes ist

daher von besonderer Bedeutung. Um nun zur Suche des Linienminimums eine effiziente Näherungsmethode zu gewinnen, gibt es verschiedene Methoden, die in der Regel ein Einschließungsintervall des Minimums ermitteln.

Wenngleich Liniensuchalgorithmen in der Regel nicht programmiert werden müssen, weil sie mit der Minimierungssoftware verfügbar sind, ist die Kenntnis ihrer Strategie sehr von Vorteil, um Fehlermeldungen bei fehlender oder mangelhafter Konvergenz zu verstehen und Abhilfe zu schaffen.

Vorab sei eine Betrachtung zur erreichbaren Genauigkeit dieses Intervalls gestellt. In der Nähe des Minimums b gilt nach Taylor, wenn der Gradient bereits vernachlässigbar klein gegenüber den anderen Termen ist,

$$h(x) \approx h(b) + 1/2h''(b)(x-b)^2. \tag{7.17}$$

Bei einer bei Gleitkommazahlendarstellung mit minimal darstellbarer Zahl $\operatorname{eps} h(b) = h(x) - h(b) \leq 1/2h''(b)(x-b)^2$ ist eine Abweichung $|x-b|$ vom Aufpunkt b zu klein, um erfasst zu werden falls

$$|x-b| < \sqrt{\operatorname{eps}}\,|b| \sqrt{\frac{2\,|h(b)|}{b^2 h''(b)}}. \tag{7.18}$$

Somit ist die erreichbare Genauigkeit *nur der Wurzel* von eps proportional!

Die Liniensuche ist eine iterative Methode, die eine gegen das Minimum konvergierende Reihe $\{\alpha_i\}$ bestimmt. Dies wurde mit einem einfacheren Verfahren bereits in (6.30) vorgestellt. Eine Bedingung für die Suche in Richtung größerer α lautet $s^{(k)} g(x^{(k)} + \alpha s^{(k)}) < 0$.

Es bezeichne nun $\bar{\alpha}^{(k)}$ das minimale α mit $h(x^{(k)} + \alpha s^{(k)}) = h(x^{(k)})$ (also das gleiche Höhenniveau des nächsten "Gegenhangs"). Dann $\alpha \in (0, \bar{\alpha}^{(k)})$

7.3.2 Verfahren nach Goldstein und Wolfe-Powell

Die Bedingungen nach Goldstein (1965) in der verkürzenden Schreibweise $h(\alpha) := h(x^{(k)} + \alpha s^{(k)})$ lauten etwa um α nach oben zu beschränken

$$h(\alpha) \leq h(0) + \alpha\rho h'(0), \quad \Leftrightarrow \quad h(x^{(k)}) - h(x^{(k+1)}) \geq -\alpha\rho g(x^{(k)})^T s^{(k)}, \tag{7.19}$$

wobei die Äquivalenz aus der Definition der Richtungsableitung folgt. Eine Beschränkung von α nach unten ist

$$h(\alpha) \geq h(0) + \alpha(1-\rho)h'(0), \tag{7.20}$$

wobei $\rho \in (0, 1/2)$ wählbar aber fest ist. Bei nicht-quadratischen Funktionen $h(\alpha)$ kann die Bedingung nach unten (7.20) das richtige Minimum ausschließen. Daher wurde sie bei Wolfe (1968) und Powell (1976) ersetzt durch

$$h'(\alpha) \geq \sigma h'(0) \qquad \sigma \in (\rho, 1) \quad \Leftrightarrow \quad g(x^{(k)} + \alpha s^{(k)})^T s^{(k)} \geq \sigma g(x^{(k)})^T s^{(k)}. \tag{7.21}$$

In der Praxis wird zumeist die striktere Bedingung gewählt

$$|h'(\alpha)| \leq -\sigma h'(0). \tag{7.22}$$

7.3.3 Goldene-Schnitt-Suche

Ein Verfahren, welches ohne Ableitungen auskommt bestimmt eine neue Evaluationsstelle nach dem Goldenen Schnitt. Zunächst sucht man ein das Minimum enthaltendes Intervall $[a, c]$, etwa ausgehend von $\left[a = 0, c = \bar{\alpha}^{(k)}\right]$ weiter zu verkleinern, welches das Minimum enthält. Nimmt man weiterhin an, dass nur ein Minimum längs des Linienvektors $s^{(k)}$ existiert, so ist die Existenz eines b mit $a < b < c : \; h(b) < \min\left(h(a), h(c)\right)$ eine hinreichende Bedingung hierfür. Gesucht wird nun im nächsten Schritt die Entscheidung, für welches $x : h(x) < \min\left(h(a), h(b)\right)$ oder $h(x) < \min\left(h(b), h(c)\right)$. Es gilt

$$w := \frac{b-a}{c-a}, \qquad 1 - w = \frac{c-b}{c-a}. \tag{7.23}$$

Nehmen wir ohne Beschränkung der Allgemeinheit an, für einen neuen Evaluationspunkt x gelte $b < x < c$ und setzen

$$z := \frac{x-b}{c-a}. \tag{7.24}$$

Es wird nun x so gewählt, dass

$$w + z = 1 - w \;\Rightarrow\; z = 1 - 2w. \tag{7.25}$$

Dies bedeutet also dass $b - a = c - x$. Andererseits soll aber auch (von Schritt zu Schritt) gelten

$$\frac{z}{1-w} = \frac{w}{1}. \tag{7.26}$$

Mit (7.25) und (7.26) erhält man nun die quadratische Gleichung

$$w^2 - 3w + 1 = 0 \quad \Rightarrow \quad w = \frac{3 - \sqrt{5}}{2} \approx 0.38197. \tag{7.27}$$

Insgesamt heißt dies, dass ein Intervall in jedem Schritt von einem kleineren, maximal der Länge $1 - 0.38197... = 0.61803...$ ersetzt wird.

7.3.4 Inverse Parabelinterpolation

Die oben vorgestellte Goldene-Schnitt-Methode erzwingt die Konvergenz im Rahmen der Rechengenauigkeit (7.18) auch bei sehr ungünstigem Verlauf der Kurve in Minimumnähe. Für den Fall dass aber die Umgebung des Minimums ausreichend glatt ist, kann eine parabolische Anpassung an das Tripel (a, b, c) und Ermittlung der Abszisse dieses Minimums zur effizienten Lösung führen. Das Minimum x einer Parabel durch die Werte $(h(a), h(b), h(c))$ kann explizit angegeben werden

$$x = b - \frac{(b-a)^2\left(h(b) - h(c)\right) - (b-c)^2\left(h(b) - h(a)\right)}{2(b-a)\left(h(b) - h(c)\right) - (b-c)\left(h(b) - h(a)\right)}. \tag{7.28}$$

Man kann auch nach Brent (1973) dieses Verfahren mit der Goldenen-Schnitt-Methode verknüpfen, indem man zunächst mit letzterem beginnt, und anhand der gewonnenen Zwischenergebnisse auf geeignete und dort beschriebene Weise die Glattheit und Eignung zur parabolischen Interpolation prüft.

7.4 Konjugierte-Gradienten-Verfahren

7.4.1 Konjugierte Richtungen

In vielen Fällen ist die Matrix A schwach besetzt, wobei ein festes Besetzungsschema gegeben ist, welches nicht notwendigerweise eine Bandmatrixstruktur bedeutet. Insbesondere bei der Lösung partieller Differentialgleichungen mit unregelmäßigen Randwerten liegt ein derartiger Fall vor. Hierzu bieten verschiedene Konjugierte-Gradienten-Verfahren (CG) effiziente Lösungsmethoden. Die folgende Darstellung folgt in weiten Teilen der anschaulichen Beschreibung von *Shewchuk, J.S., An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, School of Computer Science, Carnegie Mellon University, 1994* (verfügbar auf web). Wir starten wieder mit dem System (5.1), also $A \in \mathbb{R}^{n\times n}$, zusätzlich aber mit der vereinfachenden Annahme, dass A symmetrisch und positiv definit ist. Generell aber sind diese Verfahren nicht auf diese Vereinfachung angewiesen, wenngleich dann die Lösung am effizientesten ist.

Wir betrachten allerdings die äquivalente Aufgabe: minimiere die *quadratische Form*

$$h(x) = 1/2x^T Ax - b^T x + c. \tag{7.29}$$

Damit hat man einen konkreten Ausgangspunkt vom Typ (7.7). Nach der Ableitung gemäß (7.8) findet man

$$\operatorname{grad} h(x) = Ax - b = g(x) = -r(x), \tag{7.30}$$

mit der Lösungsbedingung $\operatorname{grad} h(x) = 0$ für Gleichung (5.1). Der Gradient ist also das negative Residuum $r = b - Ax = -g$ an der Stelle x. Da der Gradient nicht notwendigerweise auf das Minimum von (7.29) zeigt, aber bei einem eindeutigen Minimum, (wie durch quadratische Formen gegeben), sich dem gesuchten Minimum nähert, nutzt man den Gradienten, um ausgehend von einem Startpunkt $x_{(0)}$ längs seiner negativen Richtung ein Linienminimum $x_{(1)}$ zu finden, also

$$x_{(1)} = x_{(0)} - \alpha g(x_{(0)}) = x_{(0)} + \alpha r_{(0)} \tag{7.31}$$

(Die Iterationszählerindizes werden nunmehr tiefgestellt, da Vektoren nicht mehr komponentenweise dargestellt werden.) Die Suche nach dem α, welches $h(x_{(1)})$ minimiert, führt auf die Richtungsableitung mit der Bedingung, dass der neue Gradient $\operatorname{grad} h(x_{(1)})$ senkrecht auf dem alten $r_{(0)}$ steht

$$0 = \frac{d}{d\alpha} h(x_{(1)}(\alpha)) = \operatorname{grad} h(x_{(1)}) \frac{dx_{(1)}}{d\alpha} = \operatorname{grad} h(x_{(1)}) r_{(0)}. \tag{7.32}$$

Das Linienminimum $x_{(1)}$ist also dort, wo der neue Gradient $g_{(1)}$ senkrecht auf dem alten $g_{(0)}$ steht. Damit kann nun mit (7.31) α und somit die neue Näherung $x_{(1)}$ ermittelt werden

$$0 = g_{(1)}^T g_{(0)} = (Ax_{(1)} - b)^T g_{(0)} = (A(x_{(0)} - \alpha g_{(0)}) - b)^T g_{(0)} \tag{7.33}$$

Daher also

$$(Ax_{(0)} - b)^T g_{(0)} = \alpha (Ag_{(0)})^T g_{(0)}$$

Für das gesuchte α erhalten wir somit

$$\alpha = \frac{g_{(0)}^T g_{(0)}}{g_{(0)}^T A g_{(0)}} \tag{7.34}$$

Die Gradientenmethode lautet somit im Raum des Lösungsvektors x, und ferner nach Multiplikation mit A und Subtraktion von b im Bildraum

$$x_{(i+1)} = x_{(i)} - \alpha_{(i)} g_{(i)} \tag{7.35}$$

$$g_{(i+1)} = g_{(i)} - \alpha_{(i)} A g_{(i)}. \tag{7.36}$$

Gleichungen (7.30, 7.34, 7.35) bilden zusammen die Methode des *Steilsten Abstieges* (Steepest Descent), welche ein Minimum in Richtung des negativen Gradienten sucht. Es fehlt also die Ablenkung des negativen Gradienten in Richtung des Minimums, welche beim Quasi-Newton-Verfahren die inverse Hessematrix besorgte. Die Folge ist wieder ein umso ausgeprägterer Zickzackpfad, je größer die Konditionszahl ist.

Im trivialen Fall, wenn $A = cI, c$ Skalar, weist der Weg entgegen den Gradienten, der ja senkrecht zu den Hypersphären gleichen Residuumbetrages $\|r\| = \|-g\|$ steht, zum Minimum. Es ist lediglich eine Aufgabe der Liniensuche, hier das Minimum zu finden. Eine vergleichbare Situation haben wir im Falle der symmetrischen und positiv definiten Matrix A nur längs ihrer Eigenvektoren, deren aufwändige Berechnung wir aber vermeiden wollen. (Bei $A = cI$ ist dagegen jeder beliebige Vektor $\neq 0$ Eigenvektor.) Im trivialen Fall $A = cI$ ist ein Auffinden des Minimums bei exakter Liniensuche auch dann möglich, wenn man parallel zu einem beliebigen orthogonalen Koordinatensystem in entsprechend orthogonalen Suchrichtungen die jeweiligen Linienminima in n Schritten mittels der Orthogonalitätsbedingung (7.33) genau ermittelt. Es muss dann auch in keiner Suchrichtung (Dimension) erneut gesucht werden. Der Ansatz hierzu lautet mit einer Menge orthogonaler Suchrichtungen $d_{(0)}, d_{(1)}, \ldots, d_{(n-1)}$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)}. \tag{7.37}$$

Für den Restfehler $e_{(i+1)}$ soll dann gelten

$$0 = d_{(i)}^T e_{(i+1)} = d_{(i)}^T (e_{(i)} + \alpha_{(i)} d_{(i)}), \tag{7.38}$$

welches liefert

$$\alpha_{(i)} = -\frac{d_{(i)}^T e_{(i)}}{d_{(i)}^T d_{(i)}}. \tag{7.39}$$

Zwar ist der Fehler $e_{(i)}$ die unbekannte Größe, die das Problem lösen würde, aber bekannt ist $r_{(i)} = -Ae_{(i)}$. Nun gilt das Minimumskriterium (7.32) einer

Liniensuche nicht nur für die Richtung des Gradienten $\operatorname{grad} h(x_{(0)})$, sondern allgemein für jede weitere Suchrichtung $d_{(i)}$. Daher folgt

$$d_{(i)}^T r_{(i+1)} = d_{(i)}^T A e_{(i+1)} = 0. \tag{7.40}$$

Die folgende Betrachtung verallgemeinert die Minimumssuche von zirkularen Minima zu elliptischen: Eine geometrische Deutung der Flächen gleichen Residuums bei der symmetrischen und positiv definiten Matrix A liefert Hyperellipsopide, die man sich anschaulich längs der Hauptachsen entsprechend des Wertes der Eigenvektoren von Sphären zusammengedrängt vorstellen kann. Rechte Winkel öffnen oder schließen sich entsprechend der Drängungsrichtung. Eine verallgemeinerte Orthogonalitätstransformation ist daher sinnvoll, die eine entsprechende Orthogonalität mit transformiert. Zwei Vektoren $d_{(i)}, d_{(j)}$ sind genau dann *A-orthogonal* oder *konjugiert*, wenn gilt

$$0 = d_{(i)}^T A d_{(j)} = (A^{1/2} d_{(i)})^T A^{1/2} d_{(j)} = t_{(i)}^T t_{(j)}, \tag{7.41}$$

d.h. nach einer aufwändigen Transformation $t_{(j)} := A^{1/2} d_{(j)}$, die man sich wegen Definition der A-Orthogonalität ersparen kann, herrschten wieder triviale Bedingungen im Bildraum von $A^{1/2}$. Die hierzu in der Regel gewählte Norm $\|x\|_A$ wird oft als Energienorm oder Mahalanobisnorm bezeichnet

$$\|x\|_A := \sqrt{x^H A x}. \tag{7.42}$$

Mit der Definition (7.41) kann ein Vektor, der A-orthogonal auf dem Minimum der Suchlinie steht, weiterhin in Richtung des Minimums weisen. Nun kann mittels (7.40) für $\alpha_{(i)}$ in (7.39) ein entsprechender Ausdruck mit A-Orthogonalität darstellen, der bekannte Terme enthält

$$\alpha_{(i)} = -\frac{d_{(i)}^T A e_{(i)}}{d_{(i)}^T A d_{(i)}} = \frac{d_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \tag{7.43}$$

7.4.2 CG nach Heestenes-Stiefel und Fletcher–Reeves

Mit Hilfe von n linear unabhängigen, aber beliebigen Vektoren $u_0, u_1, \ldots, u_n$, etwa die kanonischen Einheitsvektoren, suchen wir uns nun eine Folge von n A-orthogonalen Vektoren, deren Linearkombination später von $x_{(0)}$ zum Minimum führen soll. (Später ist es sinnvoll, hierzu eine besonders geeignete Wahl von linear unabhängigen Vektoren zu treffen.) Ausgehend von $d_{(0)} := u_0$ setze

$$d_{(i)} = u_i + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)} \tag{7.44}$$

Die Koeffizienten β_{ij} ermittelt man durch Rechtsmultiplikation mit $A d_{(j)}$

$$d_{(i)}^T A d_{(j)} = u_i^T A d_{(j)} + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)}^T A d_{(j)}, \tag{7.45}$$

wobei die β_{ik} für $i > k$ definiert sind. Wegen der A-Orthogonalität der $d_{(i)}$ erhält man für $i > j$: $0 = u_i^T A d_{(j)} + \beta_{ij} d_{(j)}^T A d_{(j)}$,

$$\beta_{ij} = -\frac{u_{(i)}^T A d_{(j)}}{d_{(j)}^T A d_{(j)}}. \tag{7.46}$$

Zur weiteren Vorbereitung ist nun zu zeigen, dass $d_{(i)}^T r_{(j)} = 0$ und $u_i^T r_{(j)} = 0$. Sei nun

$$e_{(j)} := x_{(j)} - x = \sum_{l=j}^{n-1} \delta_{(l)} d_{(l)} \tag{7.47}$$

der Restfehler, der nach j Schritten übrig geblieben sei, also

$$\begin{aligned} r_{(j)} &= b - A x_{(j)} = b - A(x + e_{(j)}) = -A e_{(j)} \\ &= -A(e_{(j-1)} + \delta_{(j-1)} d_{(j-1)}) \\ &= r_{(j-1)} - \delta_{(j-1)} A d_{(j-1)} \end{aligned} \tag{7.48}$$

damit kann aus einem alten Residuum und der entsprechenden Suchrichtung ein neuer Gradient berechnet werden.

Nach Multiplikation von (7.47) mit $-d_{(i)}^T A$ von links, erhält man wegen (7.30) und der A-Orthogonalität der Suchrichtungen (7.41)

$$-d_{(i)}^T A e_{(j)} = -\sum_{l=j}^{n-1} \delta_{(l)} d_{(i)}^T A d_{(l)} \tag{7.49}$$

$$d_{(i)}^T r_{(j)} = 0 \qquad i < j. \tag{7.50}$$

Die Residuen (und damit auch die Gradienten) stehen also auch senkrecht auf *alle* vorherigen Suchrichtungen.

Multipliziert man dagegen (7.44) von rechts mit $r_{(j)}$, und trifft nachher die spezielle Wahl $u_i = -g_{(i)} = r_{(i)}$, dann folgt wegen (7.50)

$$d_{(i)}^T r_{(j)} = u_i^T r_{(j)} + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)}^T r_{(j)} \tag{7.51}$$

$$0 = u_i^T r_{(j)} = r_{(i)}^T r_{(j)} \quad i < j \tag{7.52}$$

$$d_{(i)}^T r_{(i)} = u_i^T r_{(i)} = r_{(i)}^T r_{(i)} \tag{7.53}$$

Wir suchen nun nach einem einfacheren Ausdruck für $\beta_{ij} = r_{(i)}^T A d_{(j)} / d_{(j)}^T A d_{(j)}$. Hierzu wird zunächst (7.48) mit erhöhtem Index von links mit $r_{(i)}^T$ multipliziert

$$r_{(i)}^T (7.48)_{(j+1)}: \qquad r_{(i)}^T r_{(j+1)} = r_{(i)}^T r_{(j)} - \alpha_{(j)} r_{(i)}^T A d_{(j)}$$

$$\text{mit (7.52)}: \qquad r_{(i)}^T A d_{(j)} = \begin{cases} r_{(i)}^T r_{(i)} / \alpha_{(i)} & \text{falls } i = k \\ -r_{(i)}^T r_{(i)} / \alpha_{(i-1)} & \text{falls } i = j+1 \\ 0 & \text{sonst.} \end{cases} \tag{7.54}$$

$$\Rightarrow \qquad \beta_{ij} = \begin{cases} \frac{1}{\alpha_{(i-1)}} \frac{r_{(i)}^T r_{(i)}}{d_{(i-1)}^T A d_{(i-1)}} & \text{falls } i = j+1 \\ 0 & \text{falls } i > j+1 \end{cases} \tag{7.55}$$

Wegen $i = j + 1$ in (7.55) kann man nach Substitution des inversen $\alpha_{(i-1)}$ zunächst mittels (7.43), dann mit (7.53) vereinfachend schreiben

$$\beta_{i,i+1} =: \beta_{(i)} = \frac{r_{(i)}^T r_{(i)}}{d_{(i-1)}^T r_{(i-1)}} = \frac{r_{(i)}^T r_{(i)}}{r_{(i-1)}^T r_{(i-1)}}. \tag{7.56}$$

Damit sind alle Teilausdrücke berechenbar, und insgesamt lautet das Konjugierte-Gradienten-Verfahren (CG) nach *Fletcher-Reeves* nun für $i = 1, \ldots, n$

$$d_{(0)} = r_{(0)} = b - Ax_{(0)} \tag{7.57}$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \tag{7.58}$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)} \tag{7.59}$$

$$r_{(i+1)} = r_{(i)} - \alpha_{(i)} A d_{(i)} \tag{7.60}$$

$$\beta_{i+1} = \frac{r_{(i+1)}^T r_{(i+1)}}{r_{(i)}^T r_{(i)}} \tag{7.61}$$

$$d_{(i+1)} = r_{(i+1)} + \beta_{(i+1)} d_{(i)} \tag{7.62}$$

Bei quadratischen Funktionen h ist das Minimum nach n Schritten erreicht. Das vorher von *Heestenes und Stiefel* (1952) enwickelte Verfahren unterscheidet sich vom oben gezeigten nur durch die die Bestimmung von $\alpha_{(i)}$ nicht durch (7.58), sondern durch iterative Liniensuche, wie im vorangestellten Abschnitt für verschiedene Methoden beschrieben.

Als wesentliches Merkmal bleibt festzuhalten, dass nur Werte des Iterationszyklus i gegeben sein müssen, um die Entwicklung weiterer orthogonaler Komponenten $i + 1$ zu gewährleisten, also keine der früheren $j < i$. Damit ist sowohl eine räumliche, d.h. den Speicherplatz betreffende Komplexität als auch eine "zeitliche", d.h. die Iterationsschritte betreffende Komplexität von $\mathcal{O}(n^2)$ auf $\mathcal{O}(n)$ reduziert. Darüber hinaus brauchen, etwa im Gegensatz zu den Eliminationsverfahren, Matrixeinträge, die $= 0$ sind, nicht "angefasst" werden, also bei etwa $m < n$ Einträgen $\neq 0$ ist die Komplexität dann lediglich $\mathcal{O}(m)$.

7.4.3 CG-Methoden bei nicht-quadratischen Funktionen

Im Falle allgemeiner, nicht-quadratischer Funktionen h_{nq} ist die Konvergenz zum Minimum *nicht* nach n Schritten erreicht. Oftmals wendet man einfach das Fletcher–Reeves-Verfahren wiederholt an, etwa jeweils nach n oder weniger Schritten. Dabei wird der Neustart eines jeden Zyklus wieder mit einem Schritt steilsten Abstieges begonnen. Eine Reihe weiterer verschiedener Methoden sind für den nicht-quadratischen Fall vorgeschlagen worden. Das Verfahren von Polak–Ribiere hat sich dabei als besonders leistungsfähig herausgestellt. Dabei wird das Fletcher-Reeves Verfahren weitgehend übernommen; nur (7.61) wird ersetzt durch

$$\beta_{i+1} = \frac{\left(r_{(i+1)} - r_{(i)}\right)^T r_{(i+1)}}{r_{(i)}^T r_{(i)}} \tag{7.63}$$

Hiermit wird im Allgemeinnen eine deutlich verbesserte Konvergenz erreicht. Allerdings ist die theoretische Ermittlung nach n Schritten nicht garantiert. In vielen praktischen Fällen kann man jedoch mit weniger als n Schritten eine ausreichende Genauigkeit erreichen.

7.5 Levenberg–Marquardt–Verfahren

Eine in geophysikalischen Anwendungen häufig vorkommende Problemstellung (7.7) ist eine leicht verwandelte Form von Gleichung (6.2), in der das Differenzenquadrat zwischen Messfehler y_i und den "Modellwerten" $f_k(x_1, \ldots, x_n), k = 1, \ldots, m$ mit der Standardabweichung σ_i der Messfehler skaliert wird, um mit minimalem Wert die unbekannten Parameter $(x_1, \ldots, x_n)$ zu bestimmen.

$$\min_{x_i} 1/2 \sum_{k=1}^{m} \left(\frac{y_k - f_k(x_1, \ldots, x_n)}{\sigma_k} \right)^2 =: \min_{x_i} h(x_1, \ldots, x_n). \tag{7.64}$$

Diese Summe $h(x)$ wird oft auch Kostenfunktion (cost function, merit function) genannt. Häufig wird hierzu als Notation auch χ^2 verwendet, da die Differenzen als normalverteilte Zufallswerte interpretiert werden können, die der χ^2–Verteilung folgen (siehe MATHMET 2). Das allgemeine Newton–Verfahren (7.6) lautet dann in der Notation von (7.8) und (7.9)

$$\delta x^{(i+1)} := x^{(i+1)} - x^{(i)} = -H^{-1}(x^{(i)}) g(x^{(i)}),\ i = 0, 1, 2, \ldots \tag{7.65}$$

Hierbei sind der Gradient mit offensichtlicher Schrittindizierung (i)

$$g_j^{(i)} = \frac{\partial h}{\partial x_j} = -\sum_{k=1}^{m} \frac{(y_k - f_k(x_1, \ldots, x_n))}{\sigma_k^2} \frac{\partial f_k}{\partial x_j} \tag{7.66}$$

und die Hessematrix

$$H_{jl}^{(i)} = \frac{\partial^2 h}{\partial x_j \partial x_l} = \sum_{k=1}^{m} \frac{1}{\sigma_k^2} \left\{ \frac{\partial f_k}{\partial x_j} \frac{\partial f_k}{\partial x_l} - (y_k - f_k(x_1, \ldots, x_n)) \frac{\partial^2 f_k}{\partial x_j \partial x_l}. \right\} \tag{7.67}$$

In der Regel kann man den 2. Term mit der zweifachen Ableitung in den meisten Fällen wegen seiner Kleinheit im Vergleich zum 1. Term vernachlässigen. Hinzu kommt, dass die Differenz $y_k - f_k(x)$ im Bereich des Messfehlerrauschens ist.

Allgemein erhält man das Gleichungssystem zur Lösung des allgemeinen Newton–Verfahrens mit (7.65)

$$\sum_{l=1}^{n} H_{jl}^{(i)} \delta x_l^{(i+1)} = -g_j^{(i)},\ j = 1, \ldots, n. \tag{7.68}$$

Nun hat man zwar mit (7.64) eine quadratische Form gewählt. Andererseits ändert sich bei nichtlinearen f_k die Hessematrix mit jedem Ort $x^{(i)}$ im Parameterraum. Also kann nicht mehr erwartet werden, dass bei schlecht bekannten $x^{(i)}$ die Annahme einer quadratischen Form sinnvoll ist, ja vielmehr sogar

schädigend wirkt. Dies ändert sich jedoch bei zweimal stetig differenzierbarem f_k in der Nähe des genen Minimums von $h(x)$.

Eine sicheres, aber in der Nähe des Minimums ineffizientes Verfahren war bereits mit der Methodik des Steilsten Abstieges (7.35) vorgestellt worden. Dies lautet hier nun mit einer noch zu bestimmenden Konstanten c

$$\delta x_j^{(i+1)} = -c g_j^{(i)},\ j = 1, \ldots, n. \tag{7.69}$$

Die Wahl von c erfordert eine kurze Überlegung: Ein zu kleines c behindert eine schnelle Konvergenz, während ein großes c schnell über das Minimum hinaus weisen läßt. Eine sinnvolle Wahl der Größenordnung von c kann durch die folgende "Dimensionsanalyse" gewonnen werden, wenn man annimt, dass $h(x)$ dimensionslos, aber die einzelnen Parameter x_i physikalisch dimensionsbehaftet sind. Sei $[a]_j$ die Dimension von x_j, dann ist die Dimension des Gradienten $[1/a]_j$. Hieraus folgt dann für die Dimension der Konstanten nach (7.69) $[c] = [a]_i^2$. Diesen Wert findet man aber, nach analoger Überlegung, als Inverse auf der Haupdiagonalen der Hessematrix, also $1/H_{jj}$. Um nicht der als größer angesehenen Gefahr ausgesetzt zu ein, zu große Schritte $\delta x^{(i+1)}$ auszuführen, wird ein schrittverkürzender Faktor λ eingeführt. Insgesamt kann man dann (7.69) umformulieren

$$\delta x_j^{(i+1)} = -\frac{1}{\lambda^{(i+1)} H_{jj}} g_j^{(i)}, \quad \text{oder } \lambda^{(i+1)} H_{jj} \delta x_j^{(i+1)} = -g_j^{(i)},\ j = 1, \ldots, n. \tag{7.70}$$

Die Grundidee des Levenberg-Marquardt–Verfahrens ist nun, die Newton–Methode und die des Steilsten Abstieges zu verbinden, wobei letztgenanntes Verfahren bei großem Abstand vom Optimum ein dominantes Gewicht erhält, aber dieses zugunsten der Newton–Methode in der Optimumsnähe aufgibt. Mit der Linearkombination beider Verfahren erhält man

$$\tilde{H}_{jj}^{(i)} = H_{jj}^{(i)}(1 + \lambda^{(i)}) \tag{7.71}$$

$$\tilde{H}_{jl}^{(i)} = H_{jl}^{(i)} \qquad j \neq l \tag{7.72}$$

Damit lautet die Levenberg-Marquardt–Formel

$$\sum_{l=1}^{n} H_{jl}^{(i)} \left(1 + \lambda^{(i)} \delta_{jl}\right) \delta x_l^{(i+1)} = -g_j^{(i)},\ j = 1, \ldots, n, \tag{7.73}$$

mit Kronecker δ_{jl}. In der Praxis wird eine Variable $\nu^{(i+1)} := \lambda^{(i+1)} H_{jj} \geq 0$ als schrittweise adjustierbare Größe gewählt, d.h. der Skalenbezug zu H_{jj} aufgegeben. In Matrixschreibweise findet man daher oft die Notation

$$\left(H^{(i)} + \nu^{(i)} I\right) \delta x^{(i+1)} = -g^{(i)}. \tag{7.74}$$

Die Addition von $\nu^{(i)} I$ bewirkt eine Verstärkung der Diagonaldominanz und positive Definitheit. Sie ist anschaulich gesprochen in gewisser Weise ein "Sicherheitsnetz" beim Abstieg, welches über Unregelmäßigkeiten und Abweichungen von einer quadratischen Form hinweghilft. In der Nähe des gemäß

Voraussetzung glatten Minimums wird dann der relativ an Gewicht gewinnende Newton-Teil eine bessere Näherung bieten und schneller konvergieren. Der folgende Algorithmus verwirklicht die Levenberg–Marquardt–Methode bei gegebenem $x^{(0)}$ und ν und berechnetem $h(x^{(0)})$,
für $i = 1, 2, \ldots$:

1. berechne $g(x^{(i)}), H(x^{(i)})$
2. (falls effizient möglich: ermittle positive Definitheit von $\left(H^{(i)} + \nu^{(i)} I\right)$ durch Faktorisierung. Falls nicht positiv definit, setze $\nu^{(i)} \leftarrow 4\nu^{(i)}$ und gehe zu 1.)
3. berechne $x^{(i+1)}$ durch Lösung von (7.74),
4. falls $h(x^{(i+1)}) \geq h(x^{(i)})$, vergrößere ν um etwa eine Größenordnung und gehe zu 1.,
   falls $h(x^{(i+1)}) < h(x^{(i)})$, verkleinere ν um etwa eine Größenordnung und gehe zu 1.

Eine Stopbedingung berücksichtigt den Fall, wenn weitere Schritte nicht nennenswerten Änderungen von h liefern. Ist H die Inverse einer wohlspezifizierten Beobachtungsfehler-Kovarianzmatrix, sind Änderungen von $h << 1$ nicht mehr von Bedeutung.

Kapitel 8

Eigenwert- und Singulärwertprobleme

8.1 Vorbemerkungen

Schwingungen und ungestörte Oszillationen sind typische Beispiele, für die eine Formulierung als Eigenwert– (Eigenvektor)problem (EWP) sinnvoll ist. Aber auch bei symmetrischen Matrizen, insbesondere Kovarianzmatrizen, können durch eine Eigenwert– und Eigenvektorzerlegung wertvolle statistische Merkmale des untersuchten Systems ermittelt werden. Empirische Orthogonalfunktionen und Principal Component Analyses, Principal Oscillation Patterns, u.ä. Begriffe beschreiben Größen, denen Eigenwertanalysen zugrunde liegen.

Die Auswahl der Löser für effiziente numerische Berechnungen hängt sehr von den Eigenschaften der Matrix ab. Auch wenn hier nur reelle Matrizen betrachtet werden sollen, so haben diese aber im Allgemeinen komplexe Eigenwerte. In den Bereich der Eigenwertaufgaben fallen zumeist 3 unterschiedliche Probleme:

1. Gegeben seien zwei Matrizen $A, B \in \mathbb{R}^{n\times n}$. Beim allgemeinen EWP sind $\lambda_k \in \mathbb{C}$ und $x_k \in \mathbb{R}^n$ gesucht, so dass gilt

$$Ax_k = \lambda_k B x_k,\ k = 1, 2, \ldots, n. \tag{8.1}$$

2. Im Falle des speziellen EWP ist $B = I$.

3. Ist die Matrix A nicht quadratisch, sondern $A \in \mathbb{R}^{m\times n}, m \neq n$, so sind Singulärwert- und –vektorenberechnung erforderlich.

Die numerischen Verfahren, die jeweils geeignet sind, hängen davon ab, ob eine der folgenden Situationen gegeben ist

- Ist die Matrix symmetrisch?
- Ist die Matrix dünn besetzt?

- Sind alle Eigenwerte und –vektoren erforderlich, oder nur die n-größten (kleinsten)?

Die folgende Darstellung lehnt sich an Köckler und Stör an.

8.2 Spezielles Eigenwertproblem

Die Eigenwerte λ_k einer Matrix $A \in \mathbb{R}^{n\times n}$ sind bekanntermaßen die Nullstellen des characteristischen Polynoms

$$p(\lambda) := \det(A - \lambda I).$$

Allerdings ist die numerische Bestimmung der Nullstellen dieses Polynoms oft numerisch instabil. Dieser Ansatz wird daher in der Regel vermieden.

Wir betrachten zuerst den häufig vorkommenden symmetrischen Fall. Es gilt der

Satz 8.1 *Sei Matrix $A \in \mathbb{R}^{n\times n}$ symmetrisch. Dann hat A nur relle Eigenwerte und die Eigenvektoren $x_k, k = 1, 2, \ldots, n$ sind orthogonal. Mit $U \in \mathbb{R}^{n\times n}$ als Matrix der Eigenvektoren x_k in den Spalten mit der Normierung $\|x_k\|_2 = 1$, gilt ferner*

$$U^T U = I.$$

Im Falle der nichtsymmetrische Matrizen sollen hier nur Bandmatrizen behandelt werden. Hierzu benötigen wir die

Definition 8.1 *1. Sei Matrix $T \in \mathbb{C}^{n\times n}$ regulär. So ist*

$$T^{-1}AT$$

eine Ähnlichkeitstransformation *der Matrix A.*

2. Eine Matrix $A \in \mathbb{R}^{n\times n}$ ist diagonalähnlich*, wenn es eine reguläre Matrix $T \in \mathbb{C}^{n\times n}$ gibt mit*

$$\Lambda := T^{-1}AT,$$

wobei $\Lambda := \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \Lambda \in \mathbb{C}^{n\times n}$ Diagonalmatrix ist.

Wir haben folgenden

Satz 8.2 *Sei Matrix $T \in \mathbb{C}^{n\times n}$ regulär und $C := T^{-1}AT$, so besitzen C und A dieselben Eigenwerte λ_k, und einem Eigenvektor $y \in \mathbb{C}^n$ von C entspricht der Eigenvektor $x = Ty$ von A*

$$Cy = \lambda y \;\Rightarrow\; Ax = \lambda x \quad mit \quad x = Ty.$$

Die Berechnung der Eigenwerte und -vektoren vollzieht sich nun in 4 Schritten:

1. Skalierung: Dies dient der Gewährleistung der numerischen Stabilität in den nachfolgenden Transformationen. Bei symmetrischen Matrizen werden die Diagonalelemente durch Zeilen– und Spaltenvertauschungen auf der Diagonalen dem Betrage nach aufsteigend so plaziert, dass $\min_i(A)_{ii} = (\tilde{A})_{11}$, u.s.w.. Mittels Vertauschungen nach (5.8) erhält man dann eine erste Ähnlichkeitstransformation.

Im Falle nicht–symmetrischer Matrizen wird mit einer Ähnlichkeitstransformation versucht, die Norm der transformierten Matrix $\tilde{A}$ zu minimieren, und die Norm einer Zeile mit derjenigen der entsprechenden Spalte anzugleichen. Dies soll durch eine Diagonalmatrix D geschehen,

$$\tilde{A} = D^{-1}AD, \tag{8.2}$$

deren Elemente $(D)_{ii}, i = 1, \ldots, n$ zur Vermeidung von Rundungsfehlern Vielfache der Rechnerbasis 2 sind.

2. Transfomation zur Tridiagonalmatrix: Die in Kapitel 5.4 erwähnte Householdertransformation wird im symmetrischen Fall angewandt, um $\tilde{A}$ zur Tridiagonalmatrix zu reduzieren. Im Unterschied zu (6.20) wird hierzu $\tilde{A}$ mit einer beidseitigen Transformationskette multipliziert

$$B = H_{n-2}^T \ldots H_1^T \tilde{A} H_1 \ldots H_{n-2}. \tag{8.3}$$

Mit $H := H_1 \ldots H_{n-2}$ lautet die Transformation

$$B = H^T \tilde{A} H. \tag{8.4}$$

Im Falle von nicht-symmetrischen Matrizen gelingt mit dieser Transformation eine Reduktion auf eine obere Hessenbergform, d.h. eine Matrix B mit $(B)_{ij} = 0$ für $i > j + 1$, d.h. Elemente unter der Subdiagonalen.

3. QR-Verfahren zur Diagonalisierung Die nunmehr gewonnenen Tridiagonal– (oder ggf. Hessenberg)-matrizen werden nun zu einer Diagonalmatrix (oder obere Blockdreiecksmatrix) transformiert. Zunächst wird eine erste QR–Zerlegung vorgenommen. Dabei gilt, als besondere Form der LR-Zerlegung, dass zu jeder Matrix $B \in \mathbb{R}^{n\times n}$ eine orthogonale Matrix Q und obere Dreiecksmatrix R existiert, so dass

$$B = QR. \tag{8.5}$$

Die Matrizen Q können etwa mit der Householdertransformation berechnet werden. Nach der QR-Zerlegung wird eine QR-Transformation vorgenommen. Dabei gilt

$$B = QR \;\rightarrow\; \tilde{B} = RQ. \tag{8.6}$$

Dies ist eine Ähnlichkeitstransformation, denn wegen $R = Q^T B$ folgt

$$\tilde{B} = RQ = Q^T BQ.$$

Der QR-Algorithmus fasst nun, ausgehend von $B^{(1)} := B$, die Sequenz QR-Zerlegung und QR-Transformation zu einer iterativen Folge zusammen

$$\begin{aligned} B^{(k)} &= Q^{(k)} R^{(k)} \\ B^{(k+1)} &:= R^{(k)} Q^{(k)} = Q^{(k)T} B^{(k)} Q^{(k)} \qquad k = 1, 2, \ldots . \end{aligned} \tag{8.7}$$

Im symmetrischen Fall konvergiert die Iteration gegen eine Diagonalmatrix. Im nicht–symmetrischen Fall konvergiert der Algorithmus gegen eine obere Blockdreiecksmatrix mit 2×2–Diagonalblöcken. In beiden Fällen sind die Eigenwerte von den Diagonalelementen oder Diagonalblöcken abzuleiten. Im letzten Fall sind also noch 2×2–Eigenwertaufgaben zu lösen. Bekannterweise garantieren nur symmetrische Matrizen reelle Eigenwerte.

Den QR-Algorithmus kann man beschleunigen, wenn man in geeigneter Weise die Diagonalelemente der Matrizen B_k modifiziert. Dies wird unter dem Begriff *Spektralverschiebung (shift)* verstanden. Nachdem wieder $B_1 = B$ gesetzt wurde, lautet der Algorithmus

$$\begin{aligned} B^{(k)} - \zeta^{(k)} I &= Q^{(k)} R^{(k)} \\ B^{(k+1)} &:= R^{(k)} Q^{(k)} + \zeta^{(k)} I, \quad k = 1, 2, \ldots , \qquad \text{wobei weiter gilt} \\ B^{(k+1)} &= Q^{(k)T} B^{(k)} Q^{(k)}. \end{aligned} \tag{8.8}$$

Als Anhalt zur Wahl von ζ_k kann die Empfehlung genommen werden, das letzte Diagonalelement von B_k zu wählen.

4. Berechnung der Eigenvektoren Die mit

$$Q := Q^{(1)} Q^{(2)} \cdots Q^{(m)} \tag{8.9}$$

erfolgte Transformation von einer Tridiagonalmatrix B zu einer Diagonalmatrix liefert mit den Spalten von Q die Eigenvektoren im symmetrischen Fall. Im nicht–symmetrischen Fall mit einer oberen Hessenbergmatrix müssen zuerst noch die Eigenwerte und –vektoren der oberen Dreiecksmatrix R

$$R y_k = \lambda_k y_k, \tag{8.10}$$

gefunden werden, welches direkt möglich ist. Danach werden die Eigenvektoren der oberen Hessenbergmatrix mittels

$$x_k = Q y_k \tag{8.11}$$

ermittelt. Im letzten Schritt erfolgt die Rücktransformation der Eigenvektoren der Tridiagonalmatrix und der oberen Hessenbergmatrix in die der Ausgangsmatrizen.

8.3 Allgemeines Eigenwertproblem

Grundsätzlich läßt sich das allgemeine Eigenwertproblem $Ax_k = \lambda_k Bx_k$ (8.1) auf ein spezielles EWP mit $B^{-1}Ax_k = \lambda_k x_k$ zurückführen, falls B invertierbar ist. Falls B schlecht konditioniert ist, ist von diesem Ansatz abzuraten. Hierbei können ferner spezielle Eigenschaften bei $B^{-1}A$ verloren gehen, wie etwa eine vielleicht gegebene Symmetrie.

Wir betrachten zunächst den Fall einer symmetrisch positiv–definiten Matrix B. Hierbei führt man auf der Basis der Transformation $y := L^T x$ folgende Schritte aus

1. Choleskizerlegung von $B = LL^T$
2. Invertieren von L^{-1}
3. Transformieren von A mit L^{-1}, d.h. $C := L^{-1}AL^{-T}$ (wobei die negative Transponierte die Inverse der transponierten Matrix bedeutet.)
4. Lösung des speziellen Eigenwertproblems
$$Cy = \lambda y. \tag{8.12}$$
5. Die ermittelten Eigenwerte λ gelten nicht nur für das transformierte System, sondern auch für die Ausgangsaufgabe. Mit der Rücktransformation
$$x_k := L^{-T} y_k \tag{8.13}$$
erhält man die ursprünglich gesuchten Eigenvektoren.

Sind beide Matrizen symmetrisch, aber nur positiv *semi*definit, so gilt der

Satz 8.3 *Seien $A, B \in \mathbb{R}^{n\times n}$ zwei symmetrische und positiv semidefinite Matrizen, dann gibt es eine reguläre Matrix $T \in \mathbb{R}^{n\times n}$, so dass $T^{-1}AT$ und $T^{-1}BT$ diagonal sind.*

Der QZ-Algorithmus löst den allgemeinen Fall, in dem keine der Matrizen A, B symmetrisch und positiv definit sind. Dies geschieht mit Hilfe des folgenden

Satz 8.4 *Seien $A, B \in \mathbb{R}^{n\times n}$. Dann gibt es zwei orthogonale Matrizen $Q, Z \in \mathbb{R}^{n\times n}$, so dass QAZ und QBZ beide obere Dreiecksmatrizen sind.*

Überführt man also das Eigenwertproblem $Ax = \lambda Bx$ in ein orthogonal äquivalentes $QAZy = \lambda QBZy$, so hat das letztgenannte System die selben Eigenwerte. Die Eigenvektoren kann man mit $x = Zy$ ermitteln.

Folgende Schritte verwirklichen den Algorithmus:

1. Mittels verallgemeinerter Householdertransformation wird A auf eine obdere Hessenbergmatrix und B auf eine obere Dreiecksgestalt reduziert.

2. Mittels QR-Algorithmus oder seiner shift-Variante wird A weiter auf eine obere Blockdreiecksmatrix analog (8.5) und folgende, reduziert. Dies beeinträchtigt nicht die Dreiecksform von B.

3. Eine weitere Transformation führt die obere Blockdreiecksmatrix A in eine echte Dreiecksmatrix. Hiermit werden die Eigenwerte ermittelt.

4. Die Dreiecksmatrizen werden zur Bestimmung der Eigenvektoren genutzt, die wiederum auf die ursprüngliche Aufgabe zurücktransformiert werden.

8.4 Singulärwertzerlegung

Für eine Matrix $A \in \mathbb{R}^{m \times n}$ mit $m \neq n$ ist kein Eigenwertproblem definiert. Allerdings kann man für die Matrix $A^T A$ Eigenwerte und Eigenvektoren angeben. Ein Vergleich mit Satz 6.7, in dem die Singulärwertzerlegung angesprochen worden ist, zeigt, dass man mit S die Diagonalmatrix der Singulärwerte erhalten hat, also die Wurzeln der Eigenwerte von $A^T A$. Ferner sind die Spalten von V und U die rechten und linken Singulärvektoren, die die Eigenvektoren von $A^T A$ und AA^T sind.

Die Lösung des Singulärwertproblems als Eigenwertaufgabe $A^T A$ ist wegen der quadratisch erhöhten Kondition in der Regel nicht ratsam. Der Lösungsgang wurde bereits bei der linearen Ausgleichsrechnung mit (6.20) in 6.3 beschrieben. Es blieb als eigentliche Singulärwertzerlegung von

$$B = \check{U}\check{S}\check{V}^T$$

zu zeigen, mit B Bidiagonalmatrix.

Die Zerlegung nach Golub und Reinsch (1971) gelingt iterativ mit

$$B_1 = B \tag{8.14}$$

$$B_{k+1} = U_k^T B_k V_k \; k = 1, 2, \ldots, \tag{8.15}$$

wobei U_k und V_k orthogonale Matrizen sind und die B_k bidiagonal bleiben. Dabei werden die Matrizen U_k und V_k

$$S = \lim_{k \to \infty} B_k \tag{8.16}$$

so konstruiert, dass S diagonal wird.

8.5 Dünn besetzte Matrizen

Dünn (sparse) besetzte Matrizen kommen zumeist bei der Diskretisierung partieller Differentialgleichungen vor. Dabei ist die Anzahl der Einträge $\neq 0$ so

klein und nach einem oft regelmäßigem Schema angeordnet, dass die Matrix nicht mehr explizit gespeichert werden braucht, und bei der Matrix×Vektor-Multiplikation nur auf die betreffenden Einträge $\neq 0$ zugegriffen werden muss. Häufig sind diese Matrizen symmetrisch und diagonaldominant.

Als einfaches Beispiel hierzu betrachten wir die Vektoriteration nach Mises und interessieren uns nur für den größten Eigenwert und zugehörigen Eigenvektor einer symmetrischen Matrix $A \in \mathbb{R}^{n\times n}$. Die Eigenwerte $\lambda_1 > \lambda_2 > \ldots > \lambda_n$ seien verschieden. Hierzu lautet der Algorithmus

1. Setze $z^{(0)} \in \mathbb{R}$ beliebig, $z \neq 0$, setze $i := 0$
2. Berechne
$$u^{(i)} := Az^{(i)}$$
und
$$z^{(i+1)} = \frac{u^{(i)}}{\|u^{(i)}\|}$$
3. Falls $i = 0$: dann $i := 1$, und gehe nach 2.
4. Falls
$$\left\|u^{(i)} - u^{(i-1)}\right\| < \epsilon \, \|A\|$$
gehe nach 6.
5. Setze $i := i + 1$ und gehe nach 2.
6. Bestimme größten Wert und Index k mit
$$\left|u_k^{(i)}\right| = \max_j \left|u_j^{(i)}\right|$$
7. Setze
$$\lambda_1 := \left\|u^{(i)}\right\| \cdot \operatorname{sign}\left(\frac{z_k^{(i)}}{u_k^{(i)}}\right)$$
sowie
$$x^{(1)} = z^{(i+1)}.$$
8. Dann sind $\lambda_1 \in \mathbb{R}$ und $x^{(1)} \in \mathbb{R}^n$ die gesuchten Näherungen des Eigenwertes und des zugehörigen Eigenvektors.

Kapitel 9

Gewöhnliche Differentialgleichungen

Gewöhnliche Differentialgleichungen beschreiben eine Abhängigkeit zwischen Funktionen $y(x)$, die von einer unabhängigen Variablen $x \in \mathbb{R}$ abhängen und ihren Ableitungen bis zu n-ter Ordnung $y^{(n)}(x)$. Anfangswertaufgaben (AWA) bezeichnen in geophysikalischen und meteorologieschen Anwendungen zumeist die zeitliche Entwicklung eines dynamischen Systems. Neben dem Differentialgleichungssystem müssen daher auch zur Lösung Anfangswerte gegeben sein. Gibt es Bedingungen zu zwei Zuständen, etwa zu verschiedenen Orten x oder Zeiten, so liegt eine Randwertproblem vor.

9.1 Anfangswertaufgaben

9.1.1 Problemstellung

Für ein explizites Diffentialgeichungssystem 1. Ordnung lautet die Aufgabe: Gegeben ist ein Vektor der Anfangswerte $y_0 \in \mathbb{R}^n$ und eine Funktion

$$f : \mathcal{C}\left([a,b] \times \mathbb{R}^n\right) \to \mathbb{R}^n. \tag{9.1}$$

Gesucht ist die Funktion $y(x) : \mathcal{C}^1\left([a,b] \times \mathbb{R}^n\right) \to \mathbb{R}^n$, für die gilt

$$\begin{aligned} y'(x) &= f(x, y(x)) \ \forall x \in [a,b] && \text{Differentialgleichung,} \\ y(a) &= y_0 && \text{Anfangswerte.} \end{aligned} \tag{9.2}$$

Ein implizites System 1. Ordnung lautet $y'(x) = f(x, y(x), y'(x))$.

Satz 9.1 *Jedes explizite Differntialgleichungssystem m-ter Ordnung läßt sich in ein äquivalentes System 1. Ordnung umformen.*

Beweis:
Es reicht eine Differentialgleichungskomponente zu betrachten, da mit allen

weiteren $n-1$ Komponenten in gleicher Weise verfahren werden kann. Diese laute

$$z^{(n)} = g(x, z, z', \ldots, z^{(n-1)}), \tag{9.3}$$

mit den Anfangswerten

$$z(a) = z_0,\ z'(a) = z'_0, \ldots, z^{(n-1)}(a) = z_0^{(n-1)}.$$

Dann erhält man eine Vektorfunktion $y(x)$ mit der Gleichsetzung

$$y(x) := (y_1(x), y_2(x), \ldots, y_n(x))^T \tag{9.4}$$

$$:= \left(z(x), z'(x), \ldots, z^{(n)}(x)\right)^T, \tag{9.5}$$

womit folgende Zuordnung gegeben ist

$$\begin{aligned} y'_1(x) &:= y_2(x), \\ y'_2(x) &:= y_3(x), \\ &\vdots := \vdots, \\ y'_{n-1}(x) &:= y_n(x), \\ y'_n &= g(x, z, z', \ldots, z^{(n-1)}), \end{aligned} \tag{9.6}$$

sowie

$$y(a) = y_0 = \left(z_0, z'_0, \ldots, z_0^{(n-1)}\right)^T.$$

Damit ist ein gewöhnliches DGL-System der Form (9.2) gegeben. $\diamond$

Die zwei notwendigen Bedingungen für die Existenz einer Lösung lauten

1. Die Funktion $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$ sei auf dem *Streifen*

   $$S := \{(x, y) | x \in [a, b], y \in \mathbb{R}^n\} \quad \text{mit } -\infty < a < b < \infty, \tag{9.7}$$

   definiert und stetig.

2. Es gelte die :

   Definition 9.1 Globale Lipschitzbedingung *Es gibt eine Zahl $L \in \mathbb{R}$, so dass gilt*

   $$\|f(x, y_1) - f(x, y_2)\| \leq L \|y_1 - y_2\| \ \ \forall (x, y_1), (x, y_2) \in S. \tag{9.8}$$

Anschaulich heißt dies, die Steigungen im Richtungsfeld $f(x, y) \in S$ müssen beschränkt bleiben.

DGL-Systeme, die diese Bedingungen nicht erfüllen, bedürfen besonderer Lösungsverfahren und werden hier nicht behandelt.

9.1.2 Elementare Methoden

Zur Prüfung einfacher numerischer Integrationsmethoden und ihrer numerischen Stabilität wird mit $n = 1$ die folgende "Modell"-Gleichung angewandt:

$$y' = \lambda y, \quad y(0) = 1 \;\text{ mit der Lösung }\; y(x) = \exp(\lambda x). \tag{9.9}$$

Die Lösung in dem Streifen S soll zunächst an den diskretisierten Stellen

$$x_j := a + jh,\; j = 0, 1, \ldots, N, \;\text{ wobei }\; h := \frac{b-a}{N}. \tag{9.10}$$

bestimmt werden.

Grundsätzlich kann man nun mit zwei verschiedenen Ansätzen Lösungsverfahren entwickeln:

Integralgleichung Das DGL-System (9.2) ist dem System der Integralgleichungen

$$y(x) = y_0 + \int_a^x f(\xi, y(\xi))d\xi \tag{9.11}$$

äquivalent.

Taylorreihe Die lokale Lösung kann man durch eine Taylorreihe

$$\begin{aligned} y(x_{j+1}) &= y(x_j) + hy'(x_j) + \mathcal{O}(h^2) \\ &= y(x_j) + hf\left(x_j, y(x_j)\right) + \mathcal{O}(h^2) \end{aligned} \tag{9.12}$$

approximieren.

Für alles Weitere setzt man nun $y_j \approx y(x_j)$ und $f_j := f(x_j, y_j)$. Das einfachste Verfahren ist nun das *Explizite Eulerverfahren* oder *Polygonzugverfahren*:

$$y_{j+1} = y_j + hf_j,\; j = 0, 1, \ldots, N-1, \tag{9.13}$$

welches sich sowohl aus dem Tayloransatz als auch dem Integralgleichungsansatz herleiten läßt. Im letzten Fall erhält man mittels der Rechteckregel

$$\int_a^x f(\xi, y(\xi))d\xi \approx hf(x_j, y_j). \tag{9.14}$$

Wertet man den Funktionswert f nicht bei x_j sondern am Ende des Schrittes bei x_{j+1} aus, so erhält man das *Implizite Eulerverfahren*

$$y_{j+1} = y_j + hf_{j+1},\; j = 0, 1, \ldots, N-1. \tag{9.15}$$

Es ist also hier notwendig, bei jedem Schritt ein Gleichungssystem zu lösen.

Bei beiden Lösungsverfahren wurden Schritte der Länge h durchgeführt. Daher fallen sie in die Klasse der *Einschrittverfahren* (ESV). Das einfache *Mehrschrittverfahren* (MSV) rechnet mittels der Mittelpunktregel mit einer doppelten Schrittweite, wobei man das folgende explizite 2-Schrittverfahren erhält:

$$y_{j+2} = y_j + 2hf_{j+1},\; j = 0, 1, \ldots, N. \tag{9.16}$$

Am Anfang benötigt man aber ein ESV um den Wert y_{j+1} zu ermitteln.

Anwendung auf die Modellgleichung

In den folgenden Anwendungen wird geprüft, mit welchem Aufwand und Genauigkeit die vorgenannten Verfahren in der Lage sind, die analytisch bekannte Lösung der Modellgleichung nachzubilden. Alle Verfahren haben hier den Anfangswert $y_0 = 1$.

Euler explizit	Euler implizit	Mittelpunktregel
$y_1 = 1 + \lambda h$	$y_1 = \frac{1}{1-\lambda h}$	$y_1 = \exp(\lambda h)$ exakt!
$y_2 = (1 + \lambda h)^2$	$y_2 = \frac{1}{(1-\lambda h)^2}$	$y_2 = 1 + 2h\lambda \exp(\lambda h)$
$\vdots$	$\vdots$	$\vdots$
$y_j = (1 + \lambda h)^j$	$y_j = \frac{1}{(1-\lambda h)^j}$	$y_j = \exp(\lambda j h)\left(1 - \frac{\lambda^3 jh}{6}h^2 + \frac{\lambda^3}{12}h^3\right)$ $+(-1)^j \exp(\lambda j h)\frac{\lambda^3}{12}h^3 + \mathcal{O}(h^4)$

Mit $h \to 0,\ j \to \infty$ so dass $\bar{x} = jh = \text{const}$, erhält man $y(x) = \exp(\lambda \bar{x})$ für alle 3 Verfahren.

Man erhält mit $h > 0$ und $\lambda \ll 0$ folgendes, völlig abweichendes Verhalten

Euler explizit Für den Fall dass $\lambda h < -1$ erhält man für die y_i oszillierendes Anwachsen und eine völlig falsche Lösung.

Euler implizit Die Lösung bleibt stabil für $\lambda < 0$ und fällt monoton.

Mittelpunktregel man erhält eine gute Approximation der exakten Lösung, aber erfährt mit dem Term $(-1)^j \exp(\lambda j h)\frac{\lambda^3}{12}h^3$ einen wachsend oszillierende Überlagerung.

Konsistenz

In Gleichung (2.18) wurde an Hand der Rundungsfehler gezeigt, wie sich bei einem Algorithmus Fehler fortpflanzen und in jedem Schritt neue Fehler zusätzlich generiert werden können. So besteht zum "Zeit"punkt j auch der Fehler $y_j - y(x_j)$ von Anfangswertalgorithmen aus den bis zur Berechnung von y_{j-1} summierten Fehlern und dem beim Schritt j gemachten Fehler, dem *lokalen Diskretisierungsfehler*.

Definition 9.2 *1. Ein Einschrittverfahren ist durch seine Verfahrensfunktion* Φ *gegeben*

$$y_{j+1} = y_j + h\phi(x_j, y(x_j)). \tag{9.17}$$

2. Unter einem lokalen Diskretisierungsfehler *eines ESV versteht man die Differenz*

$$d_{j+1} := \frac{1}{h}\left(y(x_{j+1}) - y(x_j)\right) - \phi(x_j, y(x_j)) \tag{9.18}$$

3. *Ein ESV Φ heisst konsistent mit der gegebenen gewöhnlichen Differentialgleichung, falls gilt*

$$\max_{j=1,\ldots,N} \|d_j\| \to 0 \quad \textit{mit} \;\; h \to 0. \tag{9.19}$$

wobei $N \to \infty$ wegen $Nh = b - a$ constant.

4. *Ein ESV Φ hat die Konsistenzordnung p, falls für $K > 0$ gilt:*

$$\max_{j=1,\ldots,N} \|d_j\| \leq Kh^p = \mathcal{O}(h^p) \;\; \textit{mit} \;\; h \to 0, \tag{9.20}$$

Der lokale Diskretisierungsfehler beschreibt also nur den Fehleranteil, den man in einem Schritt i in Bezug auf den exakten Wert zusätzlich zu allen früher akkumuierten Fehlern beiträgt. Nach der obigen Definition haben die Eulerverfahren beide die Konsistenzordnung $p = 1$.

Letztlich benötigt ist allerdings eine Lösung, die auch am gewünschten Endpunkt $b = a + hN$ nur einen tolerierbar kleinen Fehler aufweist.

Definition 9.3 1. *Der globale Diskretisierungsfehler ist durch*

$$g_j := y(x_j) - y_j \tag{9.21}$$

gegeben.

2. *Ein Verfahren zur Lösung der Anfangswertaufgabe* (9.2) *heißt* konvergent*, falls*

$$\max_{j=1,\ldots,N} \|g_j\| \to 0 \quad \textit{mit} \;\; h \to 0. \tag{9.22}$$

3. *Ein ESV hat die* Konvergenzordnung *p, wenn zudem für ein $K > 0$ gilt:*

$$\max_{j=1,\ldots,N} \|g_j\| \leq Kh^p = \mathcal{O}(h^p) \;\; \textit{mit} \;\; h \to 0, \tag{9.23}$$

Satz 9.2 *Ein konvergentes Einzelschritterfahren ist auch konsistent. Ist das Verfahren von der Konsistenzordnung p, so ist es auch von der Konvergenzordnung $\|g_j\| = \mathcal{O}(h^p)$.*

Die Konsistenz ist allerdings nur eine notwendige Vorraussetzung!

Ein weiterer Begriff ist der der Stabilität eines numerischen Algorithmus. Ohne strenge Definition wurde dieser Begriff (und seine Antinomie) bereits im Kapitel Rundungsfehler benutzt, wo die beschränkte Genauigkeit der Zahlendarstellung und Gleitpunktrechnung die Stabilität beeinträchtigen kann. In diesem Kapitel ist es die finite Darstellung der Schrittweiten $h > 0$, welches letztlich zur Verletzung der Stabilität führen kann. Es gibt eine Vielzahl von Stabilitätsbegriffen. Hier wird der Begriff der asymptotischen Stabilität verwandt:

Definition 9.4 *Ein ESV Φ ist asymptotisch stabil, wenn Φ eine Lipschitzbedingung erfüllt, d.h. es gibt eine Zahl $L > 0$, so dass*

$$\|\Phi(x_j, y_j) - \Phi(x_j, \tilde{y}_j)\| \leq L \|y_j - \tilde{y}_j\| \ \forall (x_j, y_j), (x_j, \tilde{y}_j) \in S. \tag{9.24}$$

Es gilt der folgende fundamentale

Satz 9.3 *Ein konsistentes Verfahren der Ordnung p, welches asymptotisch stabil ist, besitzt auch eine Konvergenz der Ordnung p.*

Daneben gilt auch der in der Praxis nicht so bedeutende

Satz 9.4 *Ein konsistentes Verfahren der Ordnung p, welches konvergent ist, ist auch asymptotisch stabil.*

Die Lipschitzbedingung kann auch dazu dienen, den globalen Fehler einer Verfahrensfunktion Φ abzuschätzen:

Satz 9.5 *Ein ESV sei von der Konsistenzordnung p und erfülle die Lipschitzbedingung. Eine Lipschitzfunktion E_L sei definiert als*

$$E_L(x) := \begin{cases} \frac{e^{L|x-a|}-1}{L} & \text{falls } L > 0 \\ x & \text{falls } L = 0. \end{cases} \tag{9.25}$$

So gilt

$$\|y_j - y(x_j)\| \leq h^p K E_L(x_j - a). \tag{9.26}$$

Auf Grund ihres exponentiellen Wachstums kann diese Abschätzung als schwach gelten.

Steife Differentialgleichungssysteme

Bei bestimmten Problemen findet man Differentialgleichungen, bei denen um Größenordnungen verschiedene Koeffizienten, etwa bei λ unseres Modellproblems auftreten. Derartige Systeme nennt man *steif*. Ein typisches Beispiel ist die Reaktionskinetik in der Atmosphärenchemie. Es gelte das folgende, einfache lineare Reaktionsbeispiel

$$y_1 \overset{k_1}{\to} y_2 \overset{k_2}{\to} y_3,$$

$k_1 = 1$ und $k_2 = 101$ mit den Anfangswertkonzentrationen

$$y_1(0) = 1, \ y_2(0) = 1, \ y_3(0) = 1.$$

Damit lautet gemäß Reaktionskinetik das DGL-System

$$\begin{aligned} y_1' &= -y_1 \\ y_2' &= y_1 - 101 y_2 \\ y_3' &= 101 y_2. \end{aligned} \tag{9.27}$$

Die Lösung hierzu lautet

$$\begin{aligned} y_1(x) &= e^{-x} \\ y_2(x) &= 0.01e^{-x} + 0.99e^{-101x} \\ y_3(x) &= 3 - 1.01e^{-x} - 0.99e^{-101x}. \end{aligned}$$

Die Koeffizientenmatrix

$$\begin{pmatrix} -1 & 0 & 0 \\ 1 & -101 & 0 \\ 0 & 101 & 0 \end{pmatrix} \tag{9.28}$$

hat neben dem Eigenwert $\lambda_3 = 0$ die um 2 Größenordnungen verschiedenen Eigenwerte $\lambda_1 = -1$ und $\lambda_2 = -101$. Hierbei diktiert der kleinere Eigenwert die Schrittweite. Im Allgemeinen, bei nichtlinearen Systemen wie in der Reaktionskinetik, können die die Steifheit bestimmenden Eigenwerte wechseln. Hier liefert eine Linearisierung an den verschiedenen Stellen x Analysierungsmöglichkeiten mittels der Eigenwerte der Funktionalmatrix

$$f_y(x) := \begin{pmatrix} \frac{\partial f_1(x,y)}{\partial y_1} & \cdots & \frac{\partial f_1(x,y)}{\partial y_n} \\ \vdots & & \vdots \\ \frac{\partial f_n(x,y)}{\partial y_1} & \cdots & \frac{\partial f_n(x,y)}{\partial y_n}. \end{pmatrix} \tag{9.29}$$

In diesem Falle nennt man das DGLsystem an der Stelle x steiff, wenn es dort nur negative Eigenwerte hat, die stark unterschiedlich sind, also

$$Re\lambda_i < 0,\ i = 1, \ldots, n, \qquad \max_i |Re\lambda_i| >> \min_j |Re\lambda_j|.$$

Ein Maß für die Steifheit kann man mit

$$\frac{\max_i |Re\lambda_i|}{\min_j |Re\lambda_j|} \tag{9.30}$$

angegeben werden. Vergleiche mit der Definition der Kondition eines Gleichungssystemes.

9.1.3 Einschrittverfahren

Unter den Einschrittverfahren umfassen die Runge-Kutta-Verfahren die am meisten verbreiteten Methoden. Ihre allgemeine Form lautet

$$\begin{aligned} y_{j+1} &= y_j + h\Phi(x_j, y_j) \\ \Phi(x_j, y_j) &= \sum_{l=1}^{m} \gamma_l k_l \ \text{ mit} \\ k_l &= f(x_j + \alpha_l h, y_j + h \sum_{s=1}^{m} \beta_{ls} k_s),\ l = 1, \ldots, m. \end{aligned} \tag{9.31}$$

Mit $\beta_{ls} = 0$ für $s \geq l$ ist das Verfahren explizit. Ein halbimplizites Verfahren liegt vor bei $\beta_{ls} = 0$ für $s > l$. In allen anderen Fällen ist das Verfahren implizit. Die *Stufe* eine RK-Verfahrens ist mit m gegeben, und $m^2 + 2m$ Parameter bestimmen das Verfahren.

Folgende Forderungen sind an ein Runge–Kutta Verfahren zu stellen:

- Ein Runge-Kutta–Verfahren ist konsistent, falls

$$\sum_{i=1}^{m} \gamma_i = 1. \tag{9.32}$$

- Jedes k_k soll eine h^2-Approximation von $y'(x_j + \alpha_l h)$ sein, wobei

$$\alpha_l = \sum_{s=1}^{m} \beta_{ls}. \tag{9.33}$$

- Die Konsistenzordnung soll maximal sein.

Das folgende Schema stellt die Konstruktion von RK-Verfahren dar

$$\begin{array}{l|lll} \alpha_1 & \beta_{11} & \beta_{12} \ \ldots & \beta_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_m & \beta_{m1} & \beta_{m2} \ \ldots & \beta_{mm} \\ \hline \sum_i \gamma_i = 1 & \gamma_1 & \gamma_2 \ \ldots & \gamma_m \end{array} \tag{9.34}$$

Sind die $\beta_{ij} \neq 0$ für $i > j$, so ist das Verfahren *implizit*. Sind $\beta_{ij} \neq 0$ für $i \geq j$, aber $\beta_{ij} = 0$ für $i > j$, so ist das Verfahren *semi–implizit*.

Für $m = 1$ erhält man wegen (9.32) $\gamma_1 = 1$ und daher

$$y_{i+1} = y_i + hf(x_i + \alpha_1 h, y_i + \beta_{11} k_1).$$

Im expliziten Fall ist $\beta_{11} = 0$ und wegen (9.33) $\alpha_1 = 0$, womit das explizite Eulerverfahren vorliegt. Mit $\alpha_1 = \beta_{11} = 1$ erhält man das implizite Eulerverfahren

$$\begin{aligned} y_{i+1} &= y_i + hk_1 \\ &= y_i + hf(x_i + h, y_i + hk_1) \\ &= y_i + hf(x_i + h, y_{i+1}). \end{aligned} \tag{9.35}$$

Das Runge–Kutta–Verfahren im eigentlichen Sinne ist jenes mit der Konsistenzordnung $p = 4$ und der Stufe 4

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad \text{mit} \tag{9.36}$$

$$\begin{aligned} k_1 &:= f(x_j, y_j) \\ k_2 &:= f(x_j + \frac{1}{2}h, y_j + \frac{1}{2}hk_1) \\ k_3 &:= f(x_j + \frac{1}{2}h, y_j + \frac{1}{2}hk_2) \\ k_4 &:= f(x_j + h, y_j + hk_3). \end{aligned} \tag{9.37}$$

Das entsprechende Schema lautet

$$\begin{array}{r|cccc} \alpha_i: 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline \gamma_i & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

Schrittweitensteuerung

Mit Verfahren 4. oder 5. Stufe ist es nicht möglich, eine höhere als 4. Ordnung zu erreichen. Andererseits kann dies in Bereichen von x ungenügend bleiben, wo starke Variationen von f auftreten. Passt man im gesamten Intervall $[a, b]$ die Schrittweite h diesen besonders kritischen Bereichen an, so steigt der Rechenaufwand für andere Bereiche unnötig hoch an, und es werden durch diese unnötigen Schritte immer auch die entsprechenden Rundungsfehler zusätzlich generiert.

Es bleibt daher nur als Ausweg, die Schrittweite h anzupassen. Diese **Schrittweitensteuerung** (SWS) würde idealerweise mit Kenntnis des globaren Fehlers erfolgen. Da dieser aber nicht bekannt ist und nur sehr aufwändig abgeschätzt werden kann, wird eine Schätzung des **lokalen Fehler** herangezogen. Hierzu rechnet man mit zwei Verfahren verschiedener Stufe oder Ordnung und ermittelt den Unterschied der Zwischenergebnisse. Damit sich der Rechenaufwand nicht verdoppelt, nimmt man zwei Verfahren, die weitgehend bei der Berechnung der k_i übereinstimmen, also bis auf jene zusätzlichen der höheren Stufe. Diese Verfahren nennt man *eingebettete* Runge–Kutta–Methoden.

Ein Beispiel ist das Runge–Kutta–Verfahren von Merson, welches zwei Verfahren 4. Ordnung, eines davon 4. Stufe, in das andere der 5. Stufe einbettet. Mittels k_5 gelingt dann eine Fehlerabschätzung. Das Schema lautet

$$\begin{array}{r|ccccc} 0 & & & & & \\ \frac{1}{3} & \frac{1}{3} & & & & \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} & & & \\ \frac{1}{2} & \frac{1}{8} & 0 & \frac{3}{8} & & \\ 1 & \frac{1}{2} & 0 & -\frac{3}{2} & 2 & \\ \hline y_{j+1}^{[4]} & \frac{1}{2} & 0 & -\frac{3}{2} & 2 & \\ \hline y_{j+1}^{[5]} & \frac{1}{6} & 0 & 0 & \frac{2}{3} & \frac{1}{6} \end{array}$$

Fox und Meyers (1987) geben folgende Fehlerabschätzung an

$$y(x_{j+1}) - y_{j+1}^{[5]} \approx \frac{h}{5}(y_{j+1}^{[4]} - y_{j+1}^{[5]}) = \frac{1}{30}(-2k_1 + 9k_3 - 8k_4 + k_5). \qquad (9.38)$$

Ein Algorithmus, der sich diese Fehlerabschätzung zu Nutze macht, könnte etwa bei einem $\frac{h}{5}(y_{j+1}^{[4]} - y_{j+1}^{[5]}) > \epsilon$ den Schritt verwerfen und mit halbierten h erneut berechnen. Ist dagegen die Fehlerabschätzung $< \epsilon/10$, so kann mit verdoppelter Schrittweite fortgefahren werden. In der Praxis bleiben Testrechnungen zu machen, ob Genauigkeit und Effizienz angemessen sind.

9.1.4 Mehrschrittverfahren

Wir beschränken uns auf lineare Mehrschrittverfahren (MSV), dem *Adams–Verfahren*. Anders als bei Runge-Kutta-Verfahren, bei dem Zwischenschritte bei Bruchteilen von h berechnet wurden, werden nun zur Berechnung der Lösung für $y(x_{i+m})$ an der Stelle x_{i+m} die Werte an den m vorherigen Stellen herangezogen. Es seien also
gegeben: Verfahrenskoeffizienten $a_0, a_1, \ldots, a_m$, wobei $a_m = 1$, und $b_0, b_1, \ldots, b_m$; ferner die Lösungswerte $y_j, y_{j+1}, \ldots, y_{j+m-1}$.
Gesucht ist der Lösungswert für y_{j+m}.
Der allgemeine Ansatz lautet

$$y_{j+m} + \sum_{i=0}^{m-1} a_i y_{j+i} = h \sum_{k=0}^{m} b_k f_{j+k} \tag{9.39}$$

Bei einem impliziten Verfahren ist $b_m \neq 0$.

Die grundsätzliche Idee ist nun die Funktion f in dem Intervall $[x_j, x_{j+m-1}]$ (oder $[x_j, x_{j+m}]$ im impliziten Fall) durch ein Interpolationspolynom zu ersetzen

$$P(x_{j+k}) = f(x_{j+k}, y_{j+k}),\ k = 0, 1, \ldots, m-1 \ \text{ oder } m. \tag{9.40}$$

Der numerische Ansatz der DGL lautet dann

$$y_{j+m} = y_l + \int_{x_l}^{x_{j+m}} P(x)dx \ l = j+m-1 \ \text{ oder } \ l = j+m-2. \tag{9.41}$$

Die wichtigsten Basisverfahren lauten nun
Adams–Bashforth (explizit): $l = j + m - 1$

$$y_{j+2} = y_{j+1} + \frac{h}{2}(3f_{j+1} - f_j) \tag{9.42}$$

Nyström (explizit) $l = j + m - 2$

$$y_{j+2} = y_j + 2hf_{j+1} \tag{9.43}$$

Adams–Moulton (implizit): $l = j + m - 1$

$$y_{j+2} = y_{j+1} + \frac{h}{12}(5f_{j+2} + 8f_{j+1} - f_j) \tag{9.44}$$

Milne–Simpson (implizit): $l = j + m - 2$

$$y_{j+2} = y_j + \frac{h}{3}(f_{j+2} + 4f_{j+1} + f_j) \tag{9.45}$$

Konsistenz

Der Diskretisierungsfehler (9.18) für lineare MSV lautet

$$d_{j+m} = \frac{1}{h}\sum_{i=0}^{m} a_i y(x_{j+i}) - \sum_{k=0}^{m} b_k f(x_{j+k}, y(x_{j+k})). \tag{9.46}$$

Die Verfahrenskoeffizienten definieren folgende beiden Polynome

1. charakteristisches Polynom $\rho(\zeta) = \sum_{i=0}^{m} a_i \zeta^i$,
2. charakteristisches Polynom $\sigma(\zeta) = \sum_{i=0}^{m} b_i \zeta^i$.

Es gilt nun der

Satz 9.6 *Die Startrechnung eines MSV sei konsistent. Ein MSV ist konsistent, falls gilt*

$$\rho(1) = 0 \qquad \textit{und} \qquad \rho'(1) = \sigma(1). \tag{9.47}$$

Prädiktor–Korrektor–Verfahren

Die Stabilität der Verfahren muss durch zusätzliche Bedingungen gegeben, sein, die hier aber nicht behandelt werden. Es gilt aber, dass implizite Verfahren stabiler sind als explizite. Im Falle nichtlinearer Gleichungssysteme ist jedoch oft die Auflösung impliziter Verfahren schwierig. Bei einer Kombination von expliziten und impliziten Verfahren kann dies jedoch vermieden werden.

Dieses kombinierte Verfahren besteht aus 3 Schritten

1. Prädiktorschritt (explizit)

$$y_{j+m}^{[0]} = \sum_{i=0}^{m-1} a_i^* y_{j+i} + h\sum_{k=0}^{m-1} b_k^* f_{j+k} \tag{9.48}$$

2. Auswertung der rechten Seite

$$f_{j+m} = f(x_{j+m}, y_{j+m}^{[0]}) \tag{9.49}$$

3. Korrektorverfahren (implizit)

$$y_{j+m}^{[1]} = hb_m f_{j+m} - \sum_{i=0}^{m-1} a_i y_{j+i} + h\sum_{k=0}^{m-1} b_k f_{j+k}. \tag{9.50}$$

Diese Methode verbindet die Einfachheit expliziter Verfahren mit fast der Stabilität impliziter Verfahren.

Bevor ein MSV angewandt werden kann, muss durch eine Startrechnung die Werte $y_1, \ldots, y_{j+m-1}$ berechnet worden sein. Hierzu können entweder ESV oder MSV mit wachsender Stufe und Ordnung herangezogen werden. Um hierdurch nicht die Qualität späterer Rechnungen zu beeinträchtigen, sollten variable Schrittweiten gewählt werden.

9.2 Rand- und Eigenwertaufgaben

Randwertaufgaben sind gewöhnliche DGL, deren Lösungen an *beiden* Randpunktn a *und* b Bedingungen einzuhalten haben. Eigenwertaufgaben haben zusätzlich zur Lösung noch einen Parameter λ zu bestimmen, so dass die Lösung aus einem Eigensystem besteht, welches aus abzählbar unendlich vielen Eigenwerten λ_i und zugeordneten Eigenfunktionen $y_i(x)$ besteht.

Eine typische *Randwertaufgabe 2. Ordnung* lautet
Gegeben sind eine stetige Funktion $f(x, y, y')$ und $\alpha, \beta \in \mathbb{R}$.
Gesucht ist eine Funktion $y(x) \in \mathcal{C}^2[a, b]$, für die gilt

$$\begin{aligned} y'' &= f(x, y, y') \\ y(a) = \alpha, &\qquad y(b) = \beta. \end{aligned} \tag{9.51}$$

Im Falle einer *Eigenwertaufgabe* gilt
Gegeben ist eine stetige Funktion $f(x, y, y')$.
Gesucht sind Eigenwerte λ_i und zugeordnete Funktionen $y_i(x) \in \mathcal{C}^2[a, b]$, für die gilt

$$\begin{aligned} y'' &= \lambda f(x, y, y') \\ y(a) = 0, &\qquad y(b) = 0. \end{aligned} \tag{9.52}$$

Oft liegen kompliziertere Formen von Randbedingungen vor, bei denen höhere Ableitungen vorkommen, Systeme von DGL oder implizite Systeme, sowie Randbedingungen von allgemeinerer Form gegeben sind, wie z.B.

$$r_j(y(a), y'(a), y(b), y'(b)) = 0. \quad j = 1, 2 \tag{9.53}$$

Eine Aufgabenklasse ist durch Sturm-Liouville-Probleme gegeben, die z.B. auch geeignet sind, die vertikale Schichtung der Atmosphäre oder des Ozean zu beschreiben.
Seien q, g stetige Funktionen auf dem Intervall (a, b), sowie $p(x) > 0,\ a \le x \le b$. Ferner sind gegeben $\alpha, \beta \in \mathbb{R}$, und $a_i, b_i \in \mathbb{R},\ i = 1, \ldots, 4$, so dass

$$\text{Rang} \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \end{pmatrix} = 2.$$

Gesucht sind $y(x) \in \mathcal{C}^2(a, b)$, für die gilt

$$-(p(x) \cdot y'(x))' + q(x) \cdot y(x) = g(x), \tag{9.54}$$

$$a_1 y(a) + a_2 y'(a) + a_3 y(b) + a_4 y'(b) = \alpha \tag{9.55}$$

$$b_1 y(a) + b_2 y'(a) + b_3 y(b) + b_4 y'(b) = \beta \tag{9.56}$$

Beispiel: Durchgebogener Balken
Die Länge eines homogenen, ideal elastischen Balkens, der an seinen Enden

frei beweglich aufliegt, sei 2, mit dem Längenintervall $[-1 \leq x \leq +1]$. Die Biegesteifigkeit ist das Produkt aus dem Elastizitätsmodul E und dem Flächenträgheitsmoment J, welches hier in der folgenden Weise längenabhängig parameterisiert sei: $J(x) = J_0/(1+x^2),\ J_0$ konstant. Der Balken unterliege einer konstanten transversalen Belastung h und einer axialen Kraft $P := EJ_0$. Die Differentialgleichung für das Biegemoment $M(x)$ lautet nun

$$M''(x) + \frac{P}{EJ(x)} M(x) = -h. \tag{9.57}$$

An den frei beweglichen Enden verschwinden die Biegemomente

$$M(-1) = M(1) = 0. \tag{9.58}$$

Mit der Transformation $y = -M/h$ erhält man nun das Sturm–Liouville–Problem

$$\begin{aligned} -y'' - (1+x^2)y(x) &= 1 \\ y(-1) = y(1) &= 0. \end{aligned} \tag{9.59}$$

9.2.1 Differenzenverfahren

Zur Lösung der RWA mittels Differenzenverfahren unterteilt man das Intervall $[a, b]$ in $n+1$ gleiche Teile mit den Grenzen

$$x_i := a + i \cdot h,\ i = 0, 1, \ldots, n+1$$

und der Länge

$$h := \frac{b-a}{n+1}.$$

Man kann nun die Ableitungen einer DGL durch finite Differenzen approximieren und diese dann in die DGL einsetzen. Über den Tayloransatz ermittelt man für die ersten 4 Ableitungen etwa folgende Näherungen der Ordnung $\mathcal{O}(h^2)$

$$\begin{aligned} y'(x_i) &= \frac{y(x_{i+1}) - y(x_{i-1})}{2h} + \mathcal{O}(h^2) \\ y''(x_i) &= \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} + \mathcal{O}(h^2) \\ y'''(x_i) &= \frac{y(x_{i+2}) - 3y(x_{i+1}) + 3y(x_{i-1}) - y(x_{i-2})}{2h^3} + \mathcal{O}(h^2) \\ y^{(4)}(x_i) &= \frac{y(x_{i+2}) - 4y(x_{i+1}) + 6y(x_i) - 4y(x_{i-1}) + y(x_{i-2})}{2h^3} + \mathcal{O}(h^4) \end{aligned}$$

Die Konstruktion von finiten Differenzen höherer Ordnung ist möglich und oft auch angebracht.

Beispiel 9.1 *Gegeben ist die folgende DGL*

$$\begin{aligned} -y''(x) + q(x)y(x) &= g(x) \\ y(a) = \alpha \qquad y(b) &= \beta \end{aligned} \tag{9.60}$$

Mittels der Funktionswerte $q_i := q(x_i)$, $g_i := g(x_i)$ *und als Ansatz* $y_i \approx y(x_i)$ *läßt sich das lineare Gleichungssystem*

$$\begin{aligned} y_0 &= \alpha \\ \frac{-y_{i+1} + 2y_i - y_{i-1}}{h^2} + q_i y_i &= g_i, \qquad i = 1, \ldots, n, \\ y_{n+1} &= \beta \end{aligned} \tag{9.61}$$

aufstellen. Dies führt auf die Normalform

$$\begin{aligned} Ay &= k \\ \text{wobei} \quad A &= \begin{pmatrix} 2+q_1h^2 & -1 & 0 & \ldots & 0 \\ -1 & 2+q_2h^2 & -1 & 0 & \ldots \\ 0 & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \ldots & 0 & -1 & 2+q_nh^2 \end{pmatrix} \\ y &= (y_1, y_2, \ldots, y_n)^T \\ k &= (h^2g_1 + \alpha, h^2g_2, \ldots, h^2g_{n-1}, h^2g_n + \beta)^T \end{aligned} \tag{9.62}$$

Dieses symmetrische Tridiagonalsystem ist positiv–definit, falls $q_i > 0$ *ist. Es kann mit einem für Tridiagonalmatrizen angepassten Choleskiverfahren mit der Komplexität O(n) gelöst werden.*

Satz 9.7 ***Fehlerabschätzung****: Die Randwertaufgabe* (9.61) *besitze eine viermal stetig differenzierbare Lösung y mit*

$$\left|y^{(4)}\right| \leq M \ \forall x \in [a,b], \quad \textit{und} \quad q(x) \geq 0, \tag{9.63}$$

so gilt

$$|y(x_i) - y_i| \leq \frac{Mh^2}{24}(x_i - a)(b - x_i) \tag{9.64}$$

Um den Diskretisierungsfehler abschätzen zu können, ist es sinnvoll, die Berechnung mit veränderten Schrittweiten $q \cdot h$ zu wiederholen, und den Unterschied zu bewerten, ggf. weiter zu verfeinern und die Ergebnisse y_i zu extrapolieren. Mit $q = 1/2$ gilt

$$\frac{1}{3}(4y_{2i}^{[qh]} - y_i^{[h]}) - y(x_i) = \mathcal{O}(h^4). \tag{9.65}$$

Liegt mit einer nicht-linearen Funktion $f(x, y)$ eine nicht–lineare DGL

$$\begin{aligned} -y''(x) + f(x,y) &= 0 \\ y(a) = \alpha \qquad y(b) &= \beta \end{aligned} \tag{9.66}$$

vor, so erhält man durch die Diskretisierung ein nicht-lineares Gleichungssystem

$$By + F(y) = 0, \tag{9.67}$$

mit einer nicht–linearen Vektorfunktion $F(y)$. Als Lösungsansatz kann das Newton-Verfahren genommen werden, oder ggf. andere, dem Problem angepasste Iterationsmethoden. Bei Konvergenz liegt dieselbe Fehlerordnung wie im linearen Fall vor.

Die Diskretisierung einer Eigenwertaufgabe

$$\begin{aligned} -y''(x) + \lambda q(x) y(x) &= 0 \\ y(a) = 0 \qquad y(b) &= 0 \end{aligned} \tag{9.68}$$

lautet nun

$$\begin{aligned} y_0 &= 0 \\ \frac{-y_{i+1} + 2y_i - y_{i-1}}{h^2} + \lambda q_i y_i &= 0, \qquad i = 1, \ldots, n, \\ y_{n+1} &= 0. \end{aligned} \tag{9.69}$$

Für $q_i = q(x_i) \neq 0$ erhält man die spezielle Eigenwertaufgabe

$$(A - \lambda I) y = 0 \tag{9.70}$$

$$\text{wobei} \quad A = \frac{1}{h^2} \begin{pmatrix} \frac{-2}{q_1} & \frac{1}{q_1} & 0 & \ldots & 0 \\ \frac{1}{q_2} & \frac{-2}{q_2} & \frac{1}{q_2} & 0 & \ldots \\ 0 & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \frac{1}{q_{n-1}} & \frac{-2}{q_{n-1}} & \frac{1}{q_{n-1}} \\ 0 & \ldots & 0 & \frac{1}{q_n} & \frac{-2}{q_n} \end{pmatrix}$$

$$y = (y_1, y_2, \ldots, y_n)^T$$

Die Lösungsverfahren sind im vorherigen Kapitel behandelt worden.

9.2.2 Schießverfahren

Bei Schießverfahren wird die Randwertaufgabe durch wiederholte Lösung einer Anfangswertaufgabe behandelt, bei der ein Anfangswertparameter so optimiert wird, dass die rechte Randbedingung erfüllt wird. Grundsätzlich kann man das vorgegebene Intervall $[a, b]$ bei Einfach-Schießverfahren "mit einem Schuß" überbrücken, sowie bei Mehrzielmethode das Intervall vorher aufteilen und mit den Teilintervallen entsprechende Rechnungen durchführen.

Das Randwertproblem

$$\begin{aligned} y'' &= f(x, y, y') \\ y(a) = \alpha, \qquad y(b) &= \beta \end{aligned} \tag{9.71}$$

wird als Anfangswertaufgabe

$$\begin{aligned} y'' &= f(x, y, y') \\ y(a) = \alpha, &\qquad y'(a) = s \end{aligned} \tag{9.72}$$

umgeschrieben. Der nun hinzugekommener Parameter s bewirkt eine zusätzliche Abhängigkeit der Lösung $y(x, s)$. Um der ursprünglichen Randwerdaufgabe zu genügen wird ein $\bar{s}$ gesucht mit

$$y(b, \bar{s}) = \beta. \tag{9.73}$$

Die Lösung kann gewonnen werden, wenn die Nullstelle der Funktion

$$F(s) := y(b, s) - \beta \tag{9.74}$$

gefunden wurde. Dies kann man z.B. mit einem modifizierten Newton–Verfahren erreichen, in dem die Ableitung $F'(s^{(i)})$ für jeden Newtonschritt i mittels Finiten Differenzen ermittelt wird, also die AWA (9.72) für $s = s^{(i)}$ und $s = s^{(i)}) + \Delta s^{(i)}$ gelöst wird. Damit lautet der Algorithmus bei gegebenen Startwert $s^{(0)}$

$$s^{(i+1)} = s^{(i)} - \gamma \left(\Delta F(s^{(i)})\right)^{-1} \cdot F(s^{(i)}) \quad \text{wobei} \tag{9.75}$$

$$\Delta F(s^{(i)}) := \frac{F(s^{(i)} + \Delta s^{(i)}) - F(s^{(i)})}{\Delta s^{(i)}}. \tag{9.76}$$

Dabei steuert γ die Schrittweite des Newton-Schrittes. Die AWA muss also pro Schritt zweimal gelöst werden.

Im Allgemeinen gilt jedoch das Einfach–Schießverfahren als nicht genau genug, insbesondere wenn das Intervall $[a, b]$ und die Lipschitzkonstante L groß sind. Man kann zeigen, dass der Fehler exponentiell wachsen kann

$$|y(x, s_1) - y(x, s_2)| \leq |s_1 - s_2| \exp(L(x - a)). \tag{9.77}$$

Mehrzielmethode

Ein Ausweg aus diesem Problem ist die Mehrzielmethode (multiple oder parallel shooting). Nach dem Grundsatz, dass kurze Schüsse zielsicherer sind, wird das Intervall nicht notwendigerweise äquidistant aufgeteilt

$$a = x_1 < x_2 < \ldots < x_{m-1} < x_m = b \tag{9.78}$$

$$\text{und} \quad r_k := y(x_k), \;\; s_k := y'(x_k), \;\; k = 1, \ldots, m. \tag{9.79}$$

Für jedes Teilintervall wird eine AWA gelöst

$$\begin{aligned} y'' &= f(x, y, y') \;\; \text{in} \;\; (x_k, x_{k+1}) \\ y(x_k) = r_k, &\qquad y'(x_k) = s_k \end{aligned} \tag{9.80}$$

mit der Teillösung $y(x, x_k, r_k, s_k)$. Der Vektor der Anfangswerte lautet nun

$$s := (r_1, s_1, r_2, s_2, \ldots, r_{m-1}, s_{m-1}, r_m, s_m)^T.$$

Man erhält nun ein Gleichungssystem mit $2m$ Gleichungen

$$\begin{aligned}
F_1(s) &:= y(x_2, x_1, r_1, s_1) - r_2 = 0 \\
F_2(s) &:= y'(x_2, x_1, r_1, s_1) - s_2 = 0 \\
&\vdots := \vdots \\
F_{2m-3}(s) &:= y(x_m, x_{m-1}, r_{m-1}, s_{m-1}) - r_m = 0 \\
F_{2m-2}(s) &:= y'(x_m, x_{m-1}, r_{m-1}, s_{m-1}) - s_m = 0 \\
F_{2m-1}(s) &:= r_1 - \alpha = 0 \\
F_{2m}(s) &:= r_m - \beta = 0
\end{aligned} \tag{9.81}$$

Wiederum wird die Lösung dieses nicht–linearen Gleichungssystems analog (9.75) ermittelt. Bei $m-1$ Teilintervallen hat man für jeden Newton–Schritt $3(m-1)$ AWA zu lösen, wobei 2 Lösungen zur Näherung der Funktionalmatrix $DF(s)$ aufgewandt werden (s.u.).

Der Algorithmus kann folgendermaßen zusammengefasst werden:

1. Wähle ein $s^{(0)} \in \mathbb{R}^{2m}$ und eine Genauigkeitstoleranz ϵ und setze $i := 0$.

2. Für k=1,2, …, m-1:
   berechne $y(x_{k+1}, x_k, r_k, s_k)$ und $y'(x_{k+1}, x_k, r_k, s_k)$ als Lösung der AWA
   $y'' = \lambda f(x, y, y') \qquad y(x_k) = r_k, \qquad y'(x_k) = s_k$ und dann $F(s)$.

3. Berechne eine Näherung ΔF der Funktionalmatrix DF durch Lösung der beiden AWA pro Teilintervall

$$\begin{aligned}
y'' &= f(x, y, y') \qquad y(x_k) = r_k + \delta r_k, \qquad y'(x_k) = s_k \\
y'' &= f(x, y, y') \qquad y(x_k) = r_k, \qquad y'(x_k) = s_k + \delta s_k
\end{aligned}$$

4. Löse das $2(m-1)$–dimensionale lineare Gleichungssystem $\Delta F \cdot \Delta s = -F$

5. Setze $s^{(i+1)} = s^{(i)} + \Delta s$.

6. Für $\left\| s^{(i+1)} - s^{(i)} \right\| < \epsilon$ Gehe nach 8. (Stop)

7. $i \leftarrow i + 1$, gehe nach 2.

8. Stop

Kapitel 10

Integralgleichungen

10.1 Klassifizierungen

Integralgleichungen treten in geophysikalischen Anwendungen oft bei der Auswertung von Fernerkundungsmethoden auf und führen dann auf Inverse Problemstellungen hin, nach einer Diskretisierung ähnlich wie die Normalgleichung. So gibt es eine Reihe unterschiedlicher Typen im unendlich–dimensionalen Funktionenraum, die ihre Entsprechungen in uns bereits bekannten diskreten Formen haben.

Man unterscheidet zunächst die *Fredholmschen Integralgleichungen*, bei denen feste Integrationsgrenzen gegeben sind. Eine *inhomogene Fredholmsche Integralgleichungen der 1. Art* lautet

$$g(t) = \int_a^b K(t,s)\, f(s)\, ds, \tag{10.1}$$

wobei die (bei diesem Thema meist links geschriebene) "rechte Seite" $g(t)$ die Inhomogenität bezeichnet und $f(s)$ die zu ermittelnde Unbekannte ist. In der Regel umfasst die "rechte Seite" $g(t)$ (oft fernerkundete) Messdaten. Die Funktion $K(t,s)$ wird Kern (kernel) genannt. Ist $K(t,s)$ invertierbar und $g(t) \neq 0$, so hat (10.1) eine eindeutige Lösung f.

Die oben erwähnte Analogie zu linearen Gleichungen lautet in der entsprechenden Schreibweise $Kf = g$, wobei K eine Rechteckmatrix sein kann. Fredholmsche Integralgleichungen der 1. Art sind oft schlecht konditioniert. Kleine Änderungen der gegebenen Funktionen ziehen also oft große Änderungen der Lösung nach sich. Dabei wirkt das Integral als Glättungsoperator, der Information "verschmiert". Wegen ihrer schlechten Kondition werden Fredholmsche Integralgleichungen der 1. Art meistens mit Methoden der Inversen Modellierung behandelt. Hierauf wird in Abschnitt 10.5 einführend eingegangen werden.

Die *Fredholmsche Integralgleichung der 2. Art*

$$f(t) = g(t) + \lambda \int_a^b K(t,s)\, f(s)\, ds \tag{10.2}$$

hat als Unbekannte wieder $f(s)$. Im Falle $g(t) = 0$ handelt es sich um eine homogene Problemstellung. Die diskrete Entsprechung ist die Eigenwertaufgabe $(K - \sigma I)f = g$. Allerdings entsprechen die Eigenwerte λ hier $1/\sigma$ und entsprechend ist auch die Funktion $g(t)$ durch λ zu dividieren. Ferner wird $f(s)$ als *Eigenfunktion* bezeichnet.

Schreibt man (10.2) mittels δ-Funktion als

$$\lambda \int_a^b \left(K(t,s) - \sigma\delta(t-s)\right)\, f(s)\, ds = -\sigma g(t), \tag{10.3}$$

und ist σ ausreichend groß, so ist der Kern diagonaldominant und das Problem entsprechend gut gestellt.

Gilt $K(t,s) = 0$ für $s > t$, hat man also eine variable Integrationsgrenze, so liegt eine Volterrasche Integralgleichung vor.

Die *Volterrasche Integralgleichung 1. Art* lautet damit

$$g(t) = \int_a^t K(t,s)\, f(s)\, ds. \tag{10.4}$$

und entspricht damit einer linearen Gleichung mit unterer Dreiecksmatrix. Die Lösung der Volterraschen Integralgleichung der 1. Art ist in der Regel ein gutgestelltes Problem.

Die Volterrasche Integralgleichung 2. Art lautet

$$f(t) = \int_a^t K(t,s)\, f(s)\, ds + g(t), \tag{10.5}$$

wobei wieder die Entsprechung zu einem Eigenvektorproblem mit einer unteren Dreiecksmatrix vorliegt .

Insgesamt hat man nun folgende Klassifizierungen:

- Integrationsgrenzen
  - beide fest: Fredholm Gleichung
  - eine variabel: Volterra Gleichung
- Plazierung der unbekannten Funktion
  - nur innerhalb des Integrals: 1 Art
  - innerhalb und außerhalb des Integrals: 2. Art
- Gegebene Funktion $g(t)$
  - identisch Null: homogen
  - nicht identisch Null: inhomogen

10.2 Fredholmsche Integralgleichungen der 2. Art

Zunächst bemühen wir uns um die numerische Lösung der Fredholmschen Integralgleichungen der 2. Art

$$f(t) = g(t) + \lambda \int_a^b K(t,s)\, f(s)\, ds. \tag{10.2}$$

Ein einfacher Ansatz wird mit dem Nystrom–Verfahren verfolgt, welches eine Quadraturregel anwendet, etwa die Gauß-Legendre-Quadratur, falls der Kern keine andere Wahl nahelegt

$$\int_a^b y(s)\, ds = \sum_{j=1}^{N} w_j y(s_j). \tag{10.6}$$

Wendet man diesen Ansatz auf (10.2) an, so erhält man

$$f(t) = \lambda \sum_{j=1}^{N} w_j K(t,s_j)\, f(s_j) + g(t). \tag{10.7}$$

Für die Auswertung an den Stützstellen des gewählten Quadraturverfahrens findet man dann

$$f(t_i) = \lambda \sum_{j=1}^{N} w_j K(t_i,s_j)\, f(s_j) + g(t_i). \tag{10.8}$$

Mit $f_i := f(t_i),\ g_i := g(t_i)$ und $\tilde{K}_{ij} = K(t_i,s_j)w_j$ kann man (10.8) als Matrixgleichung notieren

$$(I - \lambda\tilde{K}) \cdot f = g. \tag{10.9}$$

Hierbei wird zunächst davon ausgegangen, dass λ kein Eigenwert ist und auch nicht in der Nähe eines Eigenwertes liegt. Dann kann die übliche Dreieckszerlegung zur Lösung genutzt werden. Anderenfalls liegt ein schlecht gestelltes Problem vor. In diesem Fall sind andere Lösungsmethoden vorzuziehen.

Möchte man die Lösung an anderen Punkten als den durch die Quadratur vorbestimmten Stützstellen t_i ermitteln, so sollte man nicht den Lösungsvektor f_i interpolieren, da dies einen deutlichen Genauigkeitsverlust bewirkt. Stattdessen ist es angebracht, (10.7) zur Interpolation zu nutzen.

Im homogenen Fall mit $g = 0$ erhält man mit $\sigma = 1/\lambda$ die Standardeigenwertaufgabe

$$\tilde{K} \cdot f = \sigma f, \tag{10.10}$$

die in der nunmehr bekannten Weise gelöst werden kann. Idealerweise ist $\tilde{K}$ aus bekannten Gründen symmetrisch, da sonst eine obere Hessenbergmatrix

berechnert werden muss. Nach der Diskretisierung sind die dann notwendigen w_i zumeist nicht gleich, wodurch die Symmetrie durchbrochen wird. Mit $D = \mathrm{diag}(w_j)$ folgt aus $KD \cdot f = \sigma f$ nach Multiplikation mit $D^{1/2} = \mathrm{diag}(\sqrt{w_j})$

$$D^{1/2} K D^{1/2} h = \sigma h, \tag{10.11}$$

wobei $h := D^{1/2} f$. Damit ist die Symmetrie bewahrt. Die Diskretisierung mit N Stützstellen liefert ebenso viele Eigenwerte. Kerne von endlichem Rang (d.h. *degenerierte* oder *separable Kerne*) besitzen nur eine endliche Anzahl von Eigenwerten $\neq 0$. Diese Situation kann dadurch erkannt werden, dass Eigenwerte $= 0$ in Größenordnung der Rechnergenauigkeit um 0 liegen, und die Anzahl der Eigenwerte $\neq 0$ nach Erhöhung von N konstant bleibt.

10.3 Volterrasche Integralgleichungen

Von der Volterraschen Integralgleichung wird wiederum zunächst die Form 2. Art betrachtet

$$f(t) = \int_a^t K(t,s)\, f(s)\, ds + g(t). \tag{10.5}$$

In der Regel wird die Lösung mit $t = a$ begonnen und dann weiter fortgeschritten. Dieser Ansatz hat eine Analogie in den Anwangswertproblemen der gewhnlichen Differentialgleichungen. Im einfachsten Fall definiert man eine äquidistante Unterteilung des Intervalls

$$t_i := a + ih, \quad i = 0, 1, \ldots, N, \quad h := \frac{b-a}{N}. \tag{10.12}$$

Bei dieser Aufteilung ist die Trapezregel die einfachste numerische Integrationsmethode:

$$\int_a^{t_i} K(t,s)\, f(s)\, ds = h \left(1/2 K_{i0} f_0 + \sum_{j=1}^{i-1} K_{ij}\, f_j + 1/2 K_{ii} f_i \right). \tag{10.13}$$

Dies eingesetzt in (10.5) erhält man

$$\begin{aligned} f_0 &= g_0 \\ (1 - 1/2 h K_{ii}) f_i &= h \left(1/2 K_{i0} f_0 + \sum_{j=1}^{i-1} K_{ij}\, f_j \right) + g_i, \end{aligned} \tag{10.14}$$

also einen expliziten Ausdruck, ohne die Notwendigkeit, eine Gleichung zu lösen. Liegt eine Volterrasche Integralgleichung der 1. Art vor, so entfällt die linke "1".

In der Praxis kommen häufig Systeme von m Integralgleichungen vor, d.h. eine Vektorform von (10.5) ist zu lösen, wobei dann entsprechend $K(t,s)$ die Einträge einer $m \times m$–Matrix liefern.

10.4 Integralgleichungen mit Singularitäten

Häufig besteht das Problem, dass der Kern $K(t,s)$ oder die Lösung $f(s)$ der Integralgleichung, oder beide, Singularitäten aufweisen. Hebbare Singularitäten können durch eine Variablentransformation beseitigt werden, etwa in Falle einer Wurzel $K(t,s) \propto s^{\pm 1/2}$. Dies gelingt bei $s=0$ durch $z := s^{\pm 1/2}$. Nach dieser Transformation können dann geeignete Quadraturverfahren gewählt werden. Ein besonderer Fall stellt die singuläre δ–Funktion $K(t,s) = \delta(t-s)$ dar, die nach ihrer Integrationsregel der Einheitsoperator ist.

In einigen Fällen kann die Singularität durch Faktorisierung $K(t,s) = w(s)\bar{K}(t,s)$ isoliert werden. Das singuläre Gewicht $w(s)$ kann dann ggf. mittels der ihr angepassten Gaußquadratur behandelt werden.

Es können ferner Fälle auftreten, bei denen $K(t,s)$ bei $t=s$ auf einer Skala fast singulär ist, die viel kleiner ist als die Skala, auf der $f(t)$ variiert. In diesem Fall kann man in dem problematischen Bereich durch Polynominterpolation oder Splines die Genauigkeit erhöhen.

Bei unendlichen Integrationsgrenzen liegen dort ähnliche Probleme wie bei Singularitäten vor. Hier können bei ausreichend schneller Konvergenz z $\to 0$ mit der Gauß-Laguerre–Quadratur $w \propto \exp(-\alpha s)$ einseitig, oder der Gauß-Hermite–Quadratur $w \propto \exp(-s^2)$ zweiseitig Gewichtsfunktionen zur Faktorisierung eingeführt werden. In anderen Fällen mit $0 < s < \infty$ hilft etwa eine Transformation

$$s = \frac{2\alpha}{z+1} - \alpha, \tag{10.15}$$

die auf das Gauß–Legendre–Intervall $-1 < z < 1$ abbildet, und die Nutzung dieses Verfahren ermöglicht.

Ein weiterer typischer Problemfall, bei dem die Nystrom–Methode versagt, sind Singularitäten des Kernes $K(t,s)$ entlang der Diagonalen $t=s$. Eine Möglichkeit der Lösung bietet die Subtraktion der Singularität in der folgenden Weise

$$\begin{aligned}\int_a^b K(t,s)\, f(s)\, ds &= \int_a^b K(t,s)\ (f(s)-f(t))\ ds + \int_a^b K(t,s)\, f(t)\, ds \\ &= \int_a^b K(t,s)\ (f(s)-f(t))\ ds + r(t) f(t), \end{aligned} \tag{10.16}$$

mit $r(t) := \int_a^b K(t,s)\, ds$, welches entweder analytisch oder numerisch berechnet wird. Nun kann in der Regel die Nystrom-Methode angewandt werden. Man erhält dann anstelle (10.8)

$$f_i = \lambda \sum_{\substack{j=1 \\ j\neq i}}^{N} w_j K_{ij}\,(f_j - f_i) + \lambda r_i f_i + g_i. \tag{10.17}$$

10.5 Lösung schlecht-konditionierter Integralgleichungen

Zu Beginn dieses Kapitels wurde darauf hingewiesen, das insbesondere Fredholmsche Integralgleichungen der 1. Art oft sehr schlecht konditioniert sind oder gar unterbestimmt. Die oben genannten Verfahren versagen dann völlig. Grundsätzlich wird dann nach Information gesucht, welche geeignet ist, die Unterbestimmheit aufzuheben oder die Freiheitsgrade des Systems zu verringern. Man kann zum Beispiel letzteres erreichen, indem man gewisse Glättebedingungen erzwingt oder anderes Wissen über den Systemzustand einschließt.

Gegeben sei die Integralgleichung

$$c_i = \int r_i(x)u(x)dx + \epsilon_i, \tag{10.18}$$

wobei die c_i eine Menge von N Messungen ist und $u(x)$ ein das Messergebnis bestimmender physikalischer Prozess oder Zustand ist, den man ermitteln möchte, und die Indizierung i möglichst verschiedene Aspekte des Systems kennzeichnet. Der lineare Kern $r_i(x)$ vermittelt zwischen Messung i und dem Prozess oder Zustand, der bestimmt werden soll. Der Fehler hierzu wird mit ϵ_i bezeichnet.

Man kann nun diese Integralgleichung quadrieren und $u(x)$ so variieren, dass der Ausdruck

$$\left(c_i - \int r_i(x)u(x)dx\right)^2 =: \mathcal{A}(u), \tag{10.19}$$

also das positiv definite Funktional $\mathcal{A}(u)$ minimal wird.

Die angenommene schlechte Kondition oder gar Unterbestimmtheit des Problems erzwingt nun die Hinzunahme zusätzlicher Information in Form eines Funktionals $\mathcal{B}(u)$, welches als Eindeutigkeit gewährende *Zwangsbedingung* oder *a priori* Information einen bestimmten Wert, etwa b, annehmen soll. Man erhält somit die Variationsaufgabe

$$\frac{\delta}{\delta u}\left(\mathcal{A}(u) + \lambda(\mathcal{B}(u) - b)\right) = \frac{\delta}{\delta u}\left(\mathcal{A}(u) + \lambda\mathcal{B}(u)\right) = 0. \tag{10.20}$$

Dabei ist $0 \leq \lambda < \infty$ ein Lagrangescher Multiplikator, der die Lage des Minimums durch seine Gewichtung bestimmt. Kleine λ unterdrücken die Zwangsbedingung und lassen den Einfluss der Daten zur Geltung kommen, zum Preis das die Unterbestimmtheit zu unrealistischen Varianzen oder unruhigen Lösungen führen kann. Andererseits kann im gegenteiligen Fall die Zwangsbedingung einen die Dateninformation überragenden Beitrag leisten und durch überbewertete Glattheit die Messinformation unterdrücken. Ist also etwa

$$\mathcal{A}(u) := |Au - c|^2 \tag{10.21}$$

als Funktional gegeben mit einem degenerierten oder singulären A, so erhält man nur nach gewichteter Addition eines regulären $\mathcal{B}(u)$ eine eindeutige Lösung.

Gesucht ist nun die zur "wahren" Lösung $u(x)$ genäherte und diskretisierte Lösungsfungtion $\hat{u}(x_\mu)$, wobei angenommen wird, dass die Diskretisierung von x mit M Stützstellen fein genug ist, um nur kleine Unterschiede $u(x_\mu), u(x_{\mu\pm1})$ zu ermöglichen. Eine Diskretisierung von (10.18) lautet

$$c_i = \sum_\mu R_{i\mu} u(x_\mu) + \hat{\epsilon}_i, \tag{10.22}$$

wobei die $N \times M$–Matrix R etwa die Komponenten

$$R_{i\mu} := r_i(x_\mu)(x_{\mu+1} - x_{\mu-1})/2 \tag{10.23}$$

enthalten kann.

Eine Möglichkeit $\mathcal{A}(u)$ zu definieren ist nun

$$\mathcal{A} = \chi^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(c_i - \sum_\mu R_{i\mu} u(x_\mu) \right) S_{ij}^{-1} \left(c_j - \sum_\mu R_{j\mu} u(x_\mu) \right). \tag{10.24}$$

Hierbei wurde mit der Fehlerkovarianzmatrix $S_{ij} := Cov(\hat{\epsilon}_i, \hat{\epsilon}_j)$ skaliert (siehe MATHMET 2). Wegen der schlechten Kondition oder Unterbestimmtheit der Minimierungsaufgabe (10.24) können Verfahren der Ausgleichsrechnung mit der Normalgleichung nicht verwandt werden. Eine andere Möglichkeit ist die Anwendung der Singulärwertanalyse. Neben einer Anzahl von ungeeigneten Lösungen wird ein $|\hat{u}|$ mit einem minimalen $\sum_\mu |\hat{u}(x_\mu)|$ ermittelt werden. Diese Lösung ist ein Ergebnis von (10.20) mit $\lambda = 0$. Ist $M >> N$, so werden alle Datenpunkte in unrealistische naher Weise durch die Lösung angepasst werden können, indem sich an die üblicherweise fehlerbehafteten Daten c_i angepasst wurde. Dies vermeidet man, indem in geeigneter Weise $\lambda > 0$ gewählt wird. Die Aufgabe lautet nun

$$\text{minimiere} \quad \chi^2(\hat{u}) + \lambda(\hat{u} \cdot \hat{u}). \tag{10.25}$$

Der Zusatzterm erlaubt nun die Kontrolle der Glätte der Lösungsfunktion. Die Aufgabe wird als *Inverses Problem mit Regularisierung 0. Ordnung* bezeichnet.

Kapitel 11

Partielle Differentialgleichungen

11.1 Übersicht

Die numerische Lösung partieller Differentialgleichungen ist oft die Kernaufgaben bei naturwissenschaftlichen und technischen Anwendungen, insbesondere aber im Bereich der Geophysik und Meteorologie. Allgemein hat man folgende Aufgabenstellung:
Gesucht ist eine m-vektorwertige Funktion $u(x) : \mathbb{R}^d \to \mathbb{R}^m$ mit $x := (x_1, x_2, \ldots, x_d)^T \in \Omega \subset \mathbb{R}^d$, so dass

$$F\left(x, u(x), \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_d}, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, \ldots, \frac{\partial^p u}{\partial x_d^p}\right) = 0, \tag{11.1}$$

wobei $d > 1$ die Dimension des Gebietes der fortan unabhängigen vektorwertigen Variablen x, und p die Ordnung der Differentialgleichung ist. Zumeist bezeichnet $d = 2$ bis 4 die Raum- und Zeitdimension, also etwa $d = 4$ bezeichnet ein 3–dimensionales raum–zeitliches Problem. Ist die Zeit explizit enthalten, so ist das Problem instationär, andernfalls stationär. Eine grobe Characterisierung ist etwa

- Gleichgewichtsprobleme bezeichnen stationäre Zustände,
- Ausbreitungsvorgänge bezeichnen instationäre Zustände,
- charakteristische Systemzustände werden durch Eigenwertprobleme beschrieben.

Zunächst beschränken wir uns auf lineare partielle DGLen 2. Ordnung ($p = 2$), mit einer zu bestimmenden skalaren Funktionn u ($m = 1$), welche auf einem 2-dimensionalen Gebiet definiert ist ($d = 2$). Dies erlaubt die suggestivere Schreibweise $x := x_1,\ y := x_2$, und $u_{xy} := \frac{\partial^2 u}{\partial x \partial y}$. Dann lauten DGLen, die wir betrachten wollen

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu = G \qquad \forall (x, y) \in \Omega. \tag{11.2}$$

Definition 11.1 *Eine lineare partielle DGL 2. Ordnung* (11.2) *mit* $A^2+B^2+C^2 \neq 0$ *heißt*

$$\textit{elliptisch, falls } AC - B^2 > 0 \quad \forall (x,y) \in \Omega, \tag{11.3}$$

$$\textit{parabolisch, falls } AC - B^2 = 0 \quad \forall (x,y) \in \Omega, \tag{11.4}$$

$$\textit{hyperbolisch, falls } AC - B^2 < 0 \quad \forall (x,y) \in \Omega. \tag{11.5}$$

Zur eindeutigen Lösung sind noch Bedingungen auf dem Rand $\partial\Omega$ des Gebietes Ω zu erfüllen. Diese *Randbedingungen* können unter anderem folgende, am häufigste vorkommende Formen annehmen

$$\text{Dirichletsche Randbedingung } u(x,y) := \phi(x,y) \qquad \text{auf } \partial\Omega_1 \subset \Omega \tag{11.6}$$

$$\text{v. Neumannsche Randbedingung } \frac{\partial u(x,y)}{\partial n} := \gamma(x,y) \qquad \text{auf } \partial\Omega_2 \subset \Omega \tag{11.7}$$

$$\text{Cauchysche Randbedingung } \frac{\partial u(x,y)}{\partial n} + \alpha u(x,y) := \beta(x,y) \qquad \text{auf } \partial\Omega_3 \subset \Omega \tag{11.8}$$

Hier sind α, ϕ, γ Funktionen, die die *Randwerte* vorgeben. Ferner ist $\frac{\partial u(x,y)}{\partial n}$ die Ableitung auf dem Rand in Richtung der äußeren Normalen.

Zusätzlich zu den Randbedingungen sind im Falle hyperbolischer und parabolischer partielle Differentialgleichungen noch Anfangswerte für den Zeitpunkt $t = 0$ auf Ω erforderlich. Die vorgenannten Bedingungen sind notwendige, jedoch nicht hinreichende Voraussetzungen für die Existenz und Eindeutigkeit von Lösungen.

Laplace- und Poissongleichung

Zur vereinfachten Notation definiert man den Laplaceoperator

Definition 11.2

$$\Delta u := u_{xx} + u_{yy}. \tag{11.9}$$

Die Poissongleichung lautet nun

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega \\ u &= g && \text{auf } \partial\Omega. \end{aligned} \tag{11.10}$$

Gleichung (11.10) beschreibt physikalisch eine Potentialgleichung. Dabei ist die Lösung u das Potentialfeld, während die "Quelle" f häufig eine Massendichte oder elektrische Ladungsdichte oder eine Potentialströmung(skomponente) beschreiben kann. Für $f = 0$ wird (11.10) oft auch als Laplacegleichung bezeichnet. Eine nichttriviale Lösung ist hier nur durch entsprechende Randwerte und ihre Geometrie gegeben. Der Verlauf des Potentialfeldes kann dann zumeist als "Minimalfläche"interpretiert werden.

Diffusions- und Wärmeleitungsgleichung

Bei zeitabhängiger Wärme– oder Konzentrationsverteilung in einer diffusiven Umgebung erhält man eine parabolische Gleichung. Ist κ etwa die räumlich durchaus variable Wärmeleitfähigkeit oder ein Diffusionskoeffizient, so lautet die entsprechende Differentialgleichung

$$\begin{aligned} \Delta u(x,y) &= \frac{1}{\kappa}\frac{\partial u}{\partial t} && \text{in } \Omega,\ \forall\, t>0 \\ u(x,y,0) &= u_0(x,y) && \text{in } \Omega \\ u(x,y,t) &= g(x,y), && (x,y)\in\partial\Omega,\ \forall\, t>0, \end{aligned} \tag{11.11}$$

wobei $g(x,y)$ die Randwerte bestimmt.

Wellengleichung

Schwingungsvorgänge wie etwa bei Druckschwankungen, mechanische (also auch atmosphärische und ozeanische) sowie elektromagnetische Wellen kann man durch Wellengleichungen darstellen. Die prototypische hyperbolische Wellengleichung lautet

$$\begin{aligned} \Delta u(x,y) &= \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} \quad \text{in } \Omega,\ \forall\, t>0 \\ u(x,y,0) &= u_0(x,y) \quad \text{in } \Omega \\ u_t(x,y,0) &= u_1(x,y) \quad \text{in } \Omega \\ u(x,y,t) &= g(x,y), \qquad (x,y)\in\partial\Omega,\ \forall\, t>0, \end{aligned} \tag{11.12}$$

11.2 Diskretisierung elliptischer Probleme

Am Beispiel der elliptischen Differentialgleichung (11.10) soll mittels des Differenzenverfahrens ein numerisches Lösungsverfahren vorgestellt werden. Hierzu wird das 2–dimensionale Gebiet Ω mit einem regelmäßigem Gitter quadratischer Maschen überdeckt werden. Der ggf. unregelmäßige Rand soll durch eine ausreichend feine Maschenweite h mit $N \times M$ Gitterpunkten approximiert werden. Die Gitterpunkte (x_i, y_j) werden dabei nach folgender Regel indiziert

$$x_i := x_0 + i \cdot h, \qquad i = 0,1,\ldots,N, \tag{11.13}$$

$$y_j := y_0 + j \cdot h, \qquad j = 0,1,\ldots,M, \tag{11.14}$$

$$\Omega \subset [x_0, x_N] \times [y_0, y_M] \qquad .$$

Die numerische Lösung von (11.10) besteht nun aus der Ermittlung von

$$u_{ij} \approx u(x_i, y_j), \qquad (x_i, y_j) \subset \Omega \setminus \partial\Omega. \tag{11.15}$$

Eine Diskretisierung führt nun von einer partiellen Differentialgleichung auf ein lineares Gleichungssystem mittels der Approximationen der Differentialoperatoren durch finite Differenzen. Eine einfache Diskretisierung lautet

$$u_{xx} \approx \frac{1}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) \tag{11.16}$$

$$u_{yy} \approx \frac{1}{h^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}). \tag{11.17}$$

Mit $f_{ij} := f(x_i, y_j)$ erhält man nun die Gleichung

$$-u_{i+1,j} - u_{i,j+1} + 4u_{i,j} - u_{i-1,j} - u_{i,j-1} = h^2 f_{ij} \qquad \forall (x_i, y_j) \subset \Omega \setminus \partial\Omega. \tag{11.18}$$

Dies führt auf ein sehr schwach besetztes lineares Gleichungssystem, wie es bei der Lösung partieller Differentialgleichungen typisch ist.

11.3 Quasilineare partielle Differentialgleichungen 1. Ordnung

11.3.1 Charakteristiken

Wir beschränken uns nun auf *quasilineare* partielle Differentialgleichung 1. Ordnung. In zweidimensionaler Form lauten sie

$$a(x, y, u(x, y))u_x + b(x, y, u(x, y))u_y = c(x, y, u(x, y)), \tag{11.19}$$

mit $(x, y) \in \Omega$, wobei $\Omega \subset \mathbb{R}^2$ ein einfach zusammenhängendes Gebiet ist. Ferner sei $a, b, c, u \in \mathcal{C}^1(\Omega)$ einmal stetig differenzierbar und $a^2 + b^2 \neq 0$ auf Ω. Dabei bezeichnen die Indizierungen x und y wie oben eingeführt die partiellen Ableitungen in den entsprechenden Richtungen.

Sind die Funktionen a und b nicht von $u(x, y)$ abhängig, so liegt eine lineare *partielle Differentialgleichung* vor.

Man erhält eine anschauliche Deutung der Differentialgleichung, wenn sie in folgender Vektorform geschrieben wird (Abb. 1):

$$(a, b, c) \cdot \nabla_{(x,y,-u)} u = 0. \tag{11.20}$$

Andererseits gilt aber auch, wegen der angenommenen stetigen Differenzierbarkeit auf einem einfach zusammenhängendem Gebiet, für u die Darstellung als vollständiges Differential

$$du = u_x dx + u_y dy \tag{11.21}$$

Dies kann man zusammen mit der Differentialgleichung in der Form (11.20) in die Vektorgleichung

$$\begin{pmatrix} dx & dy \\ a & b \end{pmatrix} \cdot \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} du \\ c \end{pmatrix} \tag{11.22}$$

überführen.

Hierbei liefert (11.21) keinerlei zusätzliche Information, (sondern ist nur eine allgemeinere Schreibweise). Daher verschwindet die Determinante und impliziert die Lösungsfläche (*Integralfläche*) als zweidimensionale Mannigfaltigkeit.

Also

$$\frac{dy}{dx} = \frac{b(x,y,u)}{a(x,y,u)}. \tag{11.23}$$

Die beiden anderen zu (11.22) alternativen Schreibweisen führen auf $\frac{du}{dx} = \frac{c(x,y,u)}{a(x,y,u)}$ oder $\frac{du}{dy} = \frac{c(x,y,u)}{b(x,y,u)}$.

Die Lösung der ersten Gleichung heißt *charakteristische Grundkurve*. Die zweite und die dritte Gleichung bestimmen mit die *charakteristische Kurve*.

Dies erlaubt aber eine Parametisierung der Variablen x, y und u mit einer unabhängigen Variablen s, so daß man nunmehr ein System von drei gewöhnlichen Differentialgleichungen erhält:

$$\frac{dx(s)}{ds} = a, \qquad \frac{dy(s)}{ds} = b, \qquad \frac{du(s)}{ds} = c. \tag{11.24}$$

Man kann jetzt zeigen, daß folgendes gilt: [1]

Satz 11.1 *Jede charakteristische Kurve, welche einen Punkt mit der Integralfläche gemeinsam hat, liegt ganz auf der Integralfläche.*

Jede Integralfläche wird aus einer einparametrigen Schar von charakteristischen Kurven erzeugt.

Eine eindeutige Problemstellung erhält man aber erst, wenn zusätzlich eine weitere parametrische Kurve, nämlich die Raumkurve der Anfangswerte

$$C := \begin{pmatrix} x_0 \\ y_0 \\ u_0 \end{pmatrix} := \begin{pmatrix} \varphi(r) \\ \psi(r) \\ \chi(r) \end{pmatrix} \tag{11.25}$$

gegeben ist, die die Lösungsfläche berandet. Es handelt sich dann um eine Anfangswertaufgabe (AWA), da die Kurve C die Anfangswerte der gewöhnlichen Differentialgleichungen (11.23) liefert.

In den hier behandelten Problemen kann man x mit der Zeit t identifizieren. Für die AWA ist dann $\varphi(r) = t, \psi(r) = 0$ und $u_0 = u_0(\varphi(r), 0)$.

Allgemein erwartet man von einer AWA, daß sie *sachgemäß gestellt (properly posed)* ist, d.h.,

1. es existiert eine Lösung,
2. diese Lösung ist eindeutig,
3. die Lösung muß in stetiger Weise von den Anfangswerten abhängen.

[1] zum Beweis siehe z.B. Courant, Hilbert II.

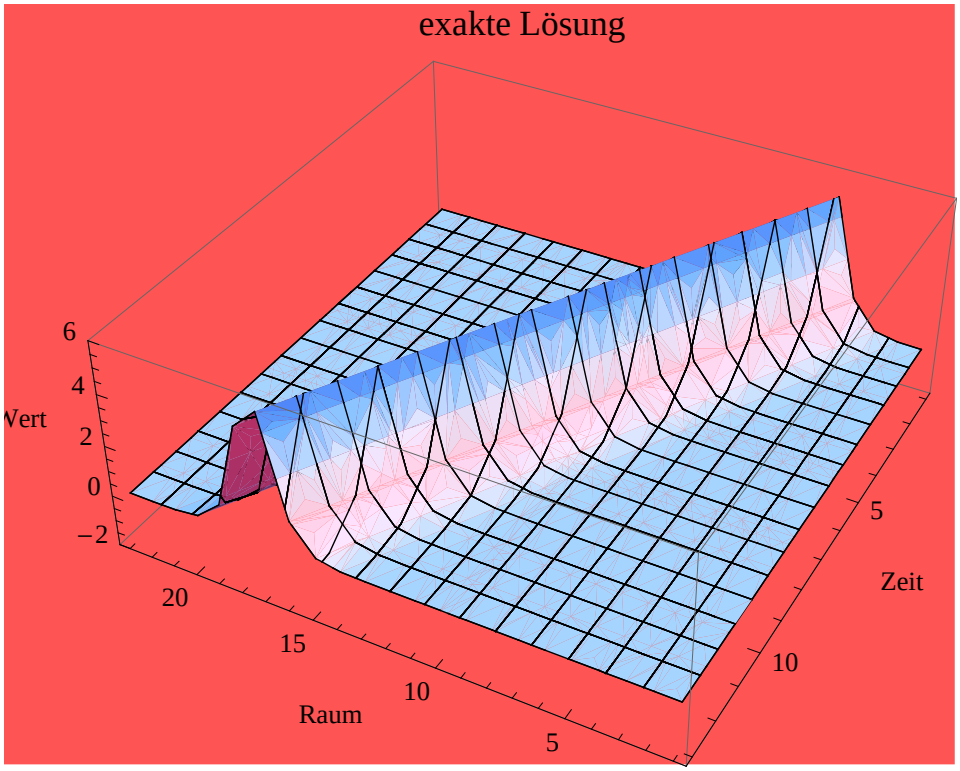


Abbildung 11.1: Raum-zeitliche Darstellung der exakten Lösung der Advektionsgleichung für eine Welle mit der Geschwindigkeit 1.

11.3.2 Diskretisierung der 1-D Advektionsgleichung

Die Advektionsgleichung, deren numerische Lösungen in den folgenden Kapiteln diskutiert werden soll, ist ein Sonderfall der quasilinearen partiellen Differentialgleichung 1. Ordnung. Die *lineare* Advektionsgleichung

$$\frac{\partial \chi(t,\mathbf{r})}{\partial t} + \mathbf{v}(t,\mathbf{r}) \cdot \nabla \chi(t,\mathbf{r}) = g(t,\mathbf{r}) \tag{11.26}$$

beschreibt in ihrer allgemeinen Form den Transport mit der Geschwindigkeit $\mathbf{v}$, die Erzeugung und Vernichtung g einer materiellen Eigenschaft oder Masse eines Spurenstoffes, Fluides oder Verbindung $\chi(t,\mathbf{r})$ in raum-zeitlicher $(t,\mathbf{r})$ Abhängigkeit. Wesentliches zur Lösung findet man aber schon im 1-dimensionalen homogenen Fall, auf den wir uns für das Weitere beschränken.

Lokaler Fehler und Konsistenz

Der Konvention folgend schreiben wir die 1-dimensionale homogene lineare Advektionsgleichung für die Lösung $u(t,x)$ in der Form

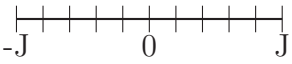
$$u_t + cu_x = 0, \tag{11.27}$$

wobei $u(t,x) \in \mathcal{C}1(\Omega), c$ nun vereinfachend eine konstante Geschwindigkeit, $(t,x) \in (0,T] \times [-1,1] =: \Omega, T$ ein fester, aber beliebig wählbarer Zeitpunkt ist. Mit der Anfangsbedingung $u(0,x) = u_o(x)$ zum Zeitpunkt $t = 0$ lautet die allgemeine Lösung

$$u(t,x) = u(x - ct). \tag{11.28}$$

Die Geraden $x - ct$ = const. sind die charakteristischen Grundkurven der Gleichung (11.20) in der (t,x)-Ebene.

Räumliche Diskretisierungen Es wird ab jetzt verlangt, daß $u \in \mathcal{C}^3(\Omega)$. Definiere Gitterpunkte x_j äquidistant auf $[-1,1]$ mit $x_j = j \cdot h, \quad j = -J, \ldots, J$, $h := 2/(2J)$ und $u_j(t) := u(t, j \cdot h)$.



Eine mögliche Diskretisierung der räumlichen Ableitung u_x sind die *zentrierten Differenzen*:

$$u_x|_{x_j} = \frac{u_{j+1} - u_{j-1}}{2h} + \frac{h^2}{12}\left(u_{xxx}|_{\xi_+} + u_{xxx}|_{\xi_-}\right) \tag{11.29}$$

mit $\xi_+ \in (x_j, x_{j+1})$ und $\xi_- \in (x_{j-1}, x_j)$, denn nach Taylor gilt

$$u_{j\pm1} = u_j \pm h u_x|_{x_j} + \frac{h^2}{2} u_{xx}|_{x_j} \pm \frac{h^3}{6} u_{xxx}|_{\xi_\pm} \tag{11.30}$$

Die Näherung (11.29) ist also von quadratischer Ordnung $\mathcal{O}(h^2)$.

Eine Variante von nur linearer Ordnung heißt *upstream*-Diskretisierung (weil die 2. Stützstelle gegen die "Stomrichtung" von c ausgestellt ist)

$$u_x|_{x_j} = \frac{u_j - u_{j-1}}{h} + \frac{h}{2} u_{xx}|_\xi, \qquad \xi \in (x_{j-1}, x_j). \tag{11.31}$$

Zeitliche Diskretisierung Von dieser Gestalt ist auch die zeitliche Vorwärtsdiskretisierung ("forward in time"), wenn man $t_n := n \cdot \Delta t, u_j^n := u(n \cdot \Delta t, j \cdot h)$ setzt, denn

$$u_t|_{tn} = \frac{u_j^{n+1} - u_j^n}{\Delta t} - \frac{\Delta t}{2} u_{tt}|_\eta, \qquad \eta \in (t_n, t_{n+1}). \tag{11.32}$$

Sie ist also auch nur von linearer Ordnung $\mathcal{O}(\Delta t)$.

Finite Differenzengleichung Gleichung (11.26) lautet also in der *finiten Differenzennäherung* nach den Ansätzen (11.29) und (11.32)

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{c}{2h}(u_{j+1}^n - u_{j-1}^n) = \tau_j^n, \tag{11.33}$$

wobei der lokale Fehler (Abschneidefehler, *truncation error*) τ_j^n von der Ordnung $O(\Delta t) + O(h^2)$ ist.

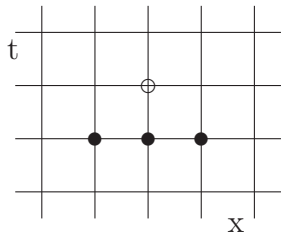


Abbildung 11.2: Diskretisierungsschema des Eulerverfahrens, volle Punkte bezeichnen bekannte Gitterwerte, der offene Punkt wird im nächsten Zeitschritt berechnet.

Wegen $u \in \mathcal{C}^3(\Omega)$ gibt es die oberen Schranken $M_t := \max_{t\in(t_n,t_{n+1})} |u_{tt}| < \infty$ und $M_x := \max_{x\in(x_{j-1},x_{x+1})} |u_{xxx}| < \infty$, so daß der totale Fehler abschätzbar wird

$$\tau_j^n \leq \frac{\Delta t}{2} M_t + |c| \frac{h^2}{12} M_x. \tag{11.34}$$

Def.: Ein Differenzenverfahren heißt *konsistent*, wenn gilt

$$\lim_{\substack{\Delta t\to 0 \\ h\to 0}} \tau_j^n = 0. \tag{11.35}$$

Dies trifft für den gerade gemachten Ansatz auch zu, wie man sofort einsieht.

Ein Lösungsansatz nach der bei den gewöhnlichen Differentialgleichungen schon kennengelernten Eulermethode für das numerische Lösungsverfahren mit der diskreten Näherungslösung U könnte nun nach (11.33) lauten:

$$U_j^{n+1} = U_j^n - \frac{\lambda}{2}(U_{j+1}^n - U_{j-1}^n) \tag{11.36}$$

$\lambda = \frac{c\cdot\Delta t}{h}$ bezeichnet hierbei die *Courant-Zahl* . Dieser Algorithmus wurde anhand eines glatten Signals getestet (Abb. 11.3.2).

Das Raum-Zeit-Diagramm zeigt aber trotz der nachgewiesenen Konsistenz sowohl eine Verstärkung des Signals als auch wachsende Verwerfungen stromaufwärts. Es tritt also eine wachsende Fehlerfortpflanzung auf (Labilität) und das Verfahren ist daher völlig unbrauchbar.

11.4 Globaler Fehler und Konvergenz

Gesucht ist nun eine Fehlerabschätzung, die sein Wachstum verständlich macht. Man vergleicht (11.33) und (11.36) und findet für den globalen Fehler (*accumulated error*)

$$e_j^{n+1} = u_j^{n+1} - U_j^{n+1} = e_j^n - \frac{\lambda}{2}(e_{j+1}^n - e_{j-1}^n) + \Delta t \tau_j^n. \tag{11.37}$$

Die Vorausetzung $u \in \mathcal{C}^3(\Omega)$ erlaubt wiederum

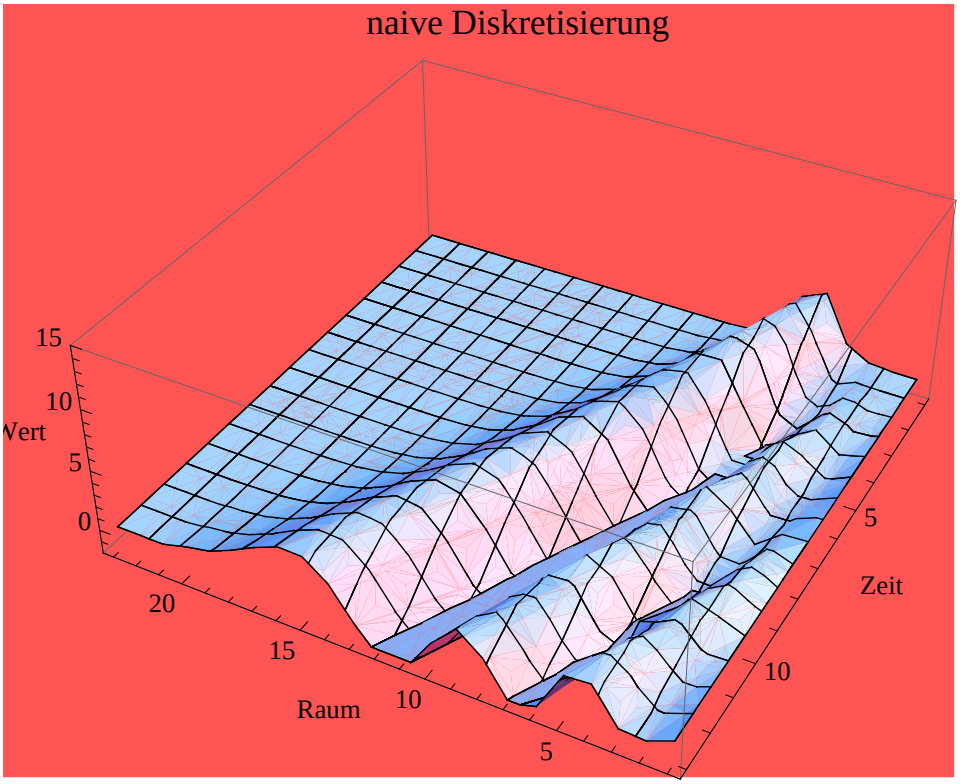


Abbildung 11.3: Raum-zeitliche Darstellung der Advektionsgleichung in der naiven, labilen Diskretisierung (11.36) für ein isoliertes, aber stetiges Anfangssignal.

$\epsilon^n = \max_j |e_j^n| < \infty$ zu setzen und $\tau := \max_{j,n} |\tau_j^n| < \infty$, so daß nunmehr mit $\epsilon^0 = 0$ gilt:

$$\epsilon^{n+1} \leq (1 + |\lambda|)\epsilon^n + \Delta t\tau = \left(1 + (1 + |\lambda|) + (1 + \lambda)^2 + \ldots + (1 + |\lambda|)^n\right) \Delta t\tau. \tag{11.38}$$

Also gilt mit der geometrischen Reihe entwickelt nach $1 + |\lambda|$

$$\epsilon^n = \Delta t\tau \sum_{k=0}^{n-1} (1 + |\lambda|)^k = \frac{\Delta t\tau}{|\lambda|} \left((1 + |\lambda|)^n - 1\right) = \frac{\tau h}{|c|} \left(\left(1 + \frac{|c|t}{nh}\right)^n - 1\right). \tag{11.39}$$

Das Verhalten der oberen Fehlerschranke bei fortschreitender Integration ist nun

$$\lim_{n\to\infty} \epsilon^n = \frac{\tau}{|c|} h \left(\exp\left(\frac{|c|t}{h}\right) - 1\right), \tag{11.40}$$

also exponential, d.h. ungünstiger geht es kaum.

Man kann nun auf die Idee kommen, in (11.36) einen den jeweiligen Startpunkt U_j^n dämpfenden Ersatzausdruck einzusetzen. Wir versuchen es einmal

mit der glättenden Mittelung

$$\frac{1}{2}(u_{j+1}^n + u_{j-1}^n) = u_j^n + \frac{h^2}{2}(u_{xx}|_{\xi_+} + u_{xx}|_{\xi_-}) = u_j^n + O(h^2). \tag{11.41}$$

Dieser Ersatzterm ist konsistent, denn mit der gleichen Argumentation wie oben dürfen wir auf die Existenz einer oberen Schranke M_s für $|u_{xx}|$ vertrauen, wenn wir bei der Grenzwertbildung Δt mit h über ein festes λ koppeln

$$\lim_{h\to 0} \tau_j^n = \frac{h^2}{2\Delta t} M_s = \frac{|c|h}{2\lambda} M_s = 0, \quad \lambda \,\text{fest}. \tag{11.42}$$

Nach Klärung dieser notwendigen Vorraussetzung prüfen wir den globalen Fehler.

Also

$$e_j^{n+1} = u_j^{n+1} - U_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{\lambda}{2}(u_{j+1}^n - u_{j-1}^n) + \Delta t \tau_j^n \tag{11.43}$$

$$- \frac{1}{2}(U_{j+1}^n + U_{j-1}^n) - \frac{\lambda}{2}(U_{j+1}^n - U_{j-1}^n) \tag{11.44}$$

$$= (\frac{1}{2} + \frac{\lambda}{2}) e_{j-1}^n + (\frac{1}{2} - \frac{\lambda}{2}) e_{j+1}^n + \Delta t \tau_j^n \tag{11.45}$$

$$\epsilon^{n+1} \leq (|\frac{1}{2} + \frac{\lambda}{2}| + |\frac{1}{2} - \frac{\lambda}{2}|)\epsilon^n + \Delta t \tau_j^n \tag{11.46}$$

$$\leq \epsilon^n + \Delta t \tau. \tag{11.47}$$

Mithin $\epsilon^n \leq n\Delta t\tau = t\tau$.

Der globale Fehler ist also proportional zur Integrationsdauer und zum lokalen Fehlermaximum. Das ist ein vernünftiges Fehlerverhalten. Es ist also folgende Definition sinnvoll:

Definition 11.3 *Ein finites Differenzenschema heißt konvergent, wenn mit* $|\lambda| < 1$ *gilt*

$$\lim_{\substack{\Delta t\to 0 \\ h\to 0}} U_j^n = u(t,x) \qquad \text{für} \quad t \in (0,T] \quad \text{fest}. \tag{11.48}$$

Abb. 11.4 stellt nun das konvergente Differenzverfahren

$$U_j^{n+1} = \frac{1}{2}(U_{j+1}^n + U_{j-1}^n) - \frac{\lambda}{2}(U_{j+1}^n - U_{j-1}^n) \tag{11.49}$$

dar.

Aber man sieht sofort, daß hier zuviel des Guten kuriert worden ist. Das Signal dissipiert zunehmend. Das Verfahren ist leider auch unbrauchbar.

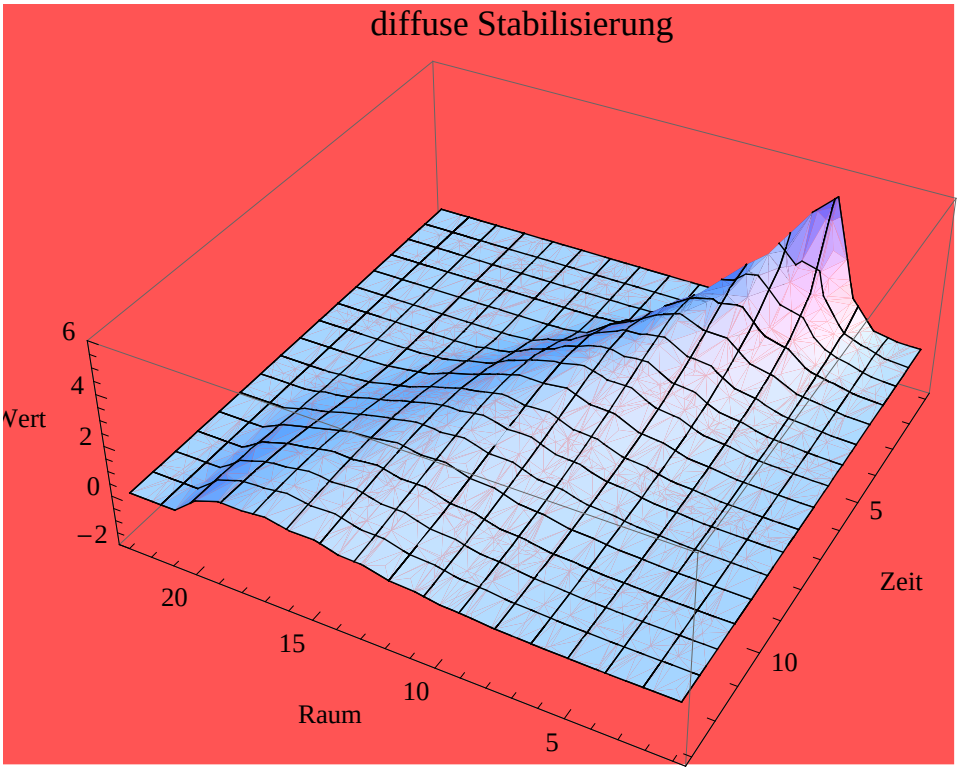


Abbildung 11.4: Wie Abb. 11.1 nur raum-zeitliche Darstellung der "stabilisierten" Advektionsgleichung mit diffusiv wirkender Mittelung (11.49).

11.4.1 Das Lax-Richtmyer-Theorem

Im allgemeinen ist es bei höheren Verfahren einfacher, die Stabilität nachzuweisen als Konvergenz. Dies ist aber unter gewissen Voraussetzungen äquivalent, wie gezeigt werden soll.

Zunächst etwas zur Begriffsklärung:

Sei $\{(\mathbf{L}(\Delta t))^n \mid n \in \mathbf{N}\}$ eine Menge linearer Operatoren.

Definition 11.4 *Eine Menge von Operatoren ist gleichmäßig beschränkt, wenn gilt* $||\mathbf{L}^n(\Delta t)\mathbf{U}|| \leq \alpha||\mathbf{U}||, \alpha > 0$ *für alle* $n \in I\!N$.

Es reicht für unsere Zwecke, sich diese Operatoren als Matrizen vorzustellen.

Definition 11.5 *Ein Differenzverfahren heißt stabil, wenn es ein* $\eta > 0$ *und ein* $\epsilon > 0$ *gibt, und die Menge der Operatoren* $\{(\mathbf{L}(\Delta t))^n | \Delta t < \eta h, n\Delta t \leq T, h < \epsilon\}$ *mit* $n \in I\!N$, *gleichmäßig beschränkt ist.*

Es gilt der:

Satz 11.2 *(Lax-Richtmyer) Ein konsistentes Differenzenverfahren zu einer sachgemäß gestellten Anfangswertaufgabe ist genau dann konvergent, wenn es stabil ist.*

Wir wollen den Beweis nur in der Richtung stabil $\Longrightarrow$ konvergent skizzieren[2].

Allgemein können wir das Differenzverfahren für eine inhomogene Gleichung in Vektorschreibweise notieren :

$$\mathbf{U}^{n+1} = \mathbf{L}\mathbf{U}^n. \tag{11.50}$$

Das labile Verfahren (11.36) z.B. lautet dann mit periodischen Randbedingungen:

$$\begin{pmatrix} U_{-J}^{n+1} \\ U_{-J+1}^{n+1} \\ \vdots \\ \vdots \\ \vdots \\ U_J^{n+1} \end{pmatrix} = \begin{pmatrix} 1 & -\lambda/2 & 0 & \dots & 0 & \lambda/2 \\ \lambda/2 & 1 & -\lambda/2 & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & \ddots & \ddots & -\lambda/2 \\ -\lambda/2 & 0 & \dots & 0 & \lambda/2 & 1 \end{pmatrix} \cdot \begin{pmatrix} U_{-J}^{n} \\ U_{-J+1}^{n} \\ \vdots \\ \vdots \\ \vdots \\ U_J^{n} \end{pmatrix}. \tag{11.51}$$

Mit $\mathbf{u}^{n+1} = \mathbf{L}\mathbf{u}^n + \Delta t\tau^n$ und (11.50) erhalten wir wieder als Fehlergleichung

$$\mathbf{e}^{n+1} = \mathbf{L}\mathbf{e}^n + \Delta t\tau^n, \tag{11.52}$$

$$= (\mathbf{L}^n\tau^0 + \mathbf{L}^{n-1}\tau^1 + \ldots + \mathbf{L}^0\tau^n)\Delta t, \tag{11.53}$$

also

$$\|\mathbf{e}^n\| \leq (\|\mathbf{L}^{n-1}\|\|\tau^0\| + \ldots + \|\mathbf{L}^0\|\|\tau^{n-1}\|)\Delta t. \tag{11.54}$$

Aus der vorausgesetzten Konsistenz schließen wir, daß für jedes $\tau > 0$ ein $\epsilon > 0$ und ein $\eta > 0$ existieren, so daß $\|\tau^n\| < \tau$ für alle $h < \epsilon$ und $\Delta t < \eta h$.

Ferner gibt es aufgrund der Stabilität ein $M > 0$ mit $\|\mathbf{L}^k\| \leq M$ für alle $n \in \mathbf{N}$ und $n\Delta t \leq T$.

Damit aber erhalten wir

$$\|e^n\| \leq n\Delta t\tau M = T\tau M. \tag{11.55}$$

Da nun τ beliebig klein gewählt werden kann, folgt die Konvergenz. $\diamondsuit$

11.4.2 Stabilitätsnachweis mittels Fourier-Methode

Neben der "Matrixmethode" und der hier nicht behandelten "Energiemethode" ist die Fourier- oder v.Neumann-Methode das gebräuchlichste Verfahren, die Stabilität, und also damit die Konvergenz, eines konsistenten Verfahrens nachzuweisen. Wir führen dies am Beispiel des labilen Eulerverfahrens durch.

[2]Eine "wasserdichte" Beweisführung, die im Rahmen der Theorie der Banachräume vonstatten geht, findet man z.B. in Meis, Marcowitz: Numerische Behandlung partieller Differenzialgleichungen. Kap. 4 u. 5.

Zur Vorbereitung findet zunächst die exakte Lösung mittels des Fourieranalyse statt. Man macht den Ansatz

$$u(t,x) = \sum_{k=-J/h}^{J/h} B_k(t) \exp(i\pi kx) \tag{11.56}$$

und setzt gleichmäßige Konvergenz der Summe voraus. Damit haben wir bekanntlich die Möglichkeit, auch beim Grenzübergang $h \to 0, \quad J/h \to \infty$ Summe und Differentialoperator zu vertauschen. Wir erhalten nach dem Einsetzen in die Advektionsgleichung (11.27) für jedes k die gewöhnliche Differentialgleichung:

$$\frac{d}{dt} B_k(t) = -i\pi kc B_k(t). \tag{11.57}$$

Die Lösung $B_k(t) = B_k(0)\exp(-i\pi kct)$ findet man unmittelbar. Somit hat man im allgemeinen Fall als vollständige Lösung

$$u(t,x) = \sum_{k=-\infty}^{\infty} B_k(0) \exp(i\pi k(x-ct)), \tag{11.58}$$

wobei bei den hier vorhandenen reellen Lösungen bekanntlich $B_k = \bar{B}_{-k}$. ist.

Im Falle des Eulerverfahrens setzt man die "Testfunktion"

$$U_j^n = \sum_{k=-J}^{J} A_k(n) \exp(i\pi kx_j) \tag{11.59}$$

in den Algorithmus (11.36) ein und erhält:

$$U_j^{n+1} = \sum_{k=-J}^{J} A_k(n)[1 - i\lambda \sin(k\pi h)] \exp(i\pi kx_j). \tag{11.60}$$

Für die einzelne Wellenzahl k verhalten sich daher die Amplituden der Zeitpunkte n und $n+1$

$$\frac{A_k(n+1)}{A_k(n)} = 1 - i\lambda \sin(\pi kh) =: M_k. \tag{11.61}$$

Dieses Verhältnis nennt man Verstärkungsfaktor M_k, der nach J. v. Neumann nicht schneller als $|M| \leq 1 + \mathcal{O}(\Delta t)$ wachsen darf.

Als diskretes Analogon zu (11.58) erhält man:

$$U_j^n = \sum_{k=-J}^{J} B_k(0) M_k^n \exp(i\pi kx_j). \tag{11.62}$$

Wie findet man nun die Erfahrung des Abschnitts 2 bestätigt, daß das Eulerverfahren labil ist?

Man könnte sich nun auf folgende Schlußweise einlassen:

Wenn die numerische Lösung $U_j^n \approx u(n\Delta t, jh)$ sein soll, so muß auch zu zeigen sein, daß mit (11.58) und (11.62) $M_k^h \approx \exp(-i\pi kct)$ folgt.

Dies ist ja auch der Fall, denn zu einem Zeitpunkt $t = n\Delta t$, also nach n Schritten erhält man

$$\lim_{\substack{\Delta t\to 0\\ h\to 0}} (M_k)^n = \lim_{\substack{\Delta t\to 0\\ h\to 0}} [1 - i\lambda(\pi kh - \frac{1}{3}(\pi kh)^3 + \ldots)]^n \tag{11.63}$$

$$= \lim_{\substack{\Delta t\to 0\\ h\to 0}} [1 - \frac{i\pi kct}{n} + \frac{1}{n}\mathcal{O}(h^2)]^n \tag{11.64}$$

$$= \lim_{h\to 0} \exp(-i\pi kct + \mathcal{O}(h^2)) = \exp(-i\pi kct). \tag{11.65}$$

Die Konvergenz ist unabhängig von λ, und das Verfahren ist anscheinend doch stabil.

Zu dem gleichen Ergebnis gelangt man auch, wenn man unmittelbar die v. Neumann-Bedingung $M_k \leq 1 + O(\Delta t)$ prüft:

$$|M_k|^2 = 1 + \lambda^2 \sin^2(\pi kh) \leq 1 + \lambda^2\pi^2k^2h^2 = 1 + c^2\pi^2k^2(\Delta t)^2 = 1 + \mathcal{O}((\Delta t)^2). \tag{11.66}$$

Also wieder $\lim_{\Delta t\to 0} |M_k|^2 = 1$.

Der Fehler beider Ansätze besteht darin, daß man immer k festgehalten hat, auch bei der Verfeinerung des Gitters. Man muß sich aber einen ständigen Blick auf die jeweils höchsten Wellenzahlen verschaffen, indem man k an J koppelt, z.B. $k = \frac{J}{2} = \frac{1}{2h}$ (was einer Wellenlänge von $4 \cdot h$ entspricht, denn $k = \frac{2Jh}{4h} = \frac{J}{2}$). Daher erhält man

$$|M_{J/2}|^2 = 1 + \lambda^2 \sin^2(\frac{\pi \overbrace{Jh}^{=1}}{2}) = 1 + \lambda^2 > 1, \tag{11.67}$$

und mit $n\Delta t, \lambda$ fest, folgt $\lim_{\Delta t\to 0} |M_{J/2}|^n \to \infty$.

Die Amplitude der Welle der Länge $4h$ wächst also exponential!

So sieht man, daß immer die kleinen Wellen den Ärger bereiten, die auch bei völlig richtigen Anfangswerten zumindest durch Rundungsfehler eingeschleppt werden und im Laufe der Integration zu störenden Größenordnungen wachsen.

11.5 Diskretisierungsverfahren

11.5.1 Lax-Wendroff-Verfahren

Gibt es ein Verfahren, daß einen Mittelweg zwischen dem labilen Eulerverfahren (11.36) und dem diffusiven Verfahren beschreitet?

Einen Ansatz hierzu bietet die Differenzengleichung

$$U_j^{n+1} = U_j^n - \frac{\lambda}{2}(U_{j+1}^n - U_{j-1}^n) + \mu(U_{j-1}^n + U_{j+1}^n - 2U_j^n), \tag{11.68}$$

da man zwischen $\mu = 0 (\cong$ Eulerverfahren) und $\mu = \frac{1}{2} (\cong$ diffusives Verfahren) beliebig variieren kann.

Wie ermittelt man das beste μ ?

Wir erinnern uns, daß das Eulerverfahren in der räumlichen Differentation von quadratischer, in der zeitlichen Differentation dagegen von nur linearer Ordnung ist.

Die Taylorzerlegungen liefern

$$u_t|_j^n = \frac{1}{\Delta t}(u_j^{n+1} - u_j^n) - \frac{\Delta t}{2}u_{tt} + \mathcal{O}((\Delta t)^2) \tag{11.69}$$

$$u_{xx}|_j^n = \frac{2}{h^2}(u_{j\pm1}^n - u_j^n \mp h u_x|_j^n) + \mathcal{O}(h^2) \tag{11.70}$$

oder nach (11.29)

$$u_{xx}|_j^n = \frac{1}{h^2}(u_{j-1}^n + u_{j+1}^n - 2u_j^n) + \mathcal{O}(h^2). \tag{11.71}$$

Für u_{tt} findet man unmittelbar durch das gegebene Problem

$$u_{tt} = -cu_{xt} = c^2 u_{xx}. \tag{11.72}$$

Somit lautet nun das Lax-Wendroff Verfahren als eine um das quadratische Glied erweiterte Taylorapproximation

$$U_j^{n+1} = U_j^n + \Delta t(-c\frac{U_{j+1}^n - U_{j-1}^n}{2h}) + \frac{\Delta t^2}{2}\big(c^2\frac{U_{j-1}^n + U_{j+1}^n - 2U_j^n}{h^2}\big), \tag{11.73}$$

d.h., $\mu = \frac{\lambda^2}{2}$ liefert das gesuchte Verfahren quadratischer Ordnung.

11.5.2 Leapfrog–Verfahren

Führt man bei der Diskretisierung der Zeitableitung auch zentrierte Differenzen ein, so erhält man das Leapfrog-Schema

$$U_j^{n+1} = U_j^{n-1} - \lambda(U_{j+1}^n - U_{j-1}^n). \tag{11.74}$$

Das Verfahren erhält also drei Zeitebenen, d.h., daß das Einsetzen der Testfunktion $U_j^n = A_k(n)\exp(ikx_j)$ zu einer Zwei-Term-Rekursion führt

$$A_k(n+1) = A_k(n-1) - 2iA_k(n)\lambda\sin(kh). \tag{11.75}$$

Setzen wir zunächst voraus, daß der Verstärkungsfaktor M_k über zwei Zeitschritte gleich ist, also $M_k = \frac{A_k(n+1)}{A_k(n)} = \frac{A_k(n)}{A_k(n-1)}$, so erhält man mit $p :=$

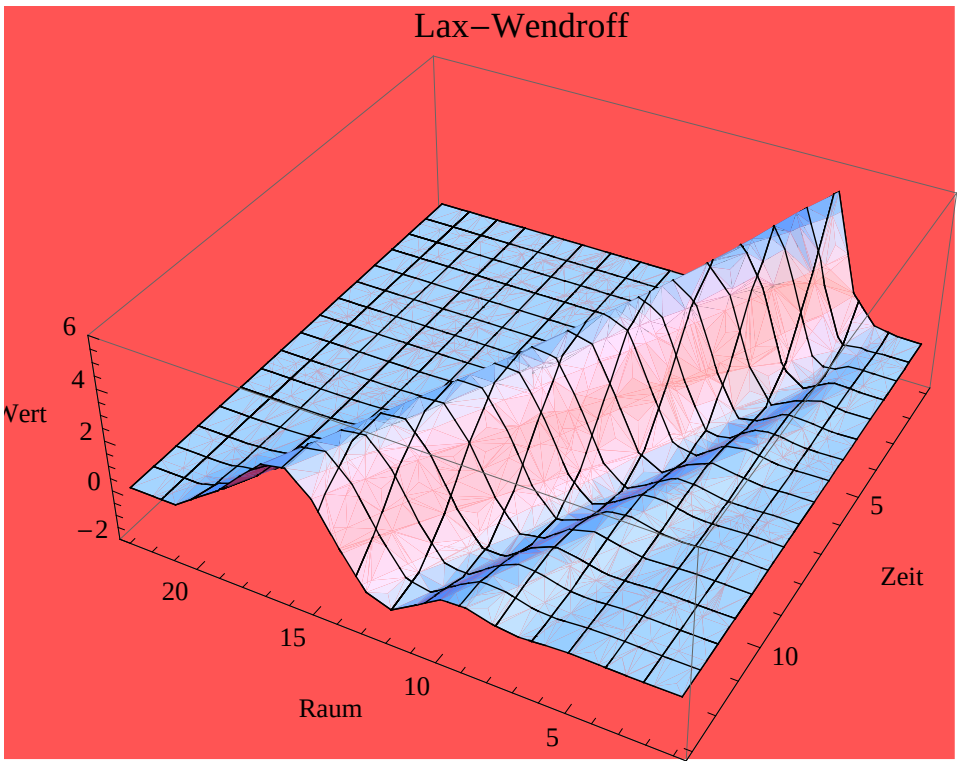


Abbildung 11.5: Wie Abb. 11.1 nur mit dem Lax-Wendroff-Verfahren (11.73).

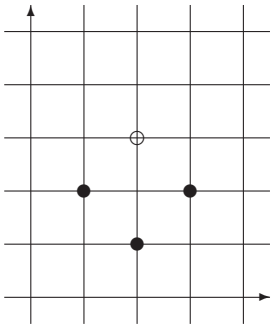


Abbildung 11.6: Schema des Leapfrog-Verfahrens. Symbole wie in Abb. 11.3.2

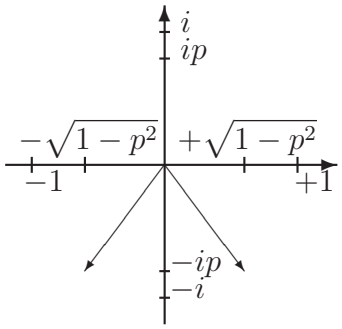


Abbildung 11.7: Verstärkungsfaktoren im leapfrog-Verfahren in der komplexen Zahlenebene; Beispiel mit $p = 4/5$.

$\lambda \sin(kh)$ die Gleichung $M_k^2 + 2ipM_k - 1 = 0$ mit $M_{k1,2} = -ip \pm \sqrt{1-p^2}$ als Lösung.

Die Verstärkungsfaktoren liegen also solange auf dem Einheitskreis, wie $|\lambda| \leq 1$. Diese Bedingung nennt man nach Courant-Friedrichs-Levy (1928) das *CFL-Kriterium*.

11.5.3 Konsequenzen aus der zweideutigen Lösung

Aber welches M_k beschreibt die physikalisch sinnvolle Lösung?

Um diese Frage zu beantworten, wird der Algorithmus um den Preis einer Vektorgleichung formal auf ein Zwei-Ebenen-Verfahren reduziert.

$$\begin{pmatrix} A_k(n+1) \\ A_k(n) \end{pmatrix} = \begin{pmatrix} -2ip & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} A_k(n) \\ A_k(n-1) \end{pmatrix} \tag{11.76}$$

Im Falle der Stabilität muß also für die Matrix $\mathbf{B} := \begin{pmatrix} -2ip & 1 \\ 1 & 0 \end{pmatrix}$ gelten $\|\mathbf{B}\| =: M \leq 1 + O(\Delta t)$.

Die Eigenwerte von B sind $\tilde{\lambda}_{1,2} = -ip \pm \sqrt{1-p^2} = M_{1,2}$.

Da nun $\left|\tilde{\lambda}_{1,2}\right| = 1$, ist die oben gemachte Voraussetzung der Konstanz der Verstärkungsfaktoren M_k also überflüssig.

Die Eigenvektoren lauten

$$\mathbf{x}_{1,2} = \begin{pmatrix} -ip \pm \sqrt{1-p^2} \\ 1 \end{pmatrix} \tag{11.77}$$

und sind also linear unabhängig.

Daher ist die Darstellung

$$\begin{pmatrix} A_k(n+1) \\ A_k(n) \end{pmatrix} = \mathbf{B}^n \begin{pmatrix} A_k(1) \\ A_k(0) \end{pmatrix} = \alpha \mathbf{B}^n \mathbf{x}_1 + \beta \mathbf{B}^n \mathbf{x}_2 \tag{11.78}$$

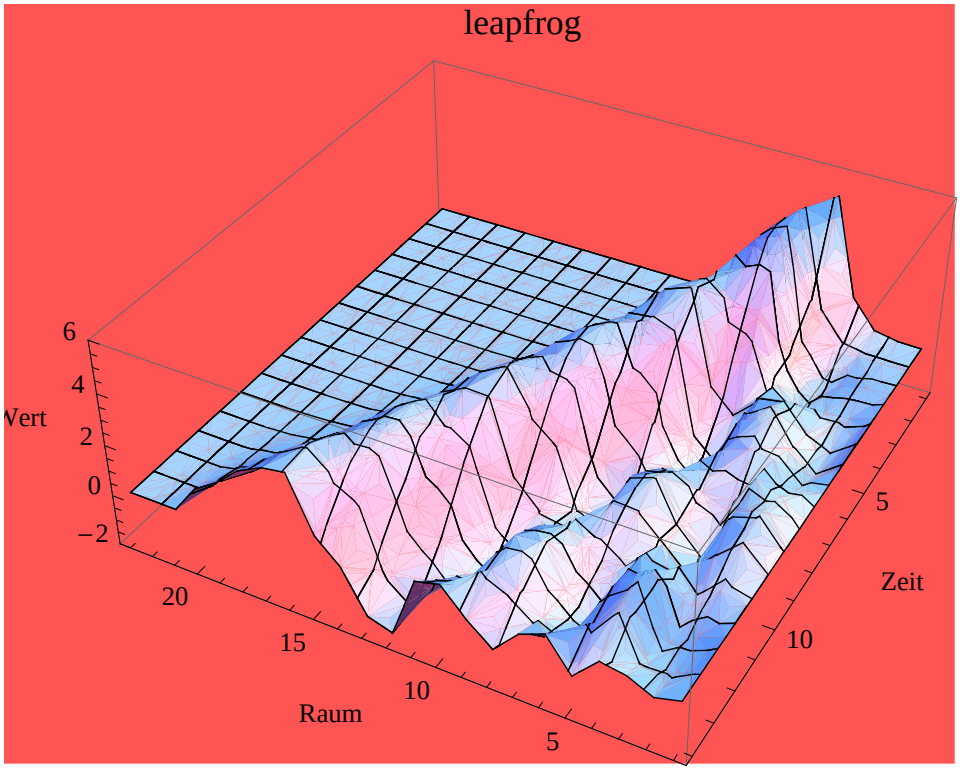


Abbildung 11.8: Wie Abb. 11.1 nur mit dem leapfrog-Verfahren (11.74).

möglich, mit der Konsequenz

$$\left\| \begin{pmatrix} A_k(n+1) \\ A_k(n) \end{pmatrix} \right\| \leq \alpha \|\mathbf{x}_1\| + \beta \|\mathbf{x}_2\| \tag{11.79}$$

weil $\|\mathbf{B}\| = 1$. Die Folgen: $\lim_{\Delta t \to 0} \tilde{\lambda}_{k1,2} = \lim_{\Delta t \to 0} M_{k1,2} = \pm 1$. Also kann nur mit $M_{k,1}$ *die gewünschte physikalische Lösung* verbunden sein, da sie sich *stetig entwickelt*, im Gegensatz zu M_{k2}, wo bei jedem Zeitschritt das *Vorzeichen wechselt*! Diese "Lösung" wird *computational mode* (auch "Geisterlösung") genannt.

Wir haben also *zwei disjunkte Gitterpunktmengen* (Punkte/Karos $\cong$ Indexsumme $j + n$ gerade/ungerade), deren *Einzellösungen sich auseinanderentwickeln können*, und die durch den *computational mode* formal zusammengehalten werden.

Ziel ist es, daß die disjunkten Lösungen zusammenfallen, ohne daß diese Geisterlösung auftritt, d.h. $\beta = 0$.

Entscheidend hierfür ist die Güte der Anfangswerte zu den Zeitpunkten $t = 0$ und $t = 1 \cdot \Delta t$. Durch ihre mangelnde Qualität und durch Rundungsfehler tritt der *computational mode* nach und nach immer wieder auf. Durch regelmäßiges Filtern, d.h. Mitteln der disjunkten Lösungen, kann er wieder gedämpft werden (z.B. etwa alle 100 Zeitschritte, aber diffusive Wirkung!).

Fazit:

Der Vergleich von Leapfrog (11.74) und Lax-Wendroff Verfahren (11.73) zeigt sofort den geringeren Rechenaufwand von ersterem. Beide sind aber Verfahren 2. Ordnung!

Entscheidend für den Vorteil von (11.74) ist es also, gute Anfangswerte zu finden. Nimmt man dagegen die Anfangswerte von $t = 0$ etwa auch zum Zeitpunkt $t = \Delta t$, so hat man zusätzlich die retrograd fortschreitende numerische Lösung. In der Praxis arbeitet man sich anfangs vom Zeitpunkt $t = 0$ durch sehr kleine "Unterzeitschritte" mit Zwei–Level–Verfahren an den Zeitpunkt $t = 1 \cdot \Delta t$ heran, um auch hier gute Anfangswerte zu gewinnen.

11.5.4 Phasenfehler

Bisherige Gütekriterien waren gegeben durch Stabilität und Konvergenz, mit dem Ziel möglichst geringe Diskretisierungsfehler zu erhalten. Ein bedeutendes Teilkriterium hiervon ist die *Phasentreue*. Die mit der Testfunktion ermittelte Lösung ist nicht dispersiv. Allgemein kann man den relativen Phasenfehler η definieren:

$$\eta := -\frac{\arctan(\Im(M_k)/\Re(M_k))}{kc\Delta t} - 1. \tag{11.80}$$

Er kann sowohl durch räumliche als auch zeitliche Diskretisierung hervorgerufen werden.

Phasenfehler durch räumliche Diskretisierung

Die räumliche Diskretisierung von $\frac{\partial U_j}{\Delta t} + \frac{c}{2h}(U_{j+1} - U_{j-1}) = 0$ mit der Testfunktion $U_j(t) = \sum_k B_k(t) \exp(ikx_j)$ liefert

$$\frac{d}{dt} B_k = -ikc \underbrace{\frac{\sin(kh)}{kh}}_{\leq 1} B_k, \tag{11.81}$$

also eine Verlangsamung insbesondere bei kurzen Wellen.

Phasenfehler durch zeitliche Diskretisierung

Es sei nun $p := +kc\Delta t = +\lambda kh$

Leapfrog-Verfahren

Mit $M_{k1} = \lambda_1 = -ip + \sqrt{1-p^2}$ folgt mit (11.80)

$$\eta = \frac{\arctan(\frac{-p}{\sqrt{1-p^2}})}{-p} - 1 = \frac{1}{p} \arctan(p + \frac{p^3}{2} + \ldots) - 1 = \frac{p^2}{6} + \ldots, \tag{11.82}$$

da nach Taylor gilt: $\arctan x = x - \frac{x^3}{2} + \frac{x^5}{5} - \ldots$.

Das Leapfrog-Verfahren wirkt also beschleunigend. Man kann zeigen, daß die verzögernde Wirkung der räumlichen Diskretisierung und die beschleunigende Wirkung der zeitlichen Diskretisierung zu einer stehenden Welle von der Länge $4h$ führen.

Lax-Wendroff-Verfahren

Nach einigen Rechnungen erhält man für das Amplitudenverhältnis:

$$\frac{A_k(n+1)}{A_k(n)} = M_k = 1 - i\lambda \sin(kh) - \lambda^2(1 - \cos kh) \tag{11.83}$$

$$= 1 - ip - \lambda^2(1 - \sqrt{1 - p^2/\lambda^2}) \tag{11.84}$$

wobei wieder $p := \lambda \sin(kh)$.

Also

$$\eta = -\frac{\arctan(-p/1 - \lambda^2(1 - \sqrt{1 - p^2/\lambda^2}))}{p} - 1 \tag{11.85}$$

$$= \frac{1}{p} \arctan[p(1 + \lambda^2(1 - \sqrt{1 - p^2\lambda}) + \ldots)] - 1 = \frac{p^2}{6} + \ldots \tag{11.86}$$

Damit stellt man also ebenfalls beschleunigende Wirkung fest.

Abbildung 11.9: Diskretisierungsschema des impliziten Eulerverfahrens (links) und des Trapezverfahrens (rechts). Symbole wie in Abb. 11.3.2

11.6 Implizite Verfahren

Mit einer formalen Zeitumkehr kann man auch die Ableitung, die zum Eulerverfahren geführt hat, anwenden, um folgende Differenzengleichung zu finden:

$$U_j^{n+1} = U_j^n - \frac{\lambda}{2}(U_{j+1}^{n+1} - U_{j-1}^{n+1}). \tag{11.87}$$

Sie heißt

implizites Euler–Verfahren

. (Euler-backward-scheme)

Bei gegebenen Randwerten $U_{\pm J}^{n+1}$ ist durch (9.1) ein lineares Gleichungssystem mit einer Tridiagonalmatrix gegeben.

Als Verstärkungsfaktor erhält man

$$M_k = \frac{1}{1 + i\lambda \sin(kh)} = \frac{1 - i\lambda \sin(kh)}{1 + \lambda^2 \sin^2(kh)}, \tag{11.88}$$

also $\|M_k\| \leq 1$ unabhängig von λ! Das Verfahren ist damit für jeden Zeitschritt stabil.

Der Phasenfehler lautet

$$\eta = -\frac{\arctan(-p)}{p} - 1 = -\frac{p^2}{3} + \frac{p^4}{5} \ldots . \tag{11.89}$$

Er wirkt also besonders bei kurzen Wellen verlangsamend und damit dispersiv.

Ein weiteres Verfahren ist das

Trapezschema

mit der Diskretisierung

$$U_j^{n+1} = U_j^n - \frac{\lambda}{2}\left(\frac{U_{j+1}^n + U_{j+1}^{n+1}}{2} - \frac{U_{j-1}^n + U_{j-1}^{n+1}}{2}\right). \tag{11.90}$$

Man findet als Verstärkungsfaktor:

$$M_k = \frac{1 - i\frac{\lambda}{2}\sin(kh)}{1 + i\frac{\lambda}{2}\sin(kh)}. \tag{11.91}$$

Dieser ist also dem Betrag nach auch immer < 1 und somit immer stabil. Der Phasenfehler ist

$$\eta = \frac{\arctan(\frac{-p}{1-p^2/4})}{p} = -p^2/12 + \ldots, \tag{11.92}$$

und zeigt damit den gleichen Phasenfehler wie das obere Verfahren, nur viermal so schwach ausgeprägt.

Der entscheidende Vorteil beider Verfahren ist also, daß der Zeitschritt Δt unabhängig von der räumlichen Diskretisierung h gewählt werden kann. Der zu entrichtende Preis ist der Aufwand zur Lösung eines linearen Gleichungssystems.

11.7 Das Courant-Friedrichs-Levy-Kriterium

Entscheidend für die Bewertung eines Verfahrens ist auch das Maß der Freiheit, die Zeitschrittgröße zu wählen. Wie oben gezeigt, befreien die impliziten Verfahren um den Preis der Lösung eines linearen Gleichungssystems von der Beschränkung

$$\lambda = \frac{c\Delta t}{h} \leq 1. \tag{11.93}$$

Diese Bedingung wird das *Courant-Friedrichs-Levy-Kriterium* (1928) genannt.

Wird diese Bedingung gefordert aber verletzt, so ist die Konvergenz nicht mehr gegeben. Dies kann man sich auf einfache Weise graphisch klar machen:

Aus Gründen der Übersicht nehmen wir ein einfaches *forward in time and upstream*-Verfahren, welches die Gleichung (11.27) so diskretisiert:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + c\frac{U_j^n - U_{j-1}^n}{h} = 0. \tag{11.94}$$

Gegeben sei das Gitter der folgenden Skizze mit der Zeitachse in y-Richtung. Die Charakteristik gibt hier den Weg eines Signals vom Punkt $(t = 0, j = 4)$ zum Punkt $(t = 6, j = 12)$ an. Bei der Berechnung dieses letztgenannten Punktes gehen nach dem gerade genannten Schema aber nur die Kreise ein, die das Anfangssignal $t = 0$ gar nicht einschließen. Das Ergebnis kann also nicht richtig sein. Halbiert man dagegen die Zeitschritte und nimmt damit einen Berechnungsmodus über die kleinen schwarzen Punkte an, so wird in diesem Falle das Anfangssignal in die Berechnung eingeschlossen, und das CFL-Kriterium ist erfüllt.