Numerische Mathematik 1
Skript zur Vorlesung

Prof. Dr. Erich Novak

Fassung vom Oktober 2011

Hinweise bitte an erich.novak@uni-jena.de.

Der korrekte Name des Moduls ist jetzt: „Einführung in die Numerische Mathematik und das Wissenschaftliche Rechnen“.

Inhaltsverzeichnis

Kapitel 1

Einführung und Übersicht

1.1 Allgemeine Problemstellung

Definition. *Gegenstand der Numerischen Mathematik ist die Konstruktion und die Analyse von Algorithmen zur Lösung stetiger Probleme der Mathematik.*

Literatur. Es gibt viele brauchbare Bücher zur Numerischen Mathematik. Ich empfehle hier nur zwei Bücher, die mir besonders gut gefallen.

D. Kincaid, W. Cheney: Numerical Analysis. Brooks/Cole. 3. Auflage 2002. (Jetzt bei der American Math. Soc. erhältlich, 89 US Dollars.)

M. Hanke-Bourgeois: Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens. Teubner Verlag, 3. Auflage 2009.

Bemerkung. Die Numerische Mathematik erscheint oft als eine Summe von Rezepten für eine Vielzahl von numerischen Problemen. Wir wollen in einem ersten Kapitel etwas Übersicht gewinnen und zeigen, wie die Probleme der Numerik *allgemein* aussehen.

Wir beginnen mit einer Liste von typischen Aufgaben, wobei wir jeweils ein Beispiel angeben.

Typische Aufgaben der Numerischen Mathematik.

1. Nichtlineare Gleichungen. Beispiel: Bestimme $x \in [0,1]$ mit $x = \cos x$ oder $f(x) = x - \cos x = 0$.

2. Rekonstruktion von Funktionen (Interpolation und Ausgleichsrechnung). Beispiel: Gesucht ist (eine Näherung von) $f \in F$, wenn nur endlich viele Funktionswerte $f(a_i) = b_i$ (für $i = 1, \ldots, n$) bekannt sind.

3. Numerische Integration. Beispiel: Gesucht ist $\int_0^1 e^{-x^2}\, dx$.

4. Lineare Gleichungssysteme. Beispiel: Sei $H = (h_{ij})$ mit $h_{ij} = 1/(i+j-1)$ für $1 \le i, j \le n$ die sogenannte *Hilbertmatrix* der Ordnung n. Gesucht ist die Lösung von $Hx = y$ für $y_i = 1/(n+i)$.

5. Lineare Optimierung: Diese Probleme sind von der folgenden Form. Gesucht ist $x \in \mathbb{R}^n$ mit $Ax \leq b$ (komponentenweise) und $L(x) = \max!$ Hierbei ist $L : \mathbb{R}^n \to \mathbb{R}$ linear und A ist eine $m \times n$ Matrix und $b \in \mathbb{R}^m$.

6. Berechnung von Eigenwerten. Beispiel: Berechne die Eigenwerte der Hilbertmatrix H der Ordnung n.

7. Nichtlineare Gleichungssysteme und nichtlineare Optimierung. Beispiel: Gesucht ist eine Lösung des Systems $x = \sin(x + y)$ und $y = \cos(x - y)$.

8. Lösung von Differential- und Integralgleichungen. Beispiel: Gesucht ist eine Lösung der Anfangswertaufgabe (*Pendelgleichung*) $y'' + \sin y = 0$ mit $y(0) = 0$ und $y'(0) = c > 0$.

Abstrakte Version der Problemstellung. Gesucht ist bei diesen Problemen jeweils eine Zahl, ein Vektor im $\mathbb{R}^k$ oder eine Funktion, allgemein ein Element g eines normierten Raumes G. Dabei hängt g in jeweils spezifischer Weise ab vom Problemelement f, das allgemein auch wieder Element eines normierten Raumes F_1 ist. Diese Abhängigkeit beschreiben wir durch eine Abbildung

$$S : F \to G,$$

die nur auf einer Teilmenge $F \subset F_1$ definiert sein muß.

Bei 1) ist etwa

$$F = \{f \in C[0,1] \mid f(0) < 0,\ f(1) > 0 \text{ und } f \text{ ist strikt monoton}\}$$

und $G = \mathbb{R}$ mit

$$S(f) = f^{-1}(0) = \text{ die Nullstelle von } f.$$

Bei 4) ist entsprechend etwa $F = \{A \in R^{n \cdot n} \mid A \text{ ist regulär}\} \times R^n$ und $G = \mathbb{R}^n$ mit

$$S(A, h) = x \qquad \text{genau dann, wenn} \qquad Ax = h,$$

d.h. $S(A, h) = A^{-1}h$.

Diese beiden Beispiele unterscheiden sich allerdings in einer wichtigen Hinsicht: Beim Gleichungssystem $Ax = h$ sind i.a. sowohl A als auch h bekannt, die Lösung x ist dann durch die Vorgaben (durch die gegebene *Information*, die aus endlich vielen Zahlen besteht) eindeutig bestimmt.

Dagegen kann eine monotone Funktion f i.a. nicht als vollständig bekannt angesehen werden, da die monotonen Funktionen eine unendlichdimensionale Menge bilden. Lediglich einzelne Funktionswerte (von f oder auch von einer Ableitung von f) können in die Rechnung eingehen, nicht f selbst.

Mit so einer *unvollständigen Information* läßt sich die Nullstelle $S(f) = f^{-1}(0)$ von f i.a. nur näherungsweise bestimmen.

Allgemein wird man also noch mit einer *Informationsabbildung* $N : F \to \mathbb{R}^n$ arbeiten müssen. Bekannt ist nicht $f \in F$, sondern lediglich $N(f) \in \mathbb{R}^n$, wobei bei einem

endlichdimensionalen F natürlich N die identische Abbildung sein kann. Man spricht dann von vollständiger Information. Ist F ein Raum von Funktionen, so kann z.B.

$$N(f) = (f(x_1), \dots, f(x_n))$$

mit gewissen *Knoten* $x_1, \dots, x_n$ gelten.

Gesucht ist also eine Näherung $\widetilde{S}$ von $S : F \to G$, wobei nur eine endliche Information N benutzt werden kann. Die Näherung $\widetilde{S}$ muß also von der Form $\widetilde{S} = \varphi \circ N$ mit $N : F \to \mathbb{R}^n$ und $\varphi : \mathbb{R}^n \to G$ sein. Im allgemeinen gilt dann $\widetilde{S} \neq S$ für alle solchen $\widetilde{S}$, man muß also einen Fehler in Kauf nehmen und wird natürlich versuchen, diesen klein zu machen.

Computermodelle. Man interessiert sich für den Aufwand, der nötig ist, um gewisse Probleme der Numerik zu lösen. Man geht dabei allerdings nicht von konkreten Computern aus (dann müßte man ja für jeden Computer eine eigene Theorie machen), sondern von abstrakten Rechenmaschinen. Da es verschiedene solcher „abstrakten Rechner“ gibt, die in der Informatik und Numerik nebeneinander benutzt werden, ist es sinnvoll, mit ein paar Bemerkungen zu diesem Thema zu beginnen.

Bei jedem Typ von Komplexitätstheorie wird festgelegt, welche *Operationen* (mit welchen Objekten) erlaubt sind und was sie (an Rechenzeit) kosten. Die *Kosten*, die für die Lösung eines Problems mindestens nötig sind, bezeichnet man als die *Komplexität* des Problems.

In der Informatik werden meist diskrete Probleme betrachtet. Die grundlegenden Objekte sind Bits, lediglich *Bit-Operationen* sind erlaubt. Die Komplexität eines Problems bestimmt sich aus der Anzahl der Bit-Operationen, die nötig sind, um es zu lösen.

Dagegen sind die grundlegenden Objekte der Numerik reelle Zahlen, Matrizen und Funktionen. Der Aufwand einer Abbildung $\varphi : \mathbb{R}^n \to G$ wird gemessen durch die Anzahl der *arithmetischen Operationen* (d.h. Grundrechenarten mit reellen Zahlen, Vergleiche von reellen Zahlen und evtl. noch andere Operationen wie Wurzelziehen) die benutzt werden, um φ (für irgendein f aus einer vorgegebenen Menge F) auszurechnen.

Bei unendlichdimensionalen Mengen F ist es unrealistisch anzunehmen, daß $f \in F$ selbst in die Rechnung eingehen kann. Wir nehmen an, daß gewisse Informationen, etwa Funktionswerte oder Werte von Ableitungen, abgefragt werden können. Als weitere Operation steht also ein *Orakel* für Funktionswerte (und/oder andere Funktionale) zur Verfügung. Pro abgefragter Wert entstehen zusätzlich die Kosten $c > 0$, wobei meist c wesentlich größer ist als 1.

Bei endlichdimensionalen Problemen nimmt man meist an, daß $f \in F$ bekannt ist, so daß keine zusätzlichen Kosten entstehen.

Zusammenfassend erhält man die drei in der folgenden Übersicht gezeigten Komplexitätstheorien.

	diskret	algebraisch	numerisch
grundlegende Objekte	Bits, rat. Zahlen	reelle Zahlen	Funktionen u.a.
Daten	= Objekte	= Objekte	z.B. Funktionswerte
Information	vollständig	vollständig	unvollständig
Kosten der Information	keine Kosten	keine Kosten	c pro Wert
Problemgröße	Anzahl der Bits	Anzahl der Zahlen	Funktionenraum
Operationen	Manipulationen mit Bits	arithmet. Oper. mit reellen Zahlen	arithmet. Oper. mit reellen Zahlen
Rechenkosten	Anzahl der Bit-Operationen	Zahl der Operationen mit reellen Zahlen	Zahl der Operationen mit reellen Zahlen

Beispiele. Wir geben Beispiele von Ergebnissen in jedem der drei Modelle an.

a) *Diskrete Komplexität*: Edmonds (1967) zeigte, daß der Gauß-Algorithmus lineare Gleichungssysteme (mit rationalen Daten) in polynomial-beschränkter Zeit löst. Es gibt also ein Polynom p mit folgender Eigenschaft. Ist ein lineares Gleichungssystem $Ax = h$ mit rationalen a_{ij} und h_i gegeben und sei l die Anzahl der Bits, die man zur Beschreibung der Daten A und h braucht. Dann ist die Rechenzeit des Gauß-Verfahrens kleiner als $p(l)$, wobei eine Bit-Operation pro Zeiteinheit durchgeführt werden kann.

Khachiyan (1979) zeigte, daß das Problem der linearen Optimierung (für rationale Daten) in polynomial beschränkter Zeit gelöst werden kann.

b) *Algebraische Komplexität*: Trivialerweise löst das Gauß-Verfahren jedes lineare Gleichungssystem mit n Gleichungen und n Unbekannten in einer Zeit, die durch konst$\cdot n^3$ abgeschätzt werden kann. Dagegen ist bis heute nicht bekannt, ob das Problem der linearen Programmierung in polynomial beschränkter Zeit gelöst werden kann – insbesondere sind die Algorithmen von Khachiyan oder von Karmarkar nicht polynomial beschränkt.

c) *Komplexität in der Numerischen Analysis* (auch *information-based complexity* genannt): Sei

$$F = \{f : [0,1]^d \to \mathbb{R} \mid f \in C^r,\ \|D^\alpha f\|_\infty \le 1 \text{ für alle part. Abl. } D^\alpha \text{ der Ordnung } r\}.$$

Das Problem der Berechnung einer Zahl β mit

$$|\beta - \int_{[0,1]^d} f(x)dx| \le \varepsilon$$

hat eine Komplexität der Ordnung $\varepsilon^{-d/r}$, falls nur Funktionsauswertungen als Informationen zulässig sind. Dies wurde von Bakhvalov (1959) bewiesen. Er hat gezeigt, daß $\widetilde{S}$ von der Form

$$\widetilde{S}(f) = \sum_{i=1}^{n} c_i f(x_i)$$

mit $n \asymp \varepsilon^{-d/r}$ gewählt werden kann. Die Kosten einer solchen Abbildung $\widetilde{S}$ betragen offensichtlich $n \cdot c + n + (n-1)$.

In der Numerik sind die grundlegenden Objekte stets reelle Zahlen, Matrizen oder Funktionen. Die diskrete Komplexität wird daher im Folgenden nicht mehr behandelt.

Die oben genannten Beispiele (lineare Gleichungssysteme und lineare Optimierung) zeigen allerdings, daß man auch stetige Probleme innerhalb der diskreten Theorie behandeln kann, sofern man sich auf rationale Daten (allgemein: Daten aus einer abzählbaren Menge) beschränkt.

Definition: Kosten, Komplexität, ε-Komplexität. Sei $\widetilde{S}$ eine Abbildung $\widetilde{S} : F \to G$, die zur Approximation von $S : F \to G$ benutzt wird. Wir nehmen an, daß $\widetilde{S}$ durch ein Rechenverfahren implementiert werden kann; es werden n Funktionale (z.B. Funktionswerte) von $f \in F$ benutzt. Dann ist $\widetilde{S} : F \to G$ von der Form $\widetilde{S} = \varphi \circ N$ mit $N : F \to \mathbb{R}^n$ und $\varphi : \mathbb{R}^n \to G$.

Dann sei $\mathrm{cost}(N, f) = c \cdot n$, wobei $c \geq 0$. Der Fall $c = 0$ entspricht dem Modell der algebraischen Komplexität. Natürlich sind i.a. nur gewisse Abbildungen $N : F \to \mathbb{R}^n$ als Informationsabbildungen zugelassen.

Weiter sei $\mathrm{cost}(\varphi, f)$ gleich der Anzahl der arithmetischen Operationen, die man bei der Berechnung von $\varphi(N(f))$ benutzt. Dann sind

$$\mathrm{cost}(\widetilde{S}, f) := \mathrm{cost}(N, f) + \mathrm{cost}(\varphi, f)$$

die (Gesamt-) Kosten zur Berechnung von $\widetilde{S}(f)$ und

$$\mathrm{cost}(\widetilde{S}) := \sup_{f \in F} \mathrm{cost}(\widetilde{S}, f)$$

sind die maximalen Kosten von $\widetilde{S}$ (auf der Klasse F). Sei nun $S : F \to G$ irgendeine Abbildung. Dann sei

$$\mathrm{comp}(S) := \inf_{\widetilde{S}=S} \mathrm{cost}(\widetilde{S})$$

die Komplexität von S. Dabei erstreckt sich das Infimum über alle $\widetilde{S} = \varphi \circ N$ mit einer erlaubten Informationsabbildung N und einer arithmetisch möglichen Abbildung φ, so daß $\widetilde{S} = S$.

Für den Fall einer endlichdimensionalen Grundmenge F haben viele wichtige Abbildungen S eine endliche Komplexität, d.h. S läßt sich in der Form $S = \widetilde{S} = \varphi \circ N$ schreiben, wobei N eine erlaubte Informationsabbildung ist und φ sich durch erlaubte arithmetische Operationen ausdrücken läßt.

Im unendlichdimensionalen Fall ist dagegen meist $\mathrm{comp}(S) = \infty$, man muß sich daher mit einer Näherung von S zufrieden geben. Dazu muß zunächst der Fehler

$$\Delta(\widetilde{S}(f), f) \geq 0$$

von $\widetilde{S}$ an der Stelle f geeignet definiert sein, z.B. durch

$$\Delta(\widetilde{S}(f), f) = \|S(f) - \widetilde{S}(f)\|$$

mit einer auf G definierten Norm. Dann ist der maximale Fehler von $\widetilde{S}$ gegeben durch

$$\Delta_{\max}(\widetilde{S}) = \sup_{f \in F} \Delta(\widetilde{S}(f), f)$$

und man definiert die ε-Komplexität des Problems durch

$$\mathrm{comp}(S, \varepsilon) = \inf\{\mathrm{cost}(\widetilde{S}) \mid \Delta_{\max}(\widetilde{S}) \leq \varepsilon\}.$$

1.2 Kondition und Stabilität

Kondition: Empfindlichkeit des Problems gegenüber Eingangsfehlern. Wir sind bisher von der Annahme ausgegangen, daß sich die Information $N(f) \in \mathbb{R}^n$ exakt ermitteln läßt und daß auch die arithmetischen Operationen bei φ exakt durchgeführt werden. Der dann enstehende Fehler des idealen Verfahrens $\widetilde{S}$ heißt auch Verfahrensfehler von $\widetilde{S} = \varphi \circ N$.

Wir haben also davon abgesehen, daß bei der Ermittlung von $N(f)$ üblicherweise mit *Datenfehlern* zu rechnen ist und daß bei den arithmetischen Operationen meist *Rundungsfehler* entstehen. Datenfehler enstehen z.B. durch ungenaue Messungen oder durch Runden der im Prinzip exakt bekannten Information $N(f)$. Datenfehler entsprechen einer Änderung von $f \in F$. Wie sehr sich dadurch die exakte Lösung $S(f)$ ändert, hängt ab von der Kondition der Abbildung $S : F \to G$. Die Abbildung $S : F \to G$ heißt gutkonditioniert, sofern kleine (absolute oder relative) Änderungen bei $f \in F$ kleine Änderungen von $S(f)$ ergeben. Ansonsten heißt S schlechtkonditioniert oder schlechtgestellt. Diese Begriffe kann man auf folgende Weise präzisieren und quantifizieren.

Die *absolute Konditionszahl* K_{abs} von S sei die kleinste Zahl (Lipschitzkonstante) mit

$$\|S(f_1) - S(f_2)\| \leq K_{\text{abs}} \cdot \|f_1 - f_2\|$$

für alle $f_1, f_2 \in F$. Es gilt also

$$K_{\text{abs}} := \sup_{f_1 \neq f_2} \frac{\|S(f_1) - S(f_2)\|}{\|f_1 - f_2\|}.$$

Wir schreiben $K_{\text{abs}} = \infty$, falls das Supremum nicht existiert.

Oft ist es sinnvoll, die Kondition lokal (d.h. in einem Punkt $f_1 \in F$) zu definieren. Die Zahl $K_{\text{abs}}(f_1)$ wird definiert durch

$$K_{\text{abs}}(f_1) := \lim_{\varepsilon \to 0} \sup_{\|f_1 - f_2\| < \varepsilon,\ f_1 \neq f_2} \frac{\|S(f_1) - S(f_2)\|}{\|f_1 - f_2\|}.$$

Neben diesen absoluten Konditionszahlen definiert man noch *relative Konditionszahlen*. Diese geben die Empfindlichkeit von S bezüglich Änderungen bei f an, die klein sind im Verhältnis von f. Das Gegenstück zu $K_{\text{abs}}(f_1)$ ist

$$K_{\text{rel}}(f_1) := \lim_{\varepsilon \to 0} \sup_{\|f_1 - f_2\| < \varepsilon,\ f_1 \neq f_2} \frac{\|S(f_1) - S(f_2)\|}{\|S(f_1)\|} : \frac{\|f_1 - f_2\|}{\|f_1\|}.$$

Diese Zahl ist nur definiert, falls $S(f_1) \neq 0$.

Denkt man bei f an einen Vektor in $\mathbb{R}^n$, wir schreiben dann x statt f, so ist der Fehler bei den Koordinaten x_i häufig (etwa beim Rechnen mit einer festen Stellenzahl) durch $\delta \cdot |x_i|$ beschränkt, d.h. durch eine Zahl, die relativ zu $|x_i|$ (und nicht relativ zu $\|x\|$) klein ist.

In so einer Situation ist es sinnvoll, die *relativen Konditionszahlen komponentenweise* zu definieren, wobei man Abbildungen $S : \mathbb{R}^n \to \mathbb{R}$ betrachtet. Die Zahl $K^i_{\text{rel}}(x)$ ist

(falls $S(x) \neq 0$) definiert durch

$$K^i_{\mathrm{rel}}(x) := \limsup_{\varepsilon \to 0} \frac{|S(x) - S((x_1, \ldots, x_{i-1}, x_i + \varepsilon, x_{i+1}, \ldots, x_n))| \cdot |x_i|}{|S(x)| \cdot \varepsilon}.$$

Stabilität: Empfindlichkeit des Verfahrens gegenüber Rundungsfehlern. Rundungsfehler entstehen während der (näherungsweisen) Berechnung von $\widetilde{S}$ dadurch, daß die arithmetischen Operationen nicht exakt ausgeführt werden. Diese zusätzlichen Fehler von realen Verfahren (im Gegensatz zu den bisher betrachteten idealen Verfahren) hängen von der benutzten Software ab. Daher ist es sinnvoll, die Untersuchung von Rundungsfehlern getrennt von der Untersuchung von Verfahrensfehlern durchzuführen. In den Anwendungen wird meist mit einer sogenannten *Gleitkommaarithmetik*, d.h. mit fester Stellenzahl, gerechnet. Es gibt aber auch Formelmanipulationssysteme und Programmiersprachen (zum Beispiel Maple oder Mathematica) die ein exaktes Rechnen mit rationalen und anderen Zahlen ermöglichen.

Sei $\widetilde{S}$ ein ideales Verfahren (zur Approximation von S) und $\widetilde{S}^*$ eine Realisierung von $\widetilde{S}$, die gewisse Datenfehler und Rundungsfehler einschließt.

Wirken sich Fehler, die erst im Verlauf der Berechnung von $\widetilde{S}^*$ (durch Rundung) entstehen, sehr stark auf das Endergebnis aus, so sagt man, das Verfahren $\widetilde{S}$ ist unstabil.

Man kann dies auf folgende Weise präzisieren. Das Verfahren $\widetilde{S}$ lasse sich in der Form

$$\widetilde{S} = T_m \circ T_{m-1} \circ \cdots \circ T_1$$

schreiben, wobei die T_i direkt ausführbare Operationen seien. Ein Fehler, der bei der Berechnung von T_i auftritt, geht in die Restabbildung

$$\widetilde{S}^{(i)} := T_m \circ \cdots \circ T_{i+1}$$

als Eingangsfehler ein und wird daher von $\widetilde{S}^{(i)}$ entsprechend seiner Kondition an das Endergebnis weitergegeben. Man wird daher einen Algorithmus $\widetilde{S}$ stabil nennen, wenn die (absoluten oder relativen) Konditionszahlen aller $\widetilde{S}^{(i)}$ nicht wesentlich größer sind als die von $\widetilde{S}$. Genauer definiert man den *Stabilitätsindex* ϱ von $\widetilde{S}$ durch

$$\varrho = \frac{\max_i K(\widetilde{S}^i)}{K(\widetilde{S})}.$$

Hierbei ist K eine der (oben definierten) Konditionszahlen, wodurch sich verschiedene Präzisierungen ergeben, etwa

$$\varrho_{\mathrm{rel}}(f_1) = \frac{\max_i K_{\mathrm{rel}}(\widetilde{S}^i, f_1)}{K_{\mathrm{rel}}(\widetilde{S}, f_1)}.$$

Ist ϱ wesentlich größer als 1, so heißt $\widetilde{S}$ unstabil. Stabilität ist also eine Eigenschaft von (idealen) Verfahren.

Beispiele. Wir diskutieren kurz die numerische Berechnung (d.h. die Kondition der Abbildungen und den Entwurf von stabilen Algorithmen) von

a) $S(x) = \sqrt{x^2+1} - 1$, $x \in \mathbb{R}$;
b) $S(x) = x - \sin(x)$, $x \in \mathbb{R}$;
c) $S(x) = \arccos(x)$, $-1 \le x \le 1$.

a) Für kleine $|x|$ tritt *Stellenauslöschung* bei der Subtraktion ein, wenn man die Formel direkt auswertet: Rechnen mit 5 gültigen Stellen ergibt zum Beispiel

$$S(0,005) = \sqrt{1,000025} - 1 \approx \sqrt{1,0000} - 1 = 0.$$

Dieses Verfahren ist also unstabil (bezüglich relativer Fehler), im Beispiel erhält man keine einzige gültige Stelle im Ergebnis, obwohl die Abbildung gut konditioniert ist (es gilt $K_{\text{abs}}(x) \le 1$ und $K_{\text{rel}}(x) \le 2$ für jedes x). Der äquivalente Ausdruck

$$S(x) = \frac{x^2}{\sqrt{x^2+1}+1}$$

ergibt ein stabiles Verfahren. Das gleiche Beispiel liefert

$$S(0,005) = \frac{0,000025}{\sqrt{1,000025}+1} \approx \frac{2,5 \cdot 10^{-5}}{2,0000} = 0,000012500$$

und damit 5 gültige Stellen ($S(0,005) = 0,00001249992\ldots$).

b) Hier ergeben sich ähnliche Schwierigkeiten, falls der Betrag von x klein ist. Lösung: Benütze für kleine $|x|$ eine Reihenentwicklung von S und werte die Formel für große $|x|$ direkt aus.

c) Die Abbildung ist für x nahe bei -1 oder 1 schlecht konditioniert. Man hat also (unabhängig von der Art der Berechnung) mit großen Fehlern zu rechnen.

Siehe auch Aufgabe 1.1, wo allgemeine Formeln für K_{abs} und K_{rel} herzuleiten sind.

Lineare Gleichungssysteme werden später genauer untersucht. Hier nur ein kleines Beispiel dafür, daß sich Fehler in der Eingabe verstärken können oder auch abschwächen können: Betrachte

$$\begin{pmatrix} 10 & 11 \\ 11 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1+\varepsilon \end{pmatrix}.$$

Die eindeutig bestimmte Lösung ist $x^t = (-1 + 11\varepsilon, 1 - 10\varepsilon)$. Ist ε ein Eingabefehler (Rundungsfehler oder Meßfehler), so verstärkt sich dieser im Ergebnis etwa um den Faktor 10.

Betrachte im Vergleich dazu das System

$$\begin{pmatrix} 10 & 11 \\ 11 & -12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1+\varepsilon \end{pmatrix}.$$

Die eindeutig bestimmte Lösung ist

$$x^t = (\frac{23 + 11\varepsilon}{241}, \frac{1 - 10\varepsilon}{241}).$$

Ist ε ein Eingabefehler, so wird dieser im Ergebnis gedämpft.

Wir werden später sehen, daß (bei symmetrischen Matrizen) der Größenunterschied der Eigenwerte eine wichtige Rolle spielt: Im ersten Beispiel sind die Beträge der Eigenwerte ($\lambda_{1,2} = 11 \pm \sqrt{122}$) sehr verschieden, im zweiten Beispiel sind sie fast gleich groß ($\lambda_{1,2} = -1 \pm \sqrt{242}$).

Zusätze und Bemerkungen.

- Zum Verständnis des Begriffs der Stabilität ist es wichtig zu wissen, wie der Computer üblicherweise rechnet (*Gleitkommaarithmetik*). Dazu findet man in vielen Büchern Hinweise, besonders anschaulich im Buch von Kincaid und Cheney (Chapter 2). Siehe auch die Hinweise bei den obigen Beispielen.
- *Grundlagenfragen* der Numerik, wie sie in Abschnitt 1.1 angedeutet wurden, werden üblicherweise wenig behandelt. Zum Weiterlesen eignen sich [28, 30].
- Das klassische Modell der *Berechenbarkeit* („bit number model“) wird in [6] sehr gut entwickelt. Für die Berechenbarkeit über der Menge der reellen Zahlen (wegen der wichtigen Arbeit [2] spricht man auch vom „BSS-Modell“) verweise ich auf die Einführung [3]. Das Berechenbarkeitsmodell der Numeriker („real number model with an oracle for function values“) wird oft nur informell beschrieben. Für eine formalere Darstellung siehe [17].

Aufgaben

1.1. Sind die folgenden Abbildungen gut konditioniert, wenn man relative bzw. absolute Fehler betrachtet?

a) $S_a(x) = \arccos(x)$, $x \in [-1, 1]$;
b) $S_b(x) = \sqrt{x}$, $x \geq 0$;
c) $S_c(x) = 1 - x$, $x \in \mathbb{R}$.

Berechnen Sie dazu zunächst die Konditionszahlen $K_{\text{abs}}(x)$ und $K_{\text{rel}}(x)$ einer beliebigen stetig differenzierbaren Abbildung $S : \mathbb{R} \to \mathbb{R}$.
Hinweis: Es ergibt sich

$$K_{\text{abs}}(x) = |S'(x)| \qquad \text{und} \qquad K_{\text{rel}}(x) = \left| \frac{x \cdot S'(x)}{S(x)} \right| ;$$

die zweite Formel gilt, falls $S(x) \neq 0$.

1.2. Diskutieren Sie die numerische Berechnung (d.h. die Kondition der Abbildung und den Entwurf eines stabilen Algorithmus) von $S(x) = \log(x - \sqrt{x^2 - 1})$, wobei $x > 1$.

1.3. Sei $\mathbb{R}^2$ mit der euklidischen Norm versehen und sei $S : \mathbb{R}^2 \to \mathbb{R}$ stetig differenzierbar.

a) Berechnen Sie die absolute Konditionszahl $K_{\text{abs}}(x)$ und (für $S(x) \neq 0$) die relativen Konditionszahlen $K^i_{\text{rel}}(x)$ von S.

b) Wenden Sie die Ergebnisse auf die Abbildungen S_a und S_b an, die gegeben sind durch $S_a(x) = x_1 + x_2$ bzw. $S_b(x) = x_1 x_2$ an.

1.4. Gesucht sei die kleinere Nullstelle von

$$x^2 - ax + 1 = 0 \tag{1.1}$$

für ein $a \geq 3$. Es sei also $F = \{a \in \mathbb{R} \mid a \geq 3\}$ und $S(a)$ sei die gesuchte Lösung von (1.1).

a) Berechnen Sie für $a = 240$ näherungsweise $S(a)$, indem Sie nach der bekannten Formel

$$S(a) = \frac{a}{2} - \sqrt{\frac{a^2}{4} - 1}$$

mit fünf gültigen Stellen rechnen.

b) Formen Sie diese Formel um, damit $S(a)$ (wieder bei Rechnung mit $a = 240$ und fünf gültigen Stellen) viel genauer herauskommt.

c) Ist dieses Problem gutkonditioniert? Bestimmen Sie dazu die Konditionszahlen $K_{\text{abs}}(a)$ und $K_{\text{rel}}(a)$.

d) Zeigen Sie, daß die Berechnungsvorschrift

$$S(a) = \frac{a}{2} - \sqrt{\frac{a^2}{4} - 1}$$

nicht stabil ist bezüglich relativer Fehler: Mit welchem relativen Fehler δ für $S(a)$ muß man rechnen, wenn bei der Berechnung von $\frac{a}{2}$ und $\sqrt{a^2/4 - 1}$ je ein relativer Fehler gemacht wird, der (betragsmäßig) gleich δ_0 ist?

1.5. Es sei

$$f(x) = \prod_{k=1}^{20} (x - k).$$

Ist das Problem, die Nullstellen von f zu finden, gut konditioniert? Betrachten Sie dazu für (betragsmäßig) kleine ε das Polynom

$$f(x) + \varepsilon \cdot g(x),$$

wobei $g(x) = x^{20}$. Es sei $S(\varepsilon)$ die Nullstelle von $f + \varepsilon \cdot g$ in der Nähe von 20. Wie groß ist $S'(0)$? Kommentieren Sie das Ergebnis.

Kapitel 2

Nichtlineare Gleichungen

In diesem Kapitel behandeln wir Verfahren zur näherungsweisen Lösung von Gleichungen mit einer Unbekannten. Gegeben sei also $f : [a, b] \to \mathbb{R}$. Gesucht sei ein Näherungswert für die (bzw. für eine) Nullstelle x^* von f. Natürlich ist eine Gleichung der Form $f(x) = g(x)$ äquivalent zu $(f - g)(x) = 0$, daher genügt die obige Formulierung.

Beispiel: Bestimme $x \in [0, 1]$ mit $x = \cos(x)$ oder $f(x) = x - \cos(x) = 0$.

2.1 Das Bisektionsverfahren

Definition: Das Bisektionsverfahren. Ist $f : [a, b] \to \mathbb{R}$ stetig mit $f(a) < 0 < f(b)$, so muß f in $[a, b]$ eine Nullstelle besitzen (Zwischenwertsatz). Wir berechnen zunächst $x_1 = \frac{a+b}{2}$ und prüfen ob $f(x_1) > 0$. Wenn ja, dann verwenden wir $[a, x_1]$ als neues Näherungsintervall, wenn nein, dann muß eine Nullstelle in $[x_1, b]$ liegen. Das neue Intervall nennen wir $[a_1, b_1]$. Wiederholung des Verfahrens mit dem neuen Intervall liefert eine Intervallschachtelung, die eine Nullstelle bestimmt.

Als Schätzwert $S_n(f)$ für eine Nullstelle x^* liefert das Verfahren nach dem n-ten Schritt, d.h. nach der Berechnung von $f(x_1), \ldots, f(x_n)$, den Mittelpunkt $S_n(f) = \frac{a_n+b_n}{2}$ des zuletzt erhaltenen Intervalls $[a_n, b_n]$.

Fehlerkriterien. Hat f nur eine Nullstelle x^*, so wird man den Fehler durch

$$\Delta(S_n(f), f) = |S_n(f) - x^*|$$

definieren. Allgemein heißt der durch

$$\Delta(S_n(f), f) = \inf\{|x - S_n(f)| \mid f(x) = 0\}$$

definierte Fehler der Fehler nach dem *Wurzelkriterium*. Im Folgenden wird stets dieser Fehlerbegriff zugrundegelegt.

Das *Restkriterium* definiert den Fehler durch $\Delta(S_n(f), f) = |f(S_n(f))|$. Auch dieser Fehlerbegriff ist zuweilen sinnvoll.

Satz 1 (Fehler des Bisektionsverfahrens)**.** *Sei $f : [a, b] \to \mathbb{R}$ stetig mit $f(a) < 0 < f(b)$. Dann gilt für das Bisektionsverfahren*

$$\Delta(S_n(f), f) \leq \left(\frac{1}{2}\right)^{n+1} \cdot (b - a).$$

Beweis: Einfacher Beweis durch Induktion.

2.2 Das Newton-Verfahren

Wir suchen Verfahren, die „besser“ sind als das Bisektionsverfahren. Will man zwei Verfahren bezüglich ihrer Güte vergleichen, so muß dazu das Fehlerkriterium und die betrachtete Funktionenklasse F festgelegt sein. Dann kann man den maximalen Fehler

$$\Delta_{\max}(\widetilde{S}) = \sup_{f \in F} \Delta(\widetilde{S}(f), f)$$

verschiedener Verfahren vergleichen. Diesen Fehler kann man dann ins Verhältnis setzen zum Aufwand der Verfahren, der bei vielen „einfachen“ Verfahren proportional zur *Knotenzahl* n ist. Für das Bisektionsverfahren gilt

$$\Delta_{\max}(S_n) = \left(\frac{1}{2}\right)^{n+1} \cdot (b - a),$$

falls

$$F^* = \{f : [a, b] \to \mathbb{R} \mid f(a) < 0 < f(b) \text{ und } f \text{ stetig}\}$$

oder falls

$$F^0 = \{f \in C^\infty([a, b]) \cap F^* \mid \text{ und } f \text{ strikt monoton}\}.$$

Damit gilt diese Fehleraussage auch für jedes F mit $F^0 \subset F \subset F^*$.

Definition: Newton-Verfahren. Nähert man die Funktion $f : [a, b] \to \mathbb{R}$ durch ihre Tangente im Punkt $x_0 \in [a, b]$ an, so erhält man durch die Nullstelle $S_1(f) = x_1$ der Tangente einen neuen Näherungswert für eine Nullstelle von f. Durch Wiederholung des Verfahrens hofft man, der Nullstelle beliebig nahe zu kommen. Man muß zunächst voraussetzen, daß f mindestens einmal stetig differenzierbar ist. Dann erhält man die Rekursionsformel

$$S_{n+1}(f) = x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

für die natürlich noch ein Startwert x_0 vorgegeben werden muß.

Fehlerbetrachtung zum Newton-Verfahren. Das Newton-Verfahren ist nur für gewisse Funktionen f überhaupt definiert. Für den Nachweis der Konvergenz muß man die Menge der betrachteten Funktionen weiter einschränken. Der Startpunkt x_0 muß festgelegt werden. Ist über die Funktion f auf $[a, b]$ nur $f(a) \cdot f(b) < 0$ bekannt, so wird man $x_0 = \frac{a+b}{2}$ wählen. Man muß beim Bewerten des Fehlers zudem berücksichtigen,

daß insgesamt $2n$ Funktionswerte von f oder f' nötig sind, um $S_n(f) = x_n$ berechnen zu können.

Für 2-mal stetig differenzierbare Funktionen f mit $f(x^*) = 0 \neq f'(x^*)$ gilt für ein ξ_n zwischen x_n und x^* (Satz von Taylor):

$$0 = f(x^*) = f(x_n) + f'(x_n) \cdot (x^* - x_n) + \frac{1}{2} f''(\xi_n) \cdot (x^* - x_n)^2.$$

Für den Fehler $e_n := x_n - x^*$ gilt also

$$e_n f'(x_n) - f(x_n) = \frac{1}{2} f''(\xi_n) \cdot e_n^2.$$

Benützt man noch die Definition von x_{n+1}, so erhält man daraus

$$e_{n+1} = x_{n+1} - x^* = x_n - \frac{f(x_n)}{f'(x_n)} + e_n - x_n = e_n - \frac{f(x_n)}{f'(x_n)} = \frac{1}{2} \frac{f''(\xi_n)}{f'(x_n)} \cdot e_n^2.$$

Beim letzten Gleichheitszeichen haben wir obige Formel durch $f'(x_n)$ dividiert, wir müssen dazu $f'(x_n) \neq 0$ voraussetzen. Dies ist (wegen der Stetigkeit von f') gewährleistet, falls e_n (betragsmäßig) klein genug ist. Wieder wegen der Stetigkeit (von f' und f'') gilt das folgende Lemma.

Lemma. *Sei $f \in C^2(\mathbb{R})$ mit $f(x^*) = 0$ und $f'(x^*) \neq 0$. Sei*

$$c > \frac{1}{2} \left| \frac{f''(x^*)}{f'(x^*)} \right|.$$

Dann gibt es eine Umgebung U von x^* mit folgender Eigenschaft: Ist $x_n \in U$ so folgt $|e_{n+1}| \leq c \cdot e_n^2$.

Satz 2 (Lokaler Fehler des Newton-Verfahrens)**.** *Sei $f \in C^2(\mathbb{R})$ und x^* eine einfache Nullstelle, d.h. $f(x^*) = 0 \neq f'(x^*)$. Dann gibt es für jedes*

$$c > \frac{1}{2} \left| \frac{f''(x^*)}{f'(x^*)} \right|$$

eine Umgebung N von x^* mit folgender Eigenschaft. Für jeden Startwert $x_0 \in N$ für das Newton-Verfahren gilt: Für jedes $n \in \mathbb{N}$ ist $x_n \in N$ und

$$\lim_{n \to \infty} x_n = x^*.$$

Für den Fehler gilt

$$|x_{n+1} - x^*| \leq c \cdot |x_n - x^*|^2.$$

Dafür sagt man auch kurz: Das Newtonverfahren konvergiert im Fall einer einfachen Nullstelle lokal quadratisch.

Zum Beweis von Satz 2 benutzt man natürlich das vorhergehende Lemma. Dieses garantiert bei $x_n \in U$ allerdings nicht, daß auch $x_{n+1} \in U$. Wir wählen daher $N =$

$]x^* - \delta, x^* + \delta[$ mit $\delta > 0$ so, daß $N \subset U$ und $\delta \leq \frac{1}{2c}$. Für $x_n \in N$ gilt dann $|e_n| \leq \frac{1}{2c}$ und

$$\frac{|e_{n+1}|}{|e_n|} \leq c \cdot |e_n| \leq \frac{1}{2}.$$

Daraus folgt $x_{n+1} \in N$ und die Konvergenz der Folge $(x_n)_n$ gegen x^*.

Nur unter starken zusätzlichen Voraussetzungen kann man zeigen, daß das Newton-Verfahren global konvergiert, d.h. für jeden Startwert. Beispielsweise gilt

Satz 3 (Globale Konvergenz des Newton-Verfahrens)**.** *Sei $f \in C^2(\mathbb{R})$ mit $f' > 0$ und $f'' > 0$, d.h. f ist strikt monoton und konvex. Weiter sei $f(\xi) < 0$ für ein $\xi \in \mathbb{R}$. Dann hat f genau eine Nullstelle x^* und das Newton-Verfahren konvergiert für jeden Startwert $x_0 \in \mathbb{R}$ gegen x^*.*

2.3 Vergleich von Bisektions- und Newtonverfahren

Optimalität des Bisektionsverfahrens. Für das Bisektionsverfahren gilt

$$\Delta_{\max}(S_n) = \left(\frac{1}{2}\right)^{n+1} \cdot (b - a),$$

für jede Funktionenklasse F mit

$$F^0 \subset F \subset F^*.$$

Man kann zeigen, daß das Bisektionsverfahren für alle diese Funktionenklassen optimal ist. Das heißt, es gibt kein Verfahren mit einer besseren Fehlerabschätzung bei gleichem Aufwand, gemessen an der Zahl der Funktionsauswertungen. Dies gilt auch, wenn man (etwa bei $F = F^0$) zusätzlich die Ableitungen an den Knoten x_i mitverwendet. Der folgende Satz zeigt, daß für gewisse andere Klassen F das Newton-Verfahren eine viel bessere Fehlerabschätzung als das Bisektionsverfahren zuläßt.

Satz 4 (Globaler Fehler des Newton-Verfahrens)**.** *Es sei*

$$F = \{f \in C^2([a,b]) \mid \left|\frac{f''(y_1)}{f'(y_2)}\right| \leq 2c \text{ für } y_i \in [a,b] \text{ und } x^* \in [a + \frac{l}{4}, b - \frac{l}{4}]\},$$

wobei $l := b - a \leq \frac{1}{c}$. Hierbei sei x^* die eindeutig bestimmte Nullstelle von f. Dann gilt für alle $f \in F$, daß das Newton-Verfahren mit Startwert $x_0 = \frac{a+b}{2}$ konvergiert. Weiter gilt für $S_n(f) = x_n$ die Fehlerabschätzung

$$\Delta_{\max}(S_n) \leq l \cdot 2^{-2^n}.$$

Zum Beweis: Die Aussage $e_0 \leq \frac{l}{2}$ ist trivial. Der Induktionsschritt folgt aus der schon bewiesenen Abschätzung

$$e_{n+1} = \frac{f''(\xi_n)}{2f'(x_n)} \cdot e_n^2.$$

Insbesondere folgt $|e_n| \leq \frac{l}{4}$ für alle $n \in \mathbb{N}$ und wegen der Voraussetzung $x^* \in [a + \frac{l}{4}, b - \frac{l}{4}]$ ist garantiert, daß die x_n in $[a, b]$ liegen.

Optimalität. Das Newton-Verfahren konvergiert also für gewisse Funktionen und hinreichend kleine Intervalle sehr schnell. Man kann sogar zeigen, daß für Mengen F wie im Satz 4 das Newton-Verfahren in gewisser Weise optimal ist, sofern man Informationen N_n von der Art

$$N_n(f) = (f(x_1), f'(x_1), \ldots, f(x_n), f'(x_n))$$

zuläßt.

Da man beim Newton-Verfahren auch Werte der Ableitung benötigt (die manchmal nicht leicht verfügbar sind), soll noch das Sekantenverfahren diskutiert werden.

2.4 Das Sekantenverfahren

Definition: Das Sekantenverfahren. Gegeben seien zwei Funktionswerte $f(x_{-1})$ und $f(x_0)$ einer stetigen Funktion $f : \mathbb{R} \to \mathbb{R}$. Um eine Nullstelle von f zu finden, approximiere man f durch die Sekante an den gegebenen Punkten und verwende deren Nullstelle als neue Näherung. Dies führt auf die *Iterationsvorschrift*

$$S_n(f) = x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})},$$

die, ähnlich wie beim Newton-Verfahren, nur für gewisse f und Startwerte konvergiert.

Satz 5 (Fehler des Sekantenverfahrens)**.** *Sei $f \in C^2(\mathbb{R})$ mit $f(x^*) = 0$ und $f'(x^*)$ sei verschieden von Null. Sind die Startwerte genügend nahe bei x^*, so folgt für das Sekantenverfahren*

$$\lim_{n\to\infty} e_n = 0 \qquad \textit{und} \qquad \lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \left|\frac{f''(x^*)}{2 \cdot f'(x^*)}\right|^{\alpha-1} \qquad \textit{mit } \alpha = \frac{\sqrt{5}+1}{2}.$$

Beweisidee: Mit dem Mittelwertsatz zeigt man zunächst

$$x_{n+2} - x^* = \frac{f''(\xi_1)}{2f'(\xi_2)}(x_n - x^*)(x_{n+1} - x^*).$$

Falls die Startwerte nahe genug bei x^* liegen folgt hieraus

$$\frac{e_{n+2}}{e_n e_{n+1}} \to \frac{f''(x^*)}{2f'(x^*)}. \tag{2.1}$$

Diese Aussage kann man mit der Theorie der Differenzengleichungen weiter bearbeiten, man betrachte die Folge $(\log|e_n|)_n$. Wir behelfen uns anders. Es gelte zunächst die stärkere Aussage

$$e_{n+2} = ce_{n+1}e_n \tag{2.2}$$

mit $c \neq 0$ und $\lim e_n = 0$, wobei wir annehmen, daß die e_n positiv sind. Der Ansatz $e_{n+1} = A \cdot e_n^\alpha$ führt auf $\alpha = (1+\sqrt{5})/2$ und $A = c^{\alpha-1}$. Aus Stetigkeitsgründen gilt in unserem Fall (wo statt (2.2) nur (2.1) gilt) noch

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \left|\frac{f''(x^*)}{2f'(x^*)}\right|^{\alpha-1}.$$

Bemerkung. Nur wenige Lehrbücher enthalten einen vollständigen Beweis von Satz 5. So wird z.B. in dem Buch von Kincaid und Cheney [11] *angenommen*, daß der Fehler von der Form

$$|e_{n+1}| \approx A\,|e_n|^\alpha$$

ist. Unter dieser Annahme wird dann ausgerechnet, daß

$$\alpha = \frac{1+\sqrt{5}}{2} \approx 1,62 \qquad \text{und} \qquad A = \left|\frac{f''(x^*)}{2f'(x^*)}\right|^{\alpha-1}.$$

Außerdem wird implizit angenommen, daß f sogar dreimal stetig differenzierbar ist. Für einen vollständigen Beweis siehe die Arbeit [32] oder das Lehrbuch [27].

Auch für das Sekantenverfahren könnte man Sätze beweisen, die den Sätzen 3 und 4 ähnlich sind.

Konvergenzordnung. Bei manchen Verfahren (z.B. Newton-Verfahren oder Sekantenverfahren) existiert

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = c_f \neq 0$$

für alle genügend glatten Funktionen mit

$$f(x^*) = 0 \neq f'(x^*) \qquad \text{und} \qquad f''(x^*) \neq 0,$$

wobei α nicht von f abhängt. Dann heißt α *Konvergenzordnung* des Verfahrens. Diese ist zwar beim Sekantenverfahren ($\approx 1,62$) kleiner als beim Newton-Verfahren ($= 2$). Da das Newton-Verfahren jedoch doppelt so viele Informationswerte benötigt, ist das Sekantenverfahren, pro Information gerechnet, schneller.

Zum Begriff der Konvergenzordnung. Beachte: Der Begriff „Konvergenzordnung“ wird später noch in einem anderen Sinn gebraucht. Der Grund besteht im Wesentlichen darin, daß die Fehler bei der Rekonstruktion von Funktionen oder bei der Numerischen Integration meist viel langsamer gegen Null konvergieren, nämlich wie $n^{-\alpha}$ für ein $\alpha > 0$. Bei dieser „polynomialen Konvergenz“ heißt dann auch α Konvergenzordnung!

Einschlußverfahren. Von den bisher diskutierten Verfahren hat das Bisektionsverfahren den Vorteil, als Einschlußverfahren stets zu konvergieren, wenn $f(a) \cdot f(b) < 0$.

Allerdings konvergiert das Bisektionsverfahren nur linear, die Abschätzung von Satz 1 läßt sich i.a. nicht verbessern, auch wenn f „sehr glatt“ ist.

Frage: Gibt es auch schnelle Einschlußverfahren? Genauer fragen wir, ob es Einschlußverfahren gibt, so daß $|e_n|$ für glatte Funktionen (also z.B. mindestens zweimal stetig differenzierbar) mit einer einfachen Nullstelle superlinear konvergiert, d.h. $|e_n| \cdot \delta^n \to 0$ für jedes $\delta > 0$.

Tatsächlich gibt es solche Verfahren, Beispiele für solche „schnelle“ Verfahren sind das Pegasus- oder das Illinois-Verfahren. Ein einfaches Verfahren (mit superlinearer Konvergenz) wird im Folgenden angegeben.

Beispiel: Das hybride Verfahren. Zunächst zur regula falsi: Diese kombiniert Bisektions- und Sekantenverfahren. Zu zwei Startwerten $f(a)$ und $f(b)$ mit $f(a) \cdot f(b) < 0$ wird mit dem Sekantenverfahren ein weiterer bestimmt. Nun werden im Iterationsschritt nicht die beiden letzten Werte weiterverwendet, sondern die beiden, zwischen denen sicher die Nullstelle liegt (Vorzeichenuntersuchung wie beim Bisektionsverfahren).

Beim einfachsten hybriden Verfahren macht man in einem Schritt zunächst einen Schritt nach dem Bisektionsverfahren und dann einen nach der regula falsi. Man kann zeigen, daß dieses hybride Verfahren für glatte Funktionen mit einfacher Nullstelle superlinear konvergiert. Dagegen konvergieren Bisektionsverfahren und regula falsi für sich genommen jeweils nur linear.

2.5 Der Banach'sche Fixpunktsatz

Viele Verfahren zur numerischen Lösung von Gleichungen sind von der Form $x_{n+1} = F(x_n)$. Man hofft, daß ein x^* existiert mit $\lim_{n\to\infty} x_n = x^*$. Ist außerdem F noch stetig, so gilt

$$F(x^*) = F(\lim_{n\to\infty} x_n) = \lim_{n\to\infty} F(x_n) = \lim_{n\to\infty} x_{n+1} = x^*.$$

Das heißt, x^* ist Fixpunkt von F. Der nachfolgende *Banach'sche Fixpunktsatz* hilft oft bei Fragen nach der Konvergenz dieser Verfahren.

Satz 6 (Banach'scher Fixpunktsatz)**.** *Sei M ein vollständiger metrischer Raum und $F : M \to M$ eine Kontraktion, d.h. es gibt ein $L < 1$ mit*

$$d(F(x), F(y)) \leq L \cdot d(x, y)$$

für $x, y \in M$. Dann hat F genau einen Fixpunkt x^* und die durch $x_{n+1} = F(x_n)$ definierte Folge konvergiert für jeden Startwert $x_0 \in M$. Außerdem gilt für $0 \leq m < n$ die Fehlerabschätzung

$$d(x^*, x_n) \leq \frac{L^{n-m}}{1-L} \cdot d(x_{m+1}, x_m).$$

Bemerkung: Anwendung für $F : \mathbb{R} \to \mathbb{R}$. Wenn $F : \mathbb{R} \to \mathbb{R}$ stetig differenzierbar ist, gilt

$$e_{n+1} = x_{n+1} - x^* = F(x_n) - F(x^*) = F'(\xi_n) \cdot (x_n - x^*) = F'(\xi_n) \cdot e_n,$$

wobei ξ_n zwischen x_n und x^* liegt. Damit ist offenbar der Fall $|F'(x^*)| < 1$ notwendig, um lokal den Banach'schen Fixpunktsatz anwenden zu können. Besonders günstig ist $F'(x^*) = 0$. Wir nehmen für ein q-mal ($q \geq 2$) stetig differenzierbares F an, daß

$$0 = F(x^*) = F'(x^*) = \cdots = F^{(q-1)}(x^*) \qquad \text{und} \qquad F^{(q)}(x^*) \neq 0.$$

Dann gilt für ein ξ_n zwischen x_n und x^*

$$e_{n+1} = x_{n+1} - x^* = F(x_n) - F(x^*) = \sum_{k=1}^{q-1} \frac{1}{k!} e_n^k F^{(k)}(x^*) + \frac{1}{q!} e_n^q F^{(q)}(\xi_n) = \frac{1}{q!} e_n^q F^{(q)}(\xi_n).$$

Wenn die Folge (x_n) gegen x^* konvergiert, dann folgt:

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^q} = \frac{1}{q!} \cdot |F^{(q)}(x^*)| \neq 0.$$

Also ist q die Konvergenzordnung des Verfahrens.

Lokalität und Asymptotik der Aussage. Die Konvergenzordnung beschreibt das Verhalten der Fehler $|e_n|$ nur „lokal“ und „asymptotisch“. Sobald der Startwert vom Fixpunkt zu weit entfernt ist – und jedes $\varepsilon > 0$ kann im Einzelfall schon eine zu große Abweichung sein – kann nicht einmal die Konvergenz selbst gefolgert werden. Zudem gilt die Konvergenzaussage auch bei konvergenten Folgen nur asymptotisch – damit sind Aussagen über die ersten Folgenglieder nicht möglich. Da man jedoch immer nur endlich viele Werte tatsächlich berechnen kann, sollte man die Bedeutung der Konvergenzordnung nicht überbewerten.

2.6 Nullstellen von Polynomen

Besonders einfache Funktionen sind die Polynome

$$p(x) = \sum_{i=0}^{n} a_i \cdot x^i.$$

Sie lassen sich zum Beispiel leicht abspeichern, differenzieren und integrieren. Trotzdem kann man die Nullstellen von p nur für grad $p \leq 4$ exakt bestimmen, man benötigt daher Näherungsverfahren. Beliebt ist insbesondere das Newton-Verfahren (und gewisse Varianten hiervon). Es entsteht das Problem, geeignete Startwerte zu finden.

Wichtig ist der *Fundamentalsatz der Algebra* (Gauß 1799): Jedes nichtkonstante komplexe Polynom hat eine komplexe Nullstelle. Daher läßt sich jedes Polynom vom Grad n schreiben als

$$p(x) = c \cdot \prod_{i=1}^{n} (x_i - x)$$

mit den Nullstellen $x_i \in \mathbb{C}$. Insbesondere hat p also (falls $p \neq 0$) höchstens n verschiedene Nullstellen.

Satz 7 (Lage der Nullstellen von Polynomen)**.** *Sei* $p(x) = \sum_{i=0}^{n} a_i \cdot x^i$ *mit* $a_i \in \mathbb{C}$ *und* $a_n \neq 0$*. Dann liegen alle komplexen Nullstellen von* p *in der abgeschlossenen Kreisscheibe um* 0 *mit Radius*

$$R = 1 + |a_n|^{-1} \cdot \max_{0 \leq k < n} |a_k|.$$

Das Horner-Schema. Eine naive Berechnung der Funktionswerte von

$$p(x) = \sum_{i=0}^{n} a_i \cdot x^i$$

erfordert zuviele Rechenoperationen. Der Algorithmus („Horner-Schema“)

$$p(x) = a_0 + x \cdot (a_1 + x \cdot (a_2 + \dots))$$

ist wesentlich besser. Man erhält so eine rechnerisch vorteilhafte Klammerung des ursprünglichen Polynoms. Durch leichte Modifikation des Verfahrens kann man auch Werte der Ableitungen von p berechnen, ohne die abgeleiteten Polynome explizit darstellen zu müssen.

Zusätze und Bemerkungen.

- In der Arbeit [13] wird gezeigt, daß das Newton-Verfahren für F wie im Satz 4 nahezu optimal ist.
- Einschließungsverfahren mit hoher Konvergenzordnung werden in [5] beschrieben und analysiert.
- Weitere Ergebnisse findet man in [21].

Aufgaben

2.1. Schreiben Sie ein Computerprogramm zur Berechnung einer Nullstelle einer differenzierbaren Funktion $f : [0, 1] \to \mathbb{R}$ mit Vorzeichenwechsel mit Hilfe der besprochenen Verfahren: Bisektion, Newton, Sekanten, regula falsi und hybrides Verfahren.

Abbruchkriterium: $n = 200$ oder $|f(x_n)| \leq 10^{-10}$. Im Fall $x_n \notin [0, 1]$ soll eine Fehlermeldung ausgegeben werden.

Testen Sie Ihr Programm anhand der Funktionen

$$f_k(x) = e^{kx} - e^{2k/7} \qquad \text{und} \qquad g_k(x) = \frac{1 - 3x}{1 + kx}$$

mit k gleich 2 oder 5 oder 10 und den Startwerten 0 und 1 (bzw. 0 oder 1 beim Newtonverfahren).

2.2. Mit dem Newtonverfahren und $f(x) = x^2 - a$ soll die Wurzel von $a > 0,006$ berechnet werden. Der Startwert x_0 sei positiv. Zeigen Sie: Ist x_n für ein $n \geq 1$ auf k Stellen hinter dem Komma genau (das heißt für den Fehler e_n gilt $|e_n| < \frac{1}{2}10^{-k}$), so ist x_{n+1} auf mindestens $(2k - 1)$ Stellen nach dem Komma genau.

2.3. Sei

$$f(x) = x^3 - 3x^2 + x - 1.$$

Zeigen Sie: Es gibt ein Intervall $I \subset \mathbb{R}$ mit der Eigenschaft, daß das Newtonverfahren (angewandt auf f) mit beliebigem Startwert $x_0 \in I$ nicht konvergiert.

2.4. Gegeben sei die Abbildung $T : C([0,1]) \to C([0,1])$ durch

$$Tf(x) = \int_0^1 \frac{1}{2}(xy - 1) \cdot (f(y) + 1)dy.$$

a) Zeigen Sie, daß T genau einen Fixpunkt hat.
b) Sei $f_0 = 0$ und $f_{n+1} = Tf_n$. Bestimmen Sie f_1 und f_2.
c) Bestimmen Sie den Fixpunkt von T.

2.5. Gegeben sei das Gleichungssystem

$$f_1(x,y) = \frac{1}{6}\sin y + \frac{1}{3}\cos x - \frac{1}{6}y = x$$

$$f_2(x,y) = \frac{1}{5}\cos y - \frac{1}{4}\sin x + \frac{1}{4}x = y.$$

a) Zeigen Sie, daß dieses Gleichungssystem genau eine Lösung $(x^*, y^*) \in \mathbb{R}^2$ hat.
b) Sei $(x_0, y_0) = (0,0)$ und $(x_{n+1}, y_{n+1}) = (f_1(x_n, y_n), f_2(x_n, y_n))$. Berechnen Sie (x_n, y_n) für $n = 1$, 2, 3 und 4.

2.6. Gegeben sei die Integralgleichung

$$f(x) = c + \int_0^1 k(x,y) \cdot f(y)^{-2} dy \qquad (0 \le x \le 1)$$

mit $c > 2^{1/3}$ und einer stetigen Funktion k mit $0 \le k \le 1$. Zeigen Sie, daß eine Lösung $f \in C([0,1])$ dieser Gleichung existiert.

Hinweis: Wenden Sie den Fixpunktsatz von Banach auf eine geeignete Teilmenge von $C([0,1])$ an.

Kapitel 3

Rekonstruktion von Funktionen

In diesem Kapitel geht es um Fragen, die in der Literatur häufig mit den Stichworten „Interpolation und Ausgleichsrechnung“ umschrieben werden. Wir bevorzugen den Namen „Rekonstruktion von Funktionen“, weil damit das Wesen der Fragestellung besser beschrieben wird.

Das Problem der Rekonstruktion von Funktionen tritt in verschiedenen Formen auf, wir geben die beiden wichtigsten Beispiele an.

– Näherung bekannter Funktionen. Hier besteht das Problem darin, eine gegebene stetige Funktion $f : [a, b] \to \mathbb{R}$ abzuspeichern, wenn kein einfacher Rechenausdruck bekannt ist.

Man kann zum Beispiel $n \in \mathbb{N}$ Funktionswerte abspeichern und, falls der Wert von f an einer weiteren Stelle gesucht ist, die Werte „glatt“ verbinden, um so eine Näherung von f zu erhalten.

– Ermittlung unbekannter Funktionen durch Meßwerte. Wie findet man zu einigen fehlerbehafteten Meßwerten diejenige Funktion, die diese Werte unter Berücksichtigung von möglichen Fehlern am besten approximiert?

Hierbei ist darauf zu achten, daß es wegen der Fehler nicht unbedingt von Vorteil ist, wenn die konstruierte Funktion die Meßwerte genau interpoliert. Vielmehr wird man häufiger solche Funktionen bevorzugen, die besonders glatt sind, aber kaum einen Meßwert direkt treffen. Probleme dieser Art heißen auch „Ausgleichsprobleme“, insbesondere dann, wenn bekannt ist, daß f Element eines m-dimensionalen Raumes ist und $n > m$ fehlerbehaftete Funktionswerte bekannt sind, so daß ein überbestimmtes Gleichungssystem entsteht.

3.1 Polynomiale Interpolation

Besonders einfach ist die Interpolation durch Polynome. Gesucht ist ein Polynom kleinsten Grades mit $p(x_i) = y_i$ für $i = 1, \ldots, n$. So ein Polynom heißt Interpolationspolynom, wir zeigen zunächst seine Existenz und Eindeutigkeit. Dazu sei

$$P_n = \left\{ p \mid p(x) = \sum_{i=0}^{n-1} a_i x^i,\ a_i \in \mathbb{R} \right\}$$

die Menge der Polynome vom Grad kleiner n. Die Bezeichnung ist so gewählt, daß wir n Interpolationsbedingungen haben und entsprechend gilt $\dim(P_n) = n$.

Satz 8 (Existenz und Eindeutigkeit des Interpolationspolynoms)**.** *Gegeben seien paarweise verschiedene reelle Zahlen $x_1, \ldots, x_n$ und beliebige $y_i \in \mathbb{R}$. Dann existiert genau ein Polynom $p \in P_n$ mit $p(x_i) = y_i$ für $i = 1, \ldots, n$.*

Das Polynom p kann in der Form

$$p(x) = \sum_{i=1}^{n} c_i \cdot \prod_{j=1}^{i-1} (x - x_j)$$

geschrieben werden. Dabei benutzen wir die Konvention $\prod_{\emptyset} = 1$. Diese Form heißt die Newton'sche Form des Interpolationspolynoms. Ist p auf diese Weise gegeben, so lassen sich einzelne Funktionswerte $p(x)$ wie beim Horner-Schema berechnen durch

$$p(x) = c_1 + (x - x_1)(c_2 + (x - x_2)(c_3 + \ldots)).$$

Die Lagrange-Form des Interpolationspolynoms. Es gibt zwar nur ein Interpolationspolynom, aber verschiedene Schreibweisen. Mit den Bezeichnungen des obigen Satzes gilt

$$p(x) = \sum_{k=1}^{n} y_k \prod_{j \neq k} \frac{x - x_j}{x_k - x_j} =: \sum_{k=1}^{n} y_k \cdot l_k(x).$$

Die l_k sind diejenigen Polynome, für die $l_k(x_j) = \delta_{kj}$ gilt.

Vandermonde-Matrizen. Will man das Interpolationspolynom p in der Form $p(x) = \sum_{i=0}^{n-1} a_i x^i$ darstellen, so muß man das Gleichungssystem

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

lösen. Die zugehörige Matrix heißt Vandermonde-Matrix. Man kann zeigen, daß für die Vandermonde-Matrix V

$$\det V = \prod_{1 \leq j < k \leq n} (x_k - x_j) \neq 0$$

gilt. Da die Vandermonde-Matrix jedoch schlecht konditioniert ist und die dadurch ermittelte Form des Interpolationspolynoms selten gebraucht wird, berechnet man selten die Lösungen dieses Systems. Für numerische Zwecke ist i.a. die Newton'sche Form am besten geeignet, für analytische Fragestellungen verwendet man in der Regel die Lagrange-Form.

Fehlerabschätzung bei der polynomialen Interpolation. Wir beginnen mit dem folgenden Satz.

Satz 9 (Fehlerabschätzung bei der Interpolation durch Polynome)**.** *Sei $f \in C^n([a,b])$, wobei $n \in \mathbb{N}$, sowie $x_1, \ldots, x_n \in [a,b]$ paarweise verschieden. Weiter sei $p \in P_n$ das Interpolationspolynom mit $p(x_i) = f(x_i)$ für $i = 1, \ldots, n$. Dann existiert für jedes $x \in [a,b]$ ein $\xi_x \in [a,b]$ mit*

$$f(x) - p(x) = \frac{1}{n!} \cdot f^{(n)}(\xi_x) \cdot \prod_{i=1}^{n} (x - x_i).$$

Zur Wahl der Knoten. Unter den Voraussetzungen von Satz 9 gilt die Fehlerabschätzung

$$\|f - p\|_\infty \leq \frac{1}{n!} \cdot \|f^{(n)}\|_\infty \cdot \left\| \prod_{i=1}^{n} (x - x_i) \right\|_\infty .$$

Wie muß man die Knoten wählen, damit

$$\left\| \prod_{i=1}^{n} (x - x_i) \right\|_\infty$$

möglichst klein wird? Solche Knoten sind optimal in dem Sinn, daß die Abschätzung von Satz 9 möglichst gut ist. Wir werden diese Frage im Satz 12 beantworten. Dazu definieren wir zunächst die Tschebyscheff-Polynome (1. Art) und beweisen einige ihrer Eigenschaften.

Tschebyscheff-Polynome (1. Art). Diese Polynome sind rekursiv definiert. Es gilt $T_0(x) = 1$ und $T_1(x) = x$ und

$$T_{n+1}(x) = 2x\, T_n(x) - T_{n-1}(x)$$

für $n > 1$. Trotz dieser rekursiven Definition kann man diese Polynome auch explizit angeben, es gilt folgende Aussage.

Satz 10 (Formel für die Tschebyscheff-Polynome)**.** *Sei $x \in [-1,1]$. Dann gilt für die Tschebyscheff-Polynome $T_n \in P_{n+1}$*

$$T_n(x) = \cos(n \cdot \arccos(x)).$$

Aus dieser Darstellung der T_n erhält man sofort die folgenden Eigenschaften. Es gilt $|T_n(x)| \leq 1$ für alle $x \in [-1,1]$,

$$T_n\left(\cos \frac{k\pi}{n}\right) = (-1)^k \qquad \text{für } k = 0, \ldots, n \text{ und}$$

$$T_n\left(\cos \frac{(2k-1)\pi}{2n}\right) = 0 \qquad \text{für } k = 1, \ldots, n.$$

Da T_n als Polynom n-ten Grades höchstens n Nullstellen haben kann, sind alle Nullstellen von T_n gegeben durch $x_k = \cos \frac{(2k-1)\pi}{2n}$, wobei man sich wegen der Periodizität der Cosinus-Funktion auf $k = 1, \ldots, n$ beschränken kann. Alle Nullstellen von T_n liegen im Intervall $[-1,1]$. Mit Hilfe dieser Eigenschaften läßt sich der folgende Satz zeigen.

Satz 11 (Normabschätzung für normierte Polynome)**.** *Sei* $p_{n+1}(x) = \sum_{i=0}^{n} a_i x^i$ *ein Polynom vom Grad* $n > 0$ *mit* $a_n = 1$*. Dann gilt*

$$\|p_{n+1}\|_\infty = \sup_{x\in[-1,1]} |p_{n+1}(x)| \geq 2^{1-n}.$$

Folgerungen. Sei $q \in P_{n+1}$ von der Form $q(x) = \prod_{i=1}^{n}(x - x_i)$ mit $x_i \in [-1, 1]$. Dann gilt

$$\|q\|_\infty = \sup_{x\in[-1,1]} |q(x)| \geq 2^{1-n}.$$

Das normierte Tschebyscheff-Polynom $q = 2^{1-n} \cdot T_n$ hat minimale Norm, es gilt

$$\|q\|_\infty = 2^{1-n} \qquad \text{und} \qquad q(x) = \prod_{i=1}^{n} \left(x - \cos\left(\frac{2i-1}{2n}\pi \right) \right).$$

Das heißt, die Nullstellen des Tschebyscheff-Polynoms sind die idealen Knoten für die polynomiale Interpolation.

Satz 12 (Optimalität der Tschebyscheff-Knoten)**.** *Sei* $f \in C^n([-1, 1])$ *und sei* $p_n \in P_n$ *das Polynom mit* $p_n(x_i) = f(x_i)$ *für die „Tschebyscheff-Knoten“*

$$x_i = \cos\left(\frac{2i-1}{2n}\pi \right),$$

mit $i = 1, \ldots, n$*. Dann gilt*

$$\|f - p_n\|_\infty \leq \frac{2^{1-n}}{n!} \cdot \|f^{(n)}\|_\infty$$

und die Knoten sind optimal, d.h. für andere Knoten gilt diese Abschätzung i.a. nicht.

Bemerkungen zur Konvergenz der Interpolationspolynome. Für eine stetige Funktion $f \in C([-1, 1])$ sei $p_n \in P_n$ das Interpolationspolynom zu den Knoten $-1 \leq x_1^{(n)} < \cdots < x_n^{(n)} \leq 1$. Kann man die Knoten so legen, daß

$$\lim_{n\to\infty} \|f - p_n\|_\infty = 0$$

für jede stetige Funktion gilt? Nach dem Satz von Weierstraß gibt es ja bekanntlich für jedes $f \in C([-1, 1])$ eine Folge $p_n \in P_n$ mit $\lim_{n\to\infty} \|f - p_n\|_\infty = 0$.

Der folgende Satz von Faber (ohne Beweis) zeigt, daß es keine Knotenfolge für Interpolationsverfahren gibt, die für jede stetige Funktion eine konvergente Folge ergibt.

Satz 13 (Satz von Faber (1914))**.** *Seien die Knoten* $x_i^{(n)}$ *beliebig gewählt. Dann gibt es eine Funktion* $f \in C([-1, 1])$ *mit der Eigenschaft, daß die Folge der Interpolationspolynome* p_n *nicht gegen* f *konvergiert.*

Wenn man nicht alle $f \in C([-1,1])$ betrachtet, kann man als positives Gegenstück zum Satz von Faber z.B. den folgenden Satz (ohne Beweis) zeigen.

Satz 14 (Konvergenz der Interpolationspolynome bei Tschebyscheff-Knoten und Lipschitz-Funktionen)**.** *Sei $f \in C([-1,1])$ Lipschitz-stetig. Dann konvergieren die Interpolationspolynome zu den Tschebyscheff-Knoten gleichmäßig gegen f.*

Vergleich mit äquidistanten Knoten. Für äquidistante Knoten gilt das letzte Ergebnis nicht. Es gibt dann sogar analytische Funktionen f, so daß die Interpolationspolynome nicht gegen f konvergieren. Bekannt ist das Beispiel der Runge-Funktion,

$$f : [-1,1] \to \mathbb{R}, \qquad f(x) = \frac{1}{1+25x^2},$$

siehe Aufgabe 3.3.

Die verschiedenen Strategien zur Knotenwahl kann man auch noch anders vergleichen. Gegeben sei die Lagrange-Darstellung des Interpolationspolynoms

$$p_n(x) = \sum_{k=1}^{n} y_k \cdot l_k(x)$$

mit

$$l_k(x) = \prod_{j \neq k} \frac{x - x_j}{x_k - x_j}.$$

Die (absoluten) Konditionszahlen der Abbildung

$$S(f) = p_n, \qquad S : C([-1,1]) \to P_n \subset C([-1,1])$$

kann man schreiben als

$$K_{\text{abs}} = \sup_{x \in [-1,1]} \sum_{k=1}^{n} |l_k(x)|.$$

Sie geben an, wie sehr sich Änderungen bei f auf das Ergebnis (d.h. auf das Interpolationspolynom p_n) auswirken können. Die Zahl K_{abs} hängt natürlich von den Knoten und von n ab. Hier einige Werte.

n	K_{abs} bei Tscheb. Knoten	K_{abs} bei äquid. Knoten
6	2,10	3,11
11	2,49	29,89
16	2,73	512,05
21	2,90	10.986,53

Auch diese Ergebnisse zeigen, daß die Tschebyscheff-Knoten besser sind als äquidistante Knoten. Die Zahlen K_{abs} heißen auch Lebesgue-Konstanten. Man kann zeigen, daß diese Zahlen im Fall der Tschebyscheff-Knoten wie $\log n$ wachsen, im Fall äquidistanter Knoten dagegen exponentiell. Siehe etwa [24, S. 90–101].

Dividierte Differenzen. Sei f eine Funktion, $x_1, \ldots, x_n$ seien verschiedene Knoten und $p \in P_n$ sei das Polynom mit $p(x_i) = f(x_i)$. Die Newton-Form von p ist

$$p(x) = \sum_{j=1}^{n} c_j \cdot \prod_{i=1}^{j-1}(x - x_i) =: \sum_{j=1}^{n} c_j \cdot q_j(x).$$

Die Interpolationsbedingungen ergeben ein lineares Gleichungssystem für die c_j:

$$\sum_{j=1}^{n} c_j \cdot q_j(x_i) = f(x_i), \qquad 1 \le i \le n.$$

Die Koeffizientenmatrix ist $A = (A_{ij}) = (q_j(x_i))$, die Unbekannten sind $c_1, \ldots, c_n$. Die Matrix A ist eine untere Dreiecksmatrix, daher lassen sich die c_i leicht in der Reihenfolge $c_1, c_2, \ldots, c_n$ ausrechnen. Außerdem hängt c_k nur ab von $x_1, \ldots, x_k$ und den entsprechenden Funktionswerten $f(x_1), \ldots, f(x_k)$.

Definition (dividierte Differenzen)**.** Wir definieren

$$f[x_1, \ldots, x_k] := c_k.$$

Man nennt $f[x_1, \ldots, x_k]$ *dividierte Differenz.*

Satz 15 (Eigenschaften der dividierten Differenzen)**.** *Die dividierten Differenzen genügen den Gleichungen*

$$f[x_1] = f(x_1) \qquad \textit{und}$$
$$f[x_1, \ldots, x_n] = \frac{f[x_2, \ldots, x_n] - f[x_1, \ldots, x_{n-1}]}{x_n - x_1}.$$

Die dividierten Differenzen sind symmetrisch in ihren Argumenten, das heißt für jede Permutation π von $\{1, 2, \ldots, n\}$ gilt

$$f[x_1, \ldots, x_n] = f[x_{\pi(1)}, \ldots, x_{\pi(n)}].$$

Sei $p \in P_n$ das Polynom, das f an den paarweise verschiedenen Knoten $x_1, \ldots, x_n$ interpoliert. Dann gilt für x, das mit keinem der Knoten übereinstimmt,

$$f(x) - p(x) = f[x_1, \ldots, x_n, x] \cdot \prod_{j=1}^{n}(x - x_j).$$

Ist zusätzlich $f \in C^{n-1}([a, b])$, so gilt

$$f[x_1, \ldots, x_n] = \frac{1}{(n-1)!} f^{(n-1)}(\xi) \qquad \textit{für ein } \xi \in [a, b].$$

Hermite-Interpolation. Gesucht sei ein Polynom von möglichst kleinem Grad mit

$$p^{(j)}(x_i) = c_{ij} \qquad \text{für } i = 1, \ldots, n \quad j = 0, \ldots, k_i - 1.$$

Dieses Problem heißt Hermite'sches Interpolationsproblem. Im Gegensatz dazu heißt das bisher betrachtete Problem (mit $k_i = 1$ für alle i) auch Lagrange'sches Interpolationsproblem.

Die Anzahl der Bedingungen ist also $m := \sum_{i=1}^{n} k_i$.

Satz 16 (Lösbarkeit des Hermite'schen Interpolationsproblems)**.** *Mit den obigen Bezeichnungen gilt: Es gibt genau ein Polynom $p \in P_m$, das das Hermite'sche Interpolationsproblem löst.*

Beispiel. Wir wollen das Hermite'sche Interpolationsproblem im Fall $k_i = 2$ für $i = 1, \ldots, n$ diskutieren. In Analogie zur Lagrange-Darstellung beim Fall $k_i = 1$ macht man den Ansatz

$$p(x) = \sum_{i=1}^{n} c_{i0} A_i(x) + \sum_{i=1}^{n} c_{i1} B_i(x).$$

Man sucht also nach Polynomen mit den Eigenschaften

$$A_i(x_j) = \delta_{ij}, \qquad B_i(x_j) = 0$$

und

$$A_i'(x_j) = 0, \qquad B_i'(x_j) = \delta_{ij}.$$

Mit Hilfe der Lagrange-Polynome $l_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$ kann man definieren

$$A_i(x) = [1 - 2(x - x_i) l_i'(x_i)] \cdot l_i^2(x)$$

und

$$B_i(x) = (x - x_i) \cdot l_i^2(x).$$

Da die l_i vom Grad $n - 1$ sind, haben die Polynome A_i und B_i höchstens den Grad $2n - 1$. Da sie, wie man durch Nachrechnen bestätigt, die gewünschten Eigenschaften haben, hat man also nach obigem Satz die Lösung des Interpolationsproblems gefunden.

Satz 17 (Fehler bei der Hermite-Interpolation)**.** *Seien $x_1, \ldots, x_n$ verschiedene Knoten in $[a, b]$ und sei $f \in C^{2n}([a, b])$. Sei $p \in P_{2n}$ das Polynom mit $p^{(j)}(x_i) = f^{(j)}(x_i)$ für $i = 1, \ldots, n$ und $j = 0, 1$. Dann existiert zu jedem $x \in [a, b]$ ein $\xi_x \in [a, b]$ mit*

$$f(x) - p(x) = \frac{f^{(2n)}(\xi_x)}{(2n)!} \cdot \prod_{i=1}^{n} (x - x_i)^2.$$

Nochmals dividierte Differenzen. Hier eine allgemeine Definition der dividierten Differenzen, die auch den Fall mehrerer gleicher Argumente abdeckt. Es gelte $f \in C^{n-1}(\mathbb{R})$ und $x_1 \leq x_2 \leq \ldots \leq x_n$. Dann setzt man

$$f[x_1, \ldots, x_n] = \begin{cases} \dfrac{f[x_2, \ldots, x_n] - f[x_1, \ldots, x_{n-1}]}{x_n - x_1}, & \text{falls } x_n \neq x_1 \\ \dfrac{1}{(n-1)!} f^{(n-1)}(x_1), & \text{falls } x_n = x_1. \end{cases}$$

Für beliebige x_i definieren wir die dividierte Differenz $f[x_1, \ldots, x_n]$ durch die Forderung, daß wir die Argumente vertauschen dürfen.

Satz 18 (Hermite-Interpolation und dividierte Differenzen)**.** *Das allgemeine Hermite'sche Interpolationsproblem wird gelöst durch* $p_m \in P_m$*,*

$$p_m(x) = \sum_{j=1}^{m} f[x_1, \ldots, x_j] \cdot \prod_{i=1}^{j-1} (x - x_i),$$

wobei die Knoten anders bezeichnet werden als bisher. Jeder bisherige Knoten x_i *soll genau* k_i*-mal in der Folge der neuen Knoten auftreten,* $m = \sum k_i$*. Die zu interpolierende Funktion* f *sei hinreichend glatt, so daß alle auftretenden dividierten Differenzen definiert sind.*

Zusammenfassung und Bemerkungen. Die Ergebnisse zeigen, daß die polynomiale Interpolation gut funktioniert, falls man die Tschebyscheff-Knoten benützt. Dies kann man weiter begründen:

Nach dem Satz von Weierstraß liegen die Polynome dicht im Raum $C([a, b])$. Der Satz von Jackson, siehe z.B. [15], ist eine quantitative Version dieser Tatsache. Dazu sei

$$E_n(f) = \inf_{p \in P_n} \|p - f\|_\infty = \inf_{p \in P_n} \sup_{x \in [a,b]} |p(x) - f(x)|$$

der Abstand von f zum Raum P_n.

Satz 19 (Satz von Jackson)**.** *Für* $k \in \mathbb{N}$ *existiert* $c_k > 0$ *so daß für alle* $f \in C^k([-1, 1])$ *und* $n \geq k$ *gilt*

$$E_n(f) \leq c_k \cdot \|f^{(k)}\|_\infty \cdot n^{-k}.$$

Ersetzt man hier das Polynom der besten Approximation durch das Polynom p_n, das man durch Interpolation an den Tschebyscheff-Knoten erhält, so erhält man die nur wenig schlechtere Abschätzung

$$\|f - p_n\|_\infty \leq \widetilde{c}_k \cdot \|f^{(k)}\|_\infty \cdot n^{-k} \cdot \log(n + 1).$$

In den Anwendungen sind allerdings die Knoten häufig vorgegeben und nicht frei wählbar, sondern z.B. äquidistant. Dann ist die polynomiale Interpolation kein geeignetes Mittel zur Rekonstruktion von Funktionen.

3.2 Spline-Interpolation

Vorbemerkung. Ursprünglich war es unser Ziel, eine Funktion anhand von einzelnen Funktionswerten mit geringem Fehler zu rekonstruieren. Dazu will man zu gegebenen Daten $f(x_i) = y_i$ für $i = 1, \ldots, n$ eine möglichst glatte Funktion $\widetilde{f}$ mit $\widetilde{f}(x_i) = y_i$ finden. Die Interpolationspolynome sind zwar einfach zu berechnen, doch Beispiele (Aufgabe 3.3) und der Satz von Faber zeigen, daß sie i.a. nicht zu einer „glatten“ Approximation führen. Dabei ist allerdings bisher nicht definiert worden, was glatt heißen soll. In diesem Abschnitt verwenden wir

$$\int_a^b s^{(m)}(x)^2 \, dx$$

als Maß für die Glattheit von s. Wir versuchen also, eine interpolierende Funktion s zu finden, so daß dieses Integral möglichst klein ist. Dieses Extremalproblem führt auf Spline-Funktionen, d.h. auf stückweise polynomiale Funktionen. Besonders einfach und wichtig sind die Fälle $m = 1$ und $m = 2$, die getrennt besprochen werden.

Die Zahl

$$\|s\|_2 = \Big(\int_a^b s(x)^2\,dx\Big)^{1/2}$$

heißt L_2-Norm von s und entsprechend ist

$$\int_a^b s^{(m)}(x)^2\,dx$$

das *Quadrat der L_2-Norm der m-ten Ableitung von s.*

Definition. Es seien $a = x_1 < x_2 < \cdots < x_n = b$ vorgegebene Knoten, $k \in \mathbb{N}_0$. Ein Spline vom Grad k mit den Knoten $x_1, \ldots, x_n$ ist eine Funktion s mit $s|_{[x_i, x_{i+1}[} \in P_{k+1}$ und $s \in C^{k-1}([a,b])$, falls $k \neq 0$. Stets sei $n \geq 2$.

Beispiele. Splines vom Grad 0 sind Treppenfunktionen. Splines vom Grad 1 sind stückweise lineare Funktionen. Splines vom Grad 3 heißen kubische Splines.

Satz 20 (Optimalitätseigenschaft linearer Splines)**.** *Sei $f \in C([a,b])$ stückweise stetig differenzierbar und sei $a = x_1 < x_2 \cdots < x_n = b$. Dann existiert genau ein linearer Spline s mit den Knoten $x_1, \ldots, x_n$, der die Daten interpoliert, d.h. s ist stetig und stückweise linear mit $s(x_i) = f(x_i)$. Weiter gilt*

$$\int_a^b s'(x)^2\,dx \leq \int_a^b f'(x)^2\,dx,$$

d.h. s ist die interpolierende stetige und stückweise C^1-Funktion mit minimaler L_2-Norm der ersten Ableitung.

Beweis: Sei $g = f - s$. Dann gilt $g(x_i) = 0$ für alle i und

$$\int_a^b (f')^2 dx = \int_a^b (s')^2 dx + \int_a^b (g')^2 dx + 2\int_a^b s'g'dx.$$

Wir sind fertig, wenn wir zeigen können, daß

$$\int_a^b s'g'(x)\,dx \geq 0.$$

Dies folgt aus

$$\int_a^b s'g'\,dx = \sum_{i=2}^n \int_{x_{i-1}}^{x_i} s'g'\,dx = \sum_{i=2}^n c_i \int_{x_{i-1}}^{x_i} g'\,dx = 0,$$

da die Steigung von s in jedem Intervall $[x_{i-1}, x_i]$ konstant ist und wegen

$$\int_{x_{i-1}}^{x_i} g'\,dx = g(x_i) - g(x_{i-1}) = 0.$$

Satz 21 (Fehler bei der Interpolation durch lineare Splines)**.** *Die Voraussetzungen seien wie bei Satz 20, zusätzlich sei $f \in C^2([a,b])$ und es sei*

$$h := \max_{i=2,\ldots,n} |x_i - x_{i-1}|$$

die sog. Feinheit der Zerlegung $a = x_1 < \cdots < x_n = b$. Dann gilt für s die Fehlerabschätzung

$$\|f - s\|_\infty \leq \frac{h^2}{8} \cdot \|f^{(2)}\|_\infty.$$

Beweis: Dies folgt aus Satz 9: Für x zwischen x_i und x_{i+1} gilt

$$f(x) - s(x) = \frac{1}{2} f''(\xi) \cdot (x - x_i) \cdot (x - x_{i+1}),$$

daraus folgt die Behauptung wegen

$$|(x - x_i)(x - x_{i+1})| \leq \frac{h^2}{4}.$$

Wir haben gesehen, daß eine ähnliche Fehlerabschätzung für die polynomiale Interpolation nicht gilt. Der Fehler hängt dort sehr stark von den Knoten ab, nicht nur von der Feinheit h.

Kubische Splines. Sei s ein kubischer Spline, der die Daten interpoliert, d.h. $s(x_i) = y_i$ für $i = 1, \ldots, n$. Sei $s_i(x) = s(x)$ für $x \in [x_i, x_{i+1}]$ und $i = 1, \ldots, n-1$. Dann ist $s_i \in P_4$. Die 4 Koeffizienten für jedes s_i ergeben zusammen $4n - 4$ Unbekannte. Demgegenüber haben wir $2n - 2$ Bedingungen wegen

$$s_i(x_{i+1}) = s_{i+1}(x_{i+1}) = y_{i+1},$$

$i = 1, \ldots, n-2$, und $s_1(x_1) = y_1$ sowie $s_{n-1}(x_n) = y_n$. Dazu kommen $2n-4$ Bedingungen wegen

$$s_i^{(k)}(x_{i+1}) = s_{i+1}^{(k)}(x_{i+1})$$

für $i = 1, \ldots, n-2$ und $k = 1$ oder $k = 2$. Gibt es einen (bzw. mehr als einen) kubischen Spline, der die vorgegebenen Daten interpoliert?

Man hat einerseits $4n - 4$ Unbekannte, andererseits nur $4n - 6$ Bedingungen. Dies heißt: Wenn es eine Lösung gibt, dann ist sie sicher nicht eindeutig. Wir werden gleich sehen, daß für paarweise verschiedene Knoten immer eine Lösung existiert. Es zeigt sich, daß

$$V := \{s \mid s \text{ ist kubischer Spline zu den Knoten } x_1 < \cdots < x_n\}$$

ein Vektorraum der Dimension $n + 2$ ist. Die Lösungen des Interpolationsproblems bilden in diesem Raum einen 2-dimensionalen affinen Unterraum.

Zur praktischen Berechnung von s versucht man zunächst, die Zahlen $z_i := s''(x_i)$ für $1 \leq i \leq n$ zu bestimmen. Man hat dann

$$s_i''(x) = \frac{z_i}{h_i}(x_{i+1} - x) + \frac{z_{i+1}}{h_i}(x - x_i) \qquad \text{für } x \in [x_i, x_{i+1}] \text{ und } h_i := x_{i+1} - x_i.$$

Zweifache Integration ergibt

$$s_i(x) = \frac{z_i}{6h_i}(x_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - x_i)^3 + C(x - x_i) + D(x_{i+1} - x).$$

Aus den Bedingungen $s_i(x_i) = y_i$ und $s_i(x_{i+1}) = y_{i+1}$ folgt

$$\begin{aligned} s_i(x) = \frac{z_i}{6h_i}(x_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - x_i)^3 + \left(\frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6}\right)(x - x_i) \\ + \left(\frac{y_i}{h_i} - \frac{z_i h_i}{6}\right)(x_{i+1} - x). \end{aligned} \tag{3.1}$$

Zur Bestimung der z_i benutzen wir die Gleichungen ($i = 2, \ldots n-1$)

$$s'_{i-1}(x_i) = s'_i(x_i).$$

Es folgt aus (3.1)

$$s'_i(x_i) = -\frac{h_i}{3}z_i - \frac{h_i}{6}z_{i+1} - \frac{y_i}{h_i} + \frac{y_{i+1}}{h_i}$$

und

$$s'_{i-1}(x_i) = \frac{h_{i-1}}{6}z_{i-1} + \frac{h_{i-1}}{3}z_i - \frac{y_{i-1}}{h_{i-1}} + \frac{y_i}{h_{i-1}}.$$

Man erhält das Gleichungssystem ($i = 2, \ldots, n-1$)

$$h_{i-1}z_{i-1} + 2(h_i + h_{i-1})z_i + h_i z_{i+1} = \frac{6}{h_i}(y_{i+1} - y_i) - \frac{6}{h_{i-1}}(y_i - y_{i-1}).$$

Dies sind $n-2$ Gleichungen für n Unbekannte. Man sieht, daß man z_1 und z_n noch frei wählen kann. Eine besonders wichtige Wahl ist $z_1 = z_n = 0$. Ein kubischer Spline mit $s''(x_1) = s''(x_n) = 0$ heißt natürlich. Mit den Bezeichnungen

$$h_i = x_{i+1} - x_i, \qquad u_i = 2(h_i + h_{i-1}), \qquad b_i = \frac{6}{h_i}(y_{i+1} - y_i), \qquad v_i = b_i - b_{i-1}$$

kann man das System so schreiben

$$\begin{pmatrix} u_2 & h_2 & & & & \\ h_2 & u_3 & h_3 & & & \\ & h_3 & u_4 & h_4 & & \\ & & h_4 & \ddots & \ddots & \\ & & & \ddots & \ddots & h_{n-2} \\ & & & & h_{n-2} & u_{n-1} \end{pmatrix} \begin{pmatrix} z_2 \\ z_3 \\ z_4 \\ \vdots \\ \vdots \\ z_{n-1} \end{pmatrix} = \begin{pmatrix} v_2 \\ v_3 \\ v_4 \\ \vdots \\ \vdots \\ v_{n-1} \end{pmatrix}.$$

Dabei ist zu beachten, daß die Matrix symmetrisch, *tridiagonal* und *diagonaldominant* ist. Eine Matrix heißt diagonaldominant, falls

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \qquad \text{für alle } i.$$

Ein Gleichungssystem mit einer solchen Matrix hat stets genau eine Lösung, die in einer zur Matrixgröße $n-2$ proportionalen Zeit berechnet werden kann.

Satz 22 (Natürliche kubische Interpolationssplines)**.** *Es sei $f \in C^2([a,b])$ und sei $a = x_1 < x_2 < \cdots < x_n = b$. Dann existiert genau ein natürlicher kubischer Spline mit den Knoten $x_1, \ldots, x_n$, der die Daten von f interpoliert. Zusätzlich gilt*

$$\int_a^b s''(x)^2 \, dx \leq \int_a^b f''(x)^2 \, dx.$$

Das heißt, s ist die interpolierende C^2-Funktion mit minimaler L_2-Norm der zweiten Ableitung.

Satz 23 (Natürliche Splines höherer Ordnung)**.** *Ein natürlicher Spline vom Grad $2m-1$ zu den Knoten $a = x_1 < x_2 < \cdots < x_n = b$ ist eine Funktion $s \in C^{2m-2}(\mathbb{R})$, die in jedem Intervall $[x_i, x_{i+1}]$ in P_{2m} liegt und in den Intervallen $]-\infty, x_1]$ und $[x_n, \infty[$ in P_m. Sei $m \in \mathbb{N}$ und $n \geq m$. Dann gibt es genau einen natürlichen Spline s mit den Knoten $x_1, \ldots, x_n$ der eine Funktion $f \in C^m(\mathbb{R})$ interpoliert und es gilt*

$$\int_a^b s^{(m)}(x)^2 \, dx \leq \int_a^b f^{(m)}(x)^2 \, dx.$$

3.3 Optimale Rekonstruktion

Problemstellung. Gegeben sei eine lineare Abbildung $S : F_1 \to G$, wobei F_1, G Räume mit Skalarprodukt, z.B. Hilberträume, sind. Wegen der allgemeineren Anwendbarkeit der Ergebnisse wollen wir allerdings nicht verlangen, daß das Skalarprodukt definit ist. Aus $(f,f) = 0$ folgt also nicht unbedingt, daß $f = 0$. Das betrachtete Skalarprodukt ist also lediglich eine symmetrische und positiv semidefinite Bilinearform. Wir schreiben $(f,g)_{F_1}$ (oder ähnlich) für das Skalarprodukt auf F_1.

Wir untersuchen folgendes Problem. Wie kann man $S(f)$ möglichst genau schätzen, wenn lediglich

$$N(f) = (f(x_1), \ldots, f(x_n)) \in \mathbb{R}^n$$

bekannt ist? Gesucht ist also ein $\widetilde{S}$ von der Form $\widetilde{S}(f) = \varphi(N(f))$ mit kleinstem Fehler

$$\Delta_{\max}(\widetilde{S}) = \sup_{f \in F} \|\widetilde{S}(f) - S(f)\|.$$

Dabei muß eine Menge $F \subseteq F_1$ zulässiger f fixiert werden, auf der der Fehler betrachtet wird. Oft wählt man

$$F = \{f \in F_1 \mid \|f\| \leq 1\}.$$

Die (Halb-) Norm ergibt sich hierbei aus dem Skalarprodukt durch $\|h\| = \sqrt{(h,h)}$.

Wichtige Beispiele.

– Interpolationsprobleme. Hier ist $F_1 \subseteq G$ und $S = id : F_1 \to G$. Zum Beispiel $F_1 = C^k([0,1])$, wobei $k \in \mathbb{N}$, und $G = C([0,1])$. Dann verwendet man die Skalarprodukte

$$(f,g)_k = \int_0^1 f^{(k)}(x) \cdot g^{(k)}(x) \, dx$$

auf F_1 und

$$(f,g)_0 = \int_0^1 f(x) \cdot g(x)\ dx$$

auf G. Man überzeugt sich leicht davon, daß $(f,g)_k$ nur für $k = 0$ positiv definit auf $C^k([0,1])$ ist.

Im Fall $k = 1$ definieren wir F_1 allerdings etwas anders als eben angegeben. Damit später alles richtig ist, müßen die stückweise linearen Funktionen in F_1 liegen. Daher definieren wir im Fall $k = 1$ die Menge F_1 als die Menge der stetigen und stückweise differenzierbaren Funktionen[1] auf $[0,1]$.

– Integrationsprobleme. F_1 ist ein Raum von Funktionen, z.B. $F_1 = C^k([0,1])$ mit dem Skalarprodukt $(f,g)_k$ und $G = \mathbb{R}$ (mit dem gewöhnlichen Produkt in $\mathbb{R}$ als Skalarprodukt). Die Abbildung S ist dann gegeben durch

$$S(f) = \int_0^1 f(x)\ dx.$$

Spline-Algorithmus. Im Folgenden nehmen wir an, daß $F = \{f \in F_1 | \|f\| \leq K\}$ mit $K > 0$. Ein $\sigma(y) \in F_1$ heißt *abstrakter (Interpolations-) Spline* zur Information $N(f) = y \in \mathbb{R}^n$, falls $N(\sigma(y)) = y$ und

$$\text{Für alle } \widetilde{f} \in F_1 \text{ mit } N(\widetilde{f}) = y \text{ gilt} \quad \|\widetilde{f}\| \geq \|\sigma(y)\|.$$

Allgemein gilt: ist die Abbildung $y \mapsto \sigma(y)$ wohldefiniert, so ist sie linear, d.h. es gilt

$$\sigma(y) = \sum_{i=1}^{n} y_i \cdot f_i$$

mit geeigneten $f_i \in F_1$. So ein abstrakter Spline existiert allerdings nicht immer. Beim Beispiel $F_1 = C^k([0,1])$ muß man etwa $k > 0$ voraussetzen, damit für jedes $y \in \mathbb{R}^n$ ein Spline $\sigma(y)$ existiert. Auch die Eindeutigkeit von $\sigma(y)$ ist i.a. nicht gegeben, wir erreichen sie durch die Forderung

$$(N(f) = 0 \ \wedge\ \|f\| = 0) \ \Rightarrow\ f = 0 \tag{3.2}$$

an die Informationsabbildung $N : F_1 \to \mathbb{R}^n$. Beim Beispiel $F_1 = C^k([0,1])$ gilt (3.2) genau dann, wenn $n \geq k$ (wobei wir voraussetzen, daß die Knoten verschieden sind).

Man kann zeigen: Gilt (3.2) und ist $F_1 \cap \{f \mid N(f) = 0\}$ vollständig bezüglich der Metrik $d(f_1, f_2) = \sqrt{(f_1 - f_2, f_1 - f_2)}$, so existiert zu jedem $y \in \mathbb{R}^n$ genau ein Spline $\sigma(y)$, siehe [31]. In Satz 24 beweisen wir ein ähnliches (aber schwächeres) Ergebnis.

Leider lassen sich diese Aussagen nicht auf den Fall $F_1 = C^k([0,1])$ anwenden, da hier die Vollständigkeit nicht gegeben ist. Als Konsequenz von Satz 23 erhalten wir aber auch in diesem Fall die Existenz der Splines.

[1] Das heißt genauer: $f : [0,1] \to \mathbb{R}$ ist stetig und es gibt endlich viele $0 = t_1 < t_2 < \cdots < t_k = 1$ so daß f eingeschränkt auf jedes Teilintervall $[t_i, t_{i+1}]$ stetig differenzierbar ist.

Folgerung. Sei $F_1 = C^k([0,1])$ und $n \geq k > 0$ mit paarweise verschiedenen Knoten x_i,

$$N(f) = (f(x_1), \ldots, f(x_n)).$$

Dann existiert zu jedem $y \in \mathbb{R}^n$ ein eindeutig bestimmter abstrakter Interpolationsspline. Er stimmt überein mit dem natürlichen Spline $\sigma(y)$ vom Grad $2k-1$, der die Werte $y_i = f(x_i)$ interpoliert.

Satz 24 (Existenz des abstrakten Interpolationssplines)**.** *Es sei F_1 ein Hilbertraum, d.h. F_1 ist vollständiger metrischer Raum bezüglich der Metrik*

$$d(f_1, f_2) = \sqrt{(f_1 - f_2, f_1 - f_2)}.$$

Weiter sei $N : F_1 \to \mathbb{R}^n$ linear und stetig. Dann existiert zu jedem $y \in N(F_1) \subset \mathbb{R}^n$ genau ein abstrakter Interpolationsspline $\sigma(y) \in F_1$.

Satz 25 (Optimalität des Spline-Algorithmus)**.** *Seien F, F_1, G, S, N wie oben, zu jedem $y \in \mathbb{R}^n$ existiere genau ein $\sigma(y)$. Sei*

$$S^*(f) = S(\sigma(N(f))) = \sum_{i=1}^{n} f(x_i) \cdot S(f_i).$$

Dann ist S^* von der Form $S^* = \varphi \circ N$ und hat unter allen diesen Abbildungen den kleinsten maximalen Fehler.

Bemerkungen. Insbesondere existiert ein optimales Verfahren, das linear ist und nicht von K abhängt. Dieses Verfahren heißt Spline-Algorithmus. Es besteht darin, daß S angewendet wird auf den abstrakten Spline σ, der die Daten interpoliert. Im Fall $F_1 = C^2([0,1])$ würde man also zunächst den kubischen natürlichen Interpolationsspline $\sigma(y)$ berechnen und dann $S^*(f) = S(\sigma(y))$ als Näherung für $S(f)$ benutzen. Dies ist das beste, was man machen kann, sofern lediglich die Funktionswerte $N(f) = y \in \mathbb{R}^n$ benutzt werden sollen. Setzt man speziell $S(f) = f$ oder $S(f) = f'$ oder $S(f) = \int_0^1 f(x)dx$, so erhält man in einheitlicher Weise optimale Methoden zur Rekonstruktion von f bzw. zur numerischen Differentiation oder numerischen Integration.

Die Spline-Methoden haben den Vorteil, daß man gute Fehlerabschätzungen beweisen kann, siehe Satz 21, Satz 23 und Satz 25. Damit der Fehler klein wird, genügt es meist, daß die Feinheit der Zerlegung h klein ist – man hat also bei der Wahl der Knoten einen großen Spielraum. Dagegen ist der Fehler bei der Interpolation durch Polynome nur für spezielle Knoten klein, etwa bei den Tschebyscheff-Knoten. Gerade für äquidistante Knoten ist dagegen der Fehler häufig sehr groß.

Spline-Funktionen haben noch einen weiteren Vorteil. Es gibt eine Basis aus Splines mit kleinem Träger, die sogenannten B-Splines. Dies führt dazu, daß die zu lösenden Gleichungssysteme dünn besetzt und gut konditioniert sind.

3.4 Probleme mit unscharfen Daten

Wir haben die Rekonstruktionsprobleme in 3.3 unter der Voraussetzung gelöst, daß die Information $N(f) = y \in \mathbb{R}^n$ exakt bekannt ist. Unter den Voraussetzungen von Satz 25 ist dann der Spline-Algorithmus $f \mapsto S^*(f) = S(\sigma(N(f)))$ optimal. In den Anwendungen ist allerdings $N(f) = y$ meist nicht exakt bekannt. Durch Meßfehler und/oder Rundungsfehler treten kleinere oder größere Datenfehler bei y auf. Wir nehmen jetzt an, daß statt $N(f)$ nur

$$\widetilde{N}(f) = (f(x_1) + \varepsilon_1, \ldots, f(x_n) + \varepsilon_n) = \widetilde{y}$$

bekannt ist. Im unverrauschten Fall war die optimale Methode von der Form

$$S^*(f) = S(\sigma(N(f))) = \sum_{i=1}^{n} f(x_i) \cdot g_i$$

mit gewissen $g_i \in G$. Entsprechend der Kondition von S^* wirken sich Datenfehler auf das Ergebnis aus. In manchen Fällen ist die Abweichung von $\widetilde{S}^*(f) = \sum_{i=1}^n \widetilde{y}_i g_i$ und $S^*(f) = \sum_{i=1}^n y_i g_i$ so groß, daß $\widetilde{S}^*(f)$ keine sinnvolle Lösung des Problems darstellt.

Insbesondere wenn man relativ viele ungenaue Daten $f(x_i) + \varepsilon_i = \widetilde{y}_i$ hat, erscheint es günstiger, bei der Konstruktion des Splines σ auf die genaue Interpolation der Daten zu verzichten und die Meßfehler geeignet „auszugleichen“. Dies ist insbesondere dann sinnvoll, wenn von der wahren Lösung $g = S(f)$ zusätzliche Eigenschaften (etwa Glattheitseigenschaften) bekannt sind. Wir nehmen der Einfachheit halber an, daß $S : F_1 \to G$ bijektiv ist. Für die Inverse schreiben wir auch $K = S^{-1}$. Wir wollen also $S(f) = g$ berechnen bzw. die Gleichung

$$K(g) = f$$

lösen. Statt der (genauen) rechten Seite f kennt man nur die *verrauschte Information* $\widetilde{y} = \widetilde{N}(f)$. Ohne weitere Zusatzinformation wird man $S(\sigma(\widetilde{y}))$ als Näherung für g betrachten.

Jetzt betrachten wir den Fall, daß

$$\|B(g) - z\| \leq E \tag{3.3}$$

bekannt sei. Hierbei sei $B : G \to Z$ eine lineare Abbildung in einen weiteren Raum mit Skalarprodukt.

Beispiele. Häufig ist die gesuchte Größe g eine physikalische Größe (Kraft, Geschwindigkeit, Beschleunigung). Dann ist es eventuell aus dem Zusammenhang heraus klar, daß eine Abschätzung $\|g\| \leq E$ oder $\|g'\| \leq E$ gelten muß. Eine Zusatzinformation der Form (3.3) kann jedenfalls nicht aus den Daten erschlossen werden, sie muß anderweitig bekannt sein.

Weiter sei bekannt, daß die Funktion f durch die verrauschten Daten $\widetilde{y} = \widetilde{N}(f)$ zumindest ungefähr gegeben ist. Wir nehmen an, daß $\|f - \sigma(\widetilde{y})\| \leq \varepsilon$, das heißt für g gilt

$$\|K(g) - \sigma(\widetilde{y})\| \leq \varepsilon. \tag{3.4}$$

Das eigentlich gesuchte g soll $K(g) = f$ oder $S(f) = g$ erfüllen. Da aber f nur näherungsweise bekannt ist – dafür aber noch die Information (3.3) über g gegeben ist – sind wir mit jedem $g \in G$ zufrieden, das (3.3) und (3.4) erfüllt, eventuell sogar in einer abgeschwächten Form. Dies ist die Idee der sogenannten *Tikhonov-Regularisierung*. Dabei nehmen wir an, daß die Norm bei (3.4) wiederum von einem Skalarprodukt herkommt. Meist wählt man eine L_2-Norm bzw. im endlichdimensionalen Fall die euklidische Norm.

Satz 26 (Eigenschaften der Tikhonov-Regularisierung)**.** *Es sei $S : F_1 \to G$ bijektiv, linear und stetig, wobei sowohl F_1 als auch G Hilberträume sind, also vollständig. Weiter sei $N : F_1 \to \mathbb{R}^n$ linear und stetig.*[2]

Es sei $\widetilde{g}$ die eindeutig bestimmte Minimalstelle des Funktionals

$$V_\alpha(g) = \|K(g) - \sigma(\widetilde{y})\|^2 + \alpha \cdot \|B(g) - z\|^2$$

mit $\alpha = \frac{\varepsilon^2}{E^2}$. Dann gilt

$$\|B(\widetilde{g}) - z\| \leq \sqrt{2} \cdot E$$

und

$$\|K(\widetilde{g}) - \sigma(\widetilde{y})\| \leq \sqrt{2} \cdot \varepsilon.$$

Außerdem ist $\widetilde{g}$ die (eindeutig bestimmte) Lösung der Gleichung

$$(K^tK + \alpha \cdot B^tB)(\widetilde{g}) = K^t(\sigma(\widetilde{y})) + \alpha \cdot B^t(z). \tag{3.5}$$

Zum Beweis: Es gilt $K : G \to F_1$ und $K^t : F_1 \to G$ mit

$$(Kg, f) = (g, K^t f)$$

für alle $f \in F_1$ und $g \in G$. Analog ist die zu B adjungierte Abbildung B^t definiert. Im Fall $F_1 = G = \mathbb{R}^m$ entspricht der adjungierten Abbildung die transponierte Matrix. Um Ableitungen im Fall von unendlichdimensionalen Räumen zu vermeiden, beschränken wir uns auf den endlichdimensionalen Fall. Das Funktional V_α ist strikt konvex, ein lokales Minimum ist zugleich globales Minimum, und das globale Minimum $\widetilde{g}$ ist bestimmt durch $\frac{\partial V_\alpha}{\partial x_i}(\widetilde{g}) = 0$. Es gilt

$$V_\alpha(g) = (K(g) - \sigma(\widetilde{y})\,,\, K(g) - \sigma(\widetilde{y})) + \alpha \cdot (B(g) - z, B(g) - z).$$

Das Skalarprodukt läßt sich schreiben als

$$(B(g) - z, B(g) - z) = g^tB^tBg - 2g^tB^tz + z^tz$$

und daher gilt

$$\operatorname{grad}(B(g) - z, B(g) - z) = 2B^tBg - 2B^tz.$$

[2]Diese Voraussetzungen kann man noch abschwächen. Wichtig ist, daß G vollständig ist und daß $(K^tK + \alpha B^tB) : G \to G$ stetig und bijektiv ist. In der Vorlesung wird der Fall bewiesen, wo $F_1 = G = \mathbb{R}^m$ und S bijektiv.

Deshalb ist $\operatorname{grad} V_\alpha(\widetilde{g}) = 0$ äquivalent zu (3.5).

Bemerkungen. a) Die Gleichung (3.5) heißt *Normalgleichung* zum Optimierungsproblem $V_\alpha(g) = \min!$. Die Zahl

$$\alpha := \frac{\varepsilon^2}{E^2}$$

heißt *Regularisierungsparameter*, das Funktional V_α nennt man auch *Regularisierungsfunktional*.

b) Grundlage für den Abschnitt über die Tikhonov-Regularisierung war die Arbeit [14]. Sehr viel mehr enthält die Monographie [23].

Beispiele. a) Es sei $F_1 = C^2([a,b])$ mit dem Skalarprodukt

$$(f,g)_2 = \int_a^b f''(x)g''(x)\,dx.$$

Im Fall der Rekonstruktion von $f \in F_1$ bei unscharfen Daten $\widetilde{N}(f) = \widetilde{y}$ kann man die Identität $S = id : C^2([a,b]) \to C^0([a,b])$ betrachten und erhält das Funktional

$$V_\alpha(f) = \|f - \sigma(\widetilde{y})\|_{L_2}^2 + \alpha \cdot \|f''\|_{L_2}^2.$$

Hierbei ist $\sigma(\widetilde{y})$ der kubische natürliche Interpolationsspline zu den unscharfen Daten $\widetilde{y}$. Es gibt genau ein $\widetilde{f} \in C^2([a,b])$, das dieses Funktional minimiert. Die Funktion $\widetilde{f}$ ist wieder ein kubischer Spline, ein sogenannter Ausgleichsspline.

b) Gesucht sei die Lösung $g \in \mathbb{R}^n$ eines linearen Gleichungssystems $Kg = f$. Statt $f \in \mathbb{R}^n$ kennt man nur $\widetilde{f}$, es gelte

$$\|f - \widetilde{f}\|_2 \le \varepsilon.$$

Weiter sei für die (wahre) Lösung g bekannt, daß

$$\|g\|_2 \le E.$$

Dann betrachtet man das Funktional

$$V_\alpha(g) = \|Kg - \widetilde{f}\|_2^2 + \alpha \cdot \|g\|_2^2$$

mit $\alpha = \frac{\varepsilon^2}{E^2}$ bzw. das Gleichungssystem

$$(K^tK + \alpha \cdot I)(g) = K^t\widetilde{f},$$

wobei I die Einheitsmatrix sei. Die Lösung $\widetilde{g}$ erfüllt dann zumindest $\|\widetilde{g}\|_2 \le \sqrt{2} \cdot E$ und $\|K\widetilde{g} - \widetilde{f}\|_2 \le \sqrt{2} \cdot \varepsilon$. Dieses Beispiel wird in Kapitel 5 noch einmal studiert.

Die Methode der kleinsten Quadrate. Schätzprobleme der Ausgleichsrechnung (d.h. $S : F_1 \to G$ soll genähert werden, bekannt ist aber nur $\widetilde{N}(f) = (f(x_i) + \varepsilon_i)_{i=1,\dots,n}$) lassen sich auch unter gewissen statistischen Annahmen über die ε_i lösen. Die ε_i seien

etwa unabhängige normalverteilte Zufallsvariablen. Die optimale Lösung von Schätzproblemen für lineare Probleme im Hilbertraum läßt sich oft durch „kleinste Quadrate“ charakterisieren. Wir beschreiben die Methode der kleinsten Quadrate im engeren Sinn und behandeln den Fall, daß F_1 endlichdimensional ist.

Da F_1 endlichdimensional ist, kann man annehmen, daß $f \in F$ die Form $f(x) = \sum_{k=1}^{m} a_k f_k(x)$ hat. Dabei sind die a_k unbekannt. Bekannt sei

$$\widetilde{N}(f) =: \widetilde{y} = (\widetilde{y}_i)_{1\le i\le n} = (f(x_i) + \varepsilon_i)_{1\le i\le n} = \left(\sum_{k=1}^{m} a_k f_k(x_i) + \varepsilon_i\right)_{1\le i\le n}.$$

Dabei sind die ε_i unbekannte (zufällige) Fehler. Die Methode der kleinsten Quadrate besagt: Schätze f (bzw. die a_k) so, daß der Wert

$$Q(f) = \sum_{i=1}^{n} (f(x_i) - \widetilde{y}_i)^2$$

minimal wird. Eine andere Schreibweise ist

$$Q(a_1, \ldots, a_m) = \sum_{i=1}^{n} \left(\sum_{k=1}^{m} a_k f_k(x_i) - \widetilde{y}_i\right)^2.$$

Eine Lösung findet man mit dem Ansatz $\frac{\partial Q}{\partial a_i} = 0$. Dies führt auf die Normalgleichung für dieses Problem.

Die Methode ist auch bekannt unter dem Stichwort „Lösung von überbestimmten linearen Gleichungssystemen“. Besonders bekannt ist der Fall der Regressionsgeraden, bei dem F_1 nur aus den linearen Funktionen $f(x) = ax + b$ besteht. Gesucht ist hier die Gerade, die ungenaue Daten am besten interpoliert.

Die numerische Realisierung der Methode der kleinsten Quadrate behandeln wir in Kapitel 5. Dort behandeln wir auch die Lösung von linearen Gleichungssystemen $Ax = b$ mit Hilfe der Tikhonov-Regularisierung noch einmal. Regularisierungsmethoden sind empfehlenswert, wenn die Daten (etwa die rechte Seite b) nicht genau bekannt sind. Bei sehr schlecht konditionierten Matrizen empfiehlt sich die Anwendung der Tikhonov-Regularisierung bereits bei sehr kleinen Fehlern, die sich beim Rechnen mit Gleitkommazahlen nicht vermeiden lassen.

Zusätze und Bemerkungen.

- Wir haben stets angenommen, daß die Funktionen auf einem Intervall definiert sind, $f : [a, b] \to \mathbb{R}$. Oft hat man Funktionen von mehreren Variablen, etwa $f : [0, 1]^d \to \mathbb{R}$, aber natürlich sind auch geometrisch viel kompliziertere Definitionsbereiche wichtig. Das Problem der Interpolation (und allgemeiner: der Rekonstruktion) stellt sich genauso für $d > 1$, aber die Ergebnisse sind viel komplizierter und es gibt noch viele offene Fragen.

  Man kann versuchen, mit einem Analogon zu Satz 8 zu beginnen: Der Raum Π_k^d aller Polynome vom Grad höchstens k hat Dimension

  $$n := \dim(\Pi_k^d) = \binom{d+k}{k}.$$

  Für welche Punkte $\{x_1, \ldots, x_n\} \subset [0, 1]^d$ ist dann das Interpolationsproblem (also: Gesucht ist $p \in \Pi_k^d$ mit vorgegebenen Funktionswerten an den Punkten x_i) eindeutig lösbar?

  Schon diese Frage ist recht schwer zu beantworten, einige Ergebnisse werden in [11] beschrieben.

Aufgaben

3.1. Die Funktion $f(x) = \log(x)$ werde quadratisch interpoliert, Stützstellen seien 10, 11 und 12. Schätzen Sie für $x = 11,1$ den Interpolationsfehler ab und vergleichen Sie mit dem tatsächlichen Fehler. Wie hängt das Vorzeichen des Fehlers von x ab?

3.2. Sei $f : [a, b] \to \mathbb{R}$ gegeben durch $f(x) = \sin x$. Es sei $x_1, x_2, \ldots$ eine Folge in $[a, b]$ aus paarweise verschiedenen Punkten und $p_n \in P_n$ das Polynom mit $p_n(x_i) = f(x_i)$ für $i = 1, 2, \ldots, n$. Zeigen Sie, daß die Folge der p_n gleichmäßig gegen f konvergiert. (Z.B. auch dann, wenn $[a, b] = [0, 1000]$ und $x_i \in [0, 1]$ für alle i.)

3.3. Schreiben Sie ein Programm, das folgendes leistet:

Eingabe: $n \in \mathbb{N}$; $x_1, x_2 \ldots, x_n$ im Intervall $[a, b]$; $y_1, y_2, \ldots, y_n$.

Ausgabe: $p_n \in P_n$ sei das Interpolationspolynom zu den Werten y_i an den Knoten x_i, die als paarweise verschieden vorausgesetzt werden. Berechnen und plotten Sie p_n. Testen Sie Ihr Programm mit folgenden Beispielen:

a) $n = 13$, $x_1 = -1$, $x_2 = -5/6, \ldots, x_{13} = 1$

$$y_i = \frac{1}{1 + 25x_i^2}, \qquad a = -1, \quad b = 1$$

und analog (d.h. wieder äquidistante Knoten mit $x_1 = -1$ und $x_n = 1$) für $n = 8$ und $n = 18$.

b) Alles analog mit den Tschebyscheff-Knoten, d.h. $n = 13$,

$$x_i = \cos \frac{(2i - 1)\pi}{2n}, \qquad y_i = \frac{1}{1 + 25x_i^2},$$

und analog für $n = 8$ und $n = 18$.

c) Diskutieren Sie die Güte der bei a bzw. b erhaltenen Polynome als Approximation der Funktion $f(x) = 1/(1+25x^2)$ im Intervall $[-1, 1]$. Dieses Beispiel wurde von *Runge* (1901) untersucht.

3.4. Für eine Funktion $f : \mathbb{R} \to \mathbb{R}$ und paarweise verschiedene reelle x_i sei definiert

$$f[x_1] = f(x_1)$$

und

$$f[x_1, \ldots, x_k] = \frac{f[x_2, \ldots, x_k] - f[x_1, \ldots, x_{k-1}]}{x_k - x_1}.$$

Zeigen Sie mit vollständiger Induktion:

a) $f[x_1, \ldots, x_n] = 0$ für jedes Polynom $f \in P_{n-1}$;

b) $f[x_1, \ldots, x_n] = 1$ für $f(x) = x^{n-1}$;

c) $f[x_1, \ldots, x_n] = \sum_{i=1}^{n} x_i$ für $f(x) = x^n$;

d) Für $f = g \cdot h$ gilt

$$f[x_1, \ldots, x_n] = \sum_{i=1}^{n} g[x_1, \ldots, x_i] \cdot h[x_i, \ldots, x_n].$$

3.5. Sei $f : [0,1] \to \mathbb{R}$ stetig differenzierbar. Weiter sei $\xi \in [0,1]$ fest gewählt. Diskutieren Sie die Lösbarkeit (Existenz & Eindeutigkeit) des folgenden Interpolationsproblems: Gesucht sei ein Polynom $p \in P_3$ mit $p(0) = f(0)$, $p(1) = f(1)$ und $p'(\xi) = f'(\xi)$.

3.6. Schreiben Sie ein Programm, das folgendes leistet:

Eingabe: $n \in \mathbb{N}$ mit $n \geq 2$; $a = x_1 < x_2 < \cdots < x_n = b$; $y_1, y_2, \ldots, y_n$.

Ausgabe: s sei der natürliche kubische Spline zu den Werten y_i an den Knoten x_i. Es soll $s(x)$ berechnet und geplottet werden.

Testen Sie Ihr Programm mit folgenden Beispielen:

a) $n = 13$, $x_1 = -1$, $x_2 = -5/6, \ldots, x_{13} = 1$

$$y_i = \frac{1}{1 + 25x_i^2},$$

und analog (d.h. wieder äquidistante Knoten mit $x_1 = -1$ und $x_n = 1$) für $n = 8$ und $n = 18$.

b) Alles analog mit den Tschebyscheff-Knoten mit $n = 8$, $n = 13$ und $n = 18$.

c) Diskutieren Sie die Güte der bei a bzw. b erhaltenen Splines als Approximation der Funktion $f(x) = 1/(1 + 25x^2)$ im Intervall $[-1, 1]$.

3.7. Gegeben sei das Gleichungssystem $Ax = y$ durch

$$\begin{pmatrix} 10 & 11 \\ 11 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 21 \\ 23 \end{pmatrix},$$

die Lösung ist $x^t = (1, 1)$. Aufgrund von Meßfehlern sei nur die Näherung $\widetilde{y}^t = (22, 23)$ und

$$\|\widetilde{y} - y\|_2 \leq 2$$

bekannt. Darüber hinaus sei aber für die Lösung x noch bekannt, daß

$$\|x\|_2 \leq 2.$$

a) Lösen Sie das System $A\widetilde{x} = \widetilde{y}$.

b) Welche Näherung ergibt sich, wenn man die Tikhonov-Regularisierung – wie in der Vorlesung besprochen – anwendet?

c) Berechnen Sie die Minimalstelle x_α des Funktionals

$$V_\alpha(x) = \|Ax - \widetilde{y}\|_2^2 + \alpha \cdot \|x\|_2^2$$

in Abhängigkeit von α, wobei $\alpha > 0$. Man kann z.B. ein Programm schreiben, um x_α für $\alpha = 2^k$ mit $k = -10, -9, \ldots, 10$ zu berechnen.

3.8. Gegeben seien Punkte $P_i(f(i), g(i))$ in der Ebene, $i = 1, \ldots, n$. Sei s_f der natürliche kubische Interpolationsspline zu den Daten $(i, f(i))$ und sei s_g der natürliche kubische Interpolationsspline zu den Daten $(i, g(i))$. Berechne s_f und s_g wie in der Aufgabe 3.6 und plotte die Kurve $t \in [0, n] \mapsto (s_f(t), s_g(t))$.

Kapitel 4

Numerische Integration

4.1 Vorbemerkungen

Wir beschäftigen uns mit der numerischen Integration von stetigen Funktionen. Im allgemeinen ist keine Stammfunktion bekannt oder diese läßt sich nur als unendliche Reihe schreiben. Ein Beispiel ist

$$S(x) = \int_0^x \frac{\sin t}{t} dt = \sum_{k=0}^{\infty} \frac{(-1)^k \cdot x^{2k+1}}{(2k+1) \cdot (2k+1)!}.$$

Wie gut solche Reihen für die Berechnung des Integrals geeignet sind, hängt sowohl von der zu integrierenden Funktion als auch vom Integrationsintervall ab. Obwohl Reihenentwicklungen oft nützlich sind, werden sie hier nicht weiter untersucht.

Stattdessen betrachten wir numerische Integrationsverfahren (sog. Quadraturformeln) der Form

$$S_n(f) = \sum_{i=1}^{n} a_i f(x_i) \approx \int_a^b f(x)dx = S(f). \tag{4.1}$$

Formeln dieser Gestalt lassen sich bereits anwenden, wenn man nur wenige Funktionswerte kennt. Dabei wird hier unterstellt, daß die Funktionswerte $f(x_i)$ (durch ein Computerprogramm oder sonstwie) erhalten werden können. Allgemein sind hier zwei Fragen zu betrachten:

– Wie wählt man die Gewichte a_i bei vorgegebenen Knoten x_i?
– Wie wählt man geeignete Knoten x_i?

Zur Wahl der Gewichte a_i bieten sich zwei Möglichkeiten an.

Integration mit polynomialer Interpolation. Eine Möglichkeit, Quadraturformeln der Gestalt (4.1) zu finden, ist die Interpolation gegebener Funktionswerte durch Polynome. Diese können leicht integriert werden und man erhält Formeln der gewünschten Art. Es gibt genau ein $p \in P_n$ mit $p(x_i) = f(x_i)$ für $i = 1, \ldots, n$. Man wähle nun die Gewichte so, daß $S_n(f) = S(p)$. Quadraturformeln S_n, die für alle $p \in P_n$ exakt sind, heißen interpolatorische Quadraturformeln. Man hofft, daß der Fehler $|S(f) - S_n(f)|$

klein wird – entweder, weil p eine gute Approximation für f ist, oder weil sich die positiven und negativen Fehler weitgehend ausgleichen. Der Fehler kann also auch dann klein sein, wenn f durch p nicht gut approximiert wird. Natürlich sind Fehlerabschätzungen nötig, um diese Hoffnung in manchen Fällen (abhängig von f und den Knoten) bestätigen zu können. Dieser Ansatz zur Wahl der a_i wird in Abschnitt 4.2 untersucht.

Verwendung von Interpolationssplines. Das folgende wurde bereits in Kapitel 3 bewiesen. Ist $F_1 = C^k([0,1])$ versehen mit dem Skalarprodukt

$$(f,g)_k = \int_0^1 f^{(k)}(x)g^{(k)}(x)dx$$

und F eine Kugel in F_1 mit Mittelpunkt 0, dann ist die Formel $S_n^*(f) = S(\sigma(f))$ mit dem natürlichen Interpolationsspline $\sigma(f)$ optimal bezüglich des maximalen Fehlers auf F. Diese Aussage liefert bei gegebenen Knoten $x_1, \ldots, x_n$ für verschiedene Werte von k verschiedene optimale Gewichte a_i und damit verschiedene S_n^*.

4.2 Interpolatorische Quadraturformeln

Sei

$$l_i(x) = \prod_{j \neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

für $i = 1, \ldots, n$. Der Index j läuft jeweils von 1 bis n, bis auf $j = i$. Dann ist das Interpolationspolynom $p \in P_n$ gegeben durch $p(x) = \sum_{i=1}^{n} f(x_i)l_i(x)$ und damit erhalten wir

$$S_n(f) = S(p) = \int_a^b \sum_{i=1}^{n} f(x_i)l_i(x)dx = \sum_{i=1}^{n} \int_a^b l_i(x)dx \; f(x_i) = \sum_{i=1}^{n} a_i f(x_i).$$

Hier ist

$$a_i = \int_a^b l_i(x)dx = \int_a^b \prod_{j \neq i}^{n} \frac{x - x_j}{x_i - x_j} dx.$$

Bei Wahl dieser Gewichte erhält man also $S_n(p) = S(p)$ für alle $p \in P_n$. Durch diese Forderung sind die Gewichte schon eindeutig festgelegt, denn für Quadraturformeln S_n der betrachteten Art gilt

$$\int_a^b l_i(x)dx = S(l_i) = S_n(l_i) = a_i$$

wegen $l_i(x_j) = \delta_{ij}$. Wir kennen schon eine Fehlerabschätzung für das Interpolationsproblem, nämlich

$$f(x) - p(x) = \frac{1}{n!} f^{(n)}(\xi_x) \prod_{i=1}^{n} (x - x_i).$$

Hieraus folgt der folgende Satz.

Satz 27 (Fehler von interpolatorischen Quadraturformeln)**.** *Sei $f \in C^n([a,b])$ und sei $S_n(f) = \sum_{i=1}^n a_i f(x_i)$ die interpolatorische Quadraturformel zu $x_1, \ldots, x_n$. Dann gilt*

$$|S(f) - S_n(f)| \leq \frac{1}{n!} \|f^{(n)}\|_\infty \int_a^b \left| \prod_{i=1}^n (x - x_i) \right| dx.$$

Bemerkungen. Allgemein ist man an Fehlerabschätzungen der Form

$$|S(f) - S_n(f)| \leq c \|f^{(k)}\|_\infty \tag{4.2}$$

interessiert. Diese Fehlerabschätzung impliziert, daß Polynome $p \in P_k$ exakt integriert werden. Bisher wurde nur der Fall $n = k$ betrachtet. An Beispielen sieht man, daß die Konstante

$$c = \frac{1}{n!} \int_a^b \left| \prod_{i=1}^n (x - x_i) \right| dx,$$

die sich im Satz 27 (für $k = n$) ergab, im allgemeinen nicht optimal ist, d.h. die Aussage von Satz 27 gilt i.a. noch für kleinere Konstanten. Statt (4.2) könnte man auch schreiben

$$\Delta_{\max}(S_n) = \sup_{f \in F} |S(f) - S_n(f)| \leq c,$$

wobei

$$F = F^k = \{f \in C^k([a,b]) \mid \|f^{(k)}\|_\infty \leq 1\}.$$

Das heißt, der maximale Fehler auf F^k stimmt überein mit der kleinsten Konstanten c, für die (4.2) gilt. Man fragt nun nach optimalen Quadraturformeln für die Klasse F^k, d.h. nach solchen S_n mit kleinstem maximalen Fehler.

Der Satz von Peano. Für den Satz von Peano betrachten wir lineare Funktionale der Form

$$L(f) = \int_a^b f(x)\, dx - \sum_{i=1}^n a_i f(x_i)$$

oder allgemeiner

$$L(f) = \sum_{i=0}^{k-1} \left\{ \int_a^b \alpha_i(x) f^{(i)}(x) dx + \sum_{j=1}^n \beta_{ij} f^{(i)}(z_{ij}) \right\},$$

wobei wir voraussetzen, daß $L(p) = 0$ für $p \in P_k$. Dabei seien $\alpha_i \in C([a,b])$ und $\beta_{ij} \in \mathbb{R}$ und $z_{ij} \in [a,b]$. Die Funktion $K_k : [a,b] \to \mathbb{R}$, definiert durch

$$K_k(t) = \frac{1}{(k-1)!} L\left((x-t)_+^{k-1}\right)$$

mit $(x-t)_+^{k-1} : [a,b] \to \mathbb{R}$, wobei

$$(x-t)_+^{k-1}(x) = \left(\max\{(x-t), 0\}\right)^{k-1},$$

heißt *k-ter Peano-Kern* von L, die Funktion $x \mapsto (x-t)_+^{k-1}$ heißt *abgebrochene Potenz*.

Satz 28 (Satz von Peano 1905/1913)**.** *Sei L ein Funktional von der angegebenen Gestalt und sei $L(p) = 0$ für alle Polynome $p \in P_k$. Dann gilt für jedes $f \in C^k([a,b])$*

$$L(f) = \int_a^b K_k(t) f^{(k)}(t)\, dt$$

Beweis: Nach dem Satz von Taylor gilt für $f \in C^k([a,b])$

$$f(x) = \sum_{j=0}^{k-1} \frac{1}{j!} \cdot f^{(j)}(a)(x-a)^j + r(x)$$

mit

$$r(x) = \frac{1}{(k-1)!} \int_a^x f^{(k)}(t)(x-t)^{k-1}\, dt.$$

Weiterhin ist $L(f) = L(r)$, da $L(p) = 0$ für $p \in P_k$. Man kann auch schreiben

$$r(x) = \frac{1}{(k-1)!} \int_a^b f^{(k)}(t) \cdot (x-t)_+^{k-1}(x)\, dt.$$

Es folgt

$$L(f) = L(r) = \int_a^b K_k(t) f^{(k)}(t)\, dt,$$

da man unter den angegebenen Voraussetzungen das Funktional L mit dem Integral vertauschen darf.

Folgerung. Für unser Problem der Fehlerabschätzung von Quadraturverfahren kann man den Satz von Peano für k und $L(f) = S(f) - S_n(f)$ anwenden, wenn Polynome $p \in P_k$ von S_n exakt integriert werden. Ist speziell S_n eine interpolatorische Quadraturformel, so kann man Satz 28 mindestens für $k = 1, \ldots, n$ anwenden.

Satz 29 (Anwendung des Satzes von Peano)**.** *Es sei $S_n(f) = \sum_{i=1}^n a_i f(x_i)$ und $S(f) = \int_a^b f(x)dx$. Für Polynome $p \in P_k$ gelte $S_n(p) = S(p)$. Dann gilt für*

$$F^k = \left\{ f \in C^k([a,b]) \mid \|f^{(k)}\|_\infty \leq 1 \right\}$$

die Fehlerdarstellung

$$\Delta_{\max}(S_n) = \sup_{f \in F^k} |S(f) - S_n(f)| = \int_a^b |K_k(t)| dt$$

wobei K_k der k-te Peano-Kern ist.

Beweis: Sei $f \in F^k$. Dann gilt die Darstellung

$$S(f) - S_n(f) = \int_a^b K_k(t) f^{(k)}(t)\, dt$$

und damit

$$|S(f) - S_n(f)| \leq \int_a^b |K_k(t)| \cdot |f^{(k)}(t)|\, dt \leq \int_a^b |K_k(t)|\, dt. \tag{4.3}$$

Aus der Definition von K_k folgt, daß K_k stückweise stetig ist und nur endlich viele Vorzeichenwechsel hat, etwa an den Stellen $a < v_1 < \cdots < v_l < b$. Betrachte nun ein $f \in F^k$ mit $\|f^{(k)}\|_\infty = 1$ und $f^{(k)}(x) = \operatorname{sgn}(K_k(x))$, falls $\inf_{1\leq i\leq l} |x - v_i| \geq \varepsilon$. Da man $\varepsilon > 0$ beliebig klein wählen kann, läßt sich die Ungleichung (4.3) nicht verbessern.

Mit Hilfe des letzten Satzes kann man die folgenden Fehleraussagen beweisen, wir verzichten auf die Nebenrechnungen.

Satz 30 (Fehler einiger Quadraturformeln)**.** *Sei $S(f) = \int_a^b f(x)\, dx$. Dann gilt für die Trapezregel $S_2(f) = \frac{1}{2}(b-a)(f(a)+f(b))$*

$$|S(f) - S_2(f)| \leq \frac{1}{12}(b-a)^3 \|f''\|_\infty;$$

für die Mittelpunktregel $S_1(f) = (b-a)f(\frac{a+b}{2})$ gilt

$$|S(f) - S_1(f)| \leq \frac{1}{24}(b-a)^3 \|f''\|_\infty;$$

und für die Simpson-Regel $S_3(f) = \frac{b-a}{6}(f(a) + 4f(\frac{a+b}{2}) + f(b))$ gilt

$$|S(f) - S_3(f)| \leq \frac{1}{2880}(b-a)^5 \|f^{(4)}\|_\infty.$$

Bemerkung: Optimale Quadraturformeln bzw. Konvergenzordnung. Wir betrachten

$$F^k = \{f \in C^k[a,b] \mid \|f^{(k)}\|_\infty \leq 1\}$$

für ein festes k und fragen nach guten Quadraturformeln S_n für F^k, auch für großes n. Wie schnell kann $\Delta_{\max}(S_n)$ für $n \to \infty$ gegen 0 konvergieren, wenn die Folge S_n optimal gewählt wird?

Mit Hilfe von Satz 29 könnte man versuchen, optimale Formeln für F^k und beliebige n zu finden. Die Rechnungen werden für $k \geq 3$ jedoch so kompliziert, daß sie bisher nicht gemacht wurden. Im Folgenden begnügen wir uns mit der Konstruktion einer Folge von Quadraturformeln (zu vorgegebenem F^k), so daß die Konvergenzordnung von $(\Delta_{\max}(S_n))_{n\in\mathbb{N}}$ optimal ist.

4.3 Zusammengesetzte Quadraturformeln

Konstruktion zusammengesetzter Quadraturformeln. Wir beginnen mit einer Näherung $S_k(f) = \sum_{i=1}^k a_i f(x_i)$ von $S(f) = \int_c^d f(x)\, dx$. Die Formel S_k sei exakt für $p \in P_k$. Es gelte die Fehlerabschätzung

$$|S(f) - S_k(f)| \leq \beta(d-c)^{k+1} \|f^{(k)}\|_\infty.$$

Die Abbildung

$$\lambda : [c,d] \to [a,b], \quad \lambda(t) = t\frac{b-a}{d-c} + \frac{da-bc}{d-c},$$

ist affin linear und surjektiv. Bei $\widetilde{S}(f) = \int_a^b f(x)\,dx$ machen wir die Substitution

$$\widetilde{S}(f) = \int_a^b f(x)\,dx = \int_c^d \frac{b-a}{d-c} f(\lambda(t))\,dt = S\left(\frac{b-a}{d-c}(f\circ\lambda)\right).$$

Wir können nun das zweite Integral mit S_k näherungsweise berechnen und erhalten

$$S_k\left(\frac{b-a}{d-c}(f\circ\lambda)\right) = \frac{b-a}{d-c}\sum_{i=1}^{k} a_i \cdot f\left(\frac{b-a}{d-c}x_i + \frac{da-bc}{d-c}\right) =: \widetilde{S}_k(f).$$

Die Formel $\widetilde{S}_k(f)$ entsteht aus S_k durch „Verschieben“, es gilt

$$\widetilde{S}(f) = S\left(\frac{b-a}{d-c}(f\circ\lambda)\right) \qquad \text{und} \qquad \widetilde{S}_k(f) = S_k\left(\frac{b-a}{d-c}(f\circ\lambda)\right).$$

Wenn nun S_k für $p \in P_k$ exakt ist, so auch $\widetilde{S}_k$. Man erhält mit

$$(f\circ\lambda)^{(k)}(x) = \left(\frac{b-a}{d-c}\right)^k \cdot f^{(k)}(\lambda(x))$$

die Fehlerabschätzung

$$\begin{aligned}\left|\widetilde{S}(f) - \widetilde{S}_k(f)\right| &= \left|S\left(\frac{b-a}{d-c}(f\circ\lambda)\right) - S_k\left(\frac{b-a}{d-c}(f\circ\lambda)\right)\right| \\ &\le \frac{b-a}{d-c}\cdot\beta(d-c)^{k+1}\|(f\circ\lambda)^{(k)}\|_\infty \le \beta(b-a)^{k+1}\|f^{(k)}\|_\infty.\end{aligned} \tag{4.4}$$

Nun sei $n = k\cdot m$ mit einem $m \in \mathbb{N}$. Dann kann man aus S_k eine *zusammengesetzte Quadraturformel* $S_{m\cdot k}$ konstruieren. Wir zerlegen dazu $[a,b]$ in m gleichlange Intervalle $I_1,\ldots,I_m$,

$$I_j := \left[a + \frac{j-1}{m}(b-a), a + \frac{j}{m}(b-a)\right] \qquad \text{für } j = 1,\ldots,m.$$

Durch Verschieben der Formel S_k auf die Intervalle $I_1,\ldots,I_m$ erhalten wir Formeln $S_k^1,\ldots,S_k^m$ zur Integration von

$$S^j(f) = \int_{I_j} f(x)\,dx.$$

Wir definieren nun die zusammengesetzte Quadraturformel S_n durch

$$S_n(f) = \sum_{i=1}^{m} S_k^i(f). \tag{4.5}$$

Dann erhält man die Fehlerabschätzung

$$|S_n(f) - S(f)| \leq m\beta \|f^{(k)}\|_\infty \left(\frac{b-a}{m}\right)^{k+1} \tag{4.6}$$

und damit

$$\Delta_{\max}(S_n) \leq \beta(b-a)^{k+1} \cdot m^{-k} = \beta(b-a)^{k+1} \left(\frac{k}{n}\right)^k = \widetilde{\beta} n^{-k}$$

für $F = F^k$.

Satz 31 (Optimale Konvergenzordnung von Quadraturformeln)**.** *Sei $k \in \mathbb{N}$. Dann gibt es Zahlen $c_1, c_2 > 0$ mit*

$$c_1 \cdot n^{-k} \leq \inf_{S_n} \Delta_{\max}(S_n) \leq c_2 \cdot n^{-k}$$

für die Funktionenklasse F^k und alle $n \geq k$. Das heißt, die optimale Konvergenzordnung von Quadraturverfahren auf der Klasse F^k ist n^{-k}.

Beweis: Die Existenz von S_n mit $\Delta_{\max}(S_n) \leq \beta n^{-k}$ wurde für F^k und $n = k \cdot m$ bereits gezeigt. Für ein beliebiges $\widetilde{n} = n + k_0 = k \cdot m + k_0$ mit $k_0 \in \{1, \ldots, k-1\}$ betrachten wir $S_{\widetilde{n}} := S_n$ und erhalten $\Delta_{\max}(S_n) \leq \beta n^{-k} \leq c_2 \cdot \widetilde{n}^{-k}$ für ein geeignetes c_2. Wir zeigen nun die Abschätzung von unten. Hierzu sei

$$S_n(f) = \sum_{i=1}^{n} a_i f(x_i)$$

eine beliebige Quadraturformel. Definiere

$$\varphi(x) = \begin{cases} r(1-x^2)^{k+1} & \text{falls } |x| \leq 1 \\ 0 & \text{sonst} \end{cases}$$

mit $r > 0$ gerade so, daß $\|\varphi^{(k)}\|_\infty = 1$. Es gilt $\varphi \in C^k(\mathbb{R})$ und wir definieren $\delta > 0$ durch

$$\int_{\mathbb{R}} \varphi(x)\, dx = \delta.$$

Weiter definieren wir für $\varepsilon > 0$ und $z \in \mathbb{R}$

$$\varphi_z^\varepsilon(x) = \varepsilon^k \cdot \varphi\left((x-z)\varepsilon^{-1}\right).$$

Dann gilt

$$\|(\varphi_z^\varepsilon)^{(k)}\|_\infty = 1,$$

d.h. $\varphi_z^\varepsilon \in F^k$. Außerdem ist $\varphi_z^\varepsilon(x) = 0$ für $|x - z| \geq \varepsilon$ und

$$\int_{\mathbb{R}} \varphi_z^\varepsilon(x)\, dx = \varepsilon^{k+1}\delta.$$

Wir nehmen an, daß die Knoten $x_1, \ldots, x_n$ von S_n geordnet sind,

$$a = x_0 \le x_1 < x_2 < \cdots < x_n \le x_{n+1} = b.$$

Definiere für $i \in \{1, \ldots, n+1\}$

$$z_i := \frac{x_i + x_{i-1}}{2} \qquad \text{und} \qquad \varepsilon_i := \frac{x_i - x_{i-1}}{2}$$

und betrachte

$$f^* = \sum_{i=1}^{n+1} \varphi_{z_i}^{\varepsilon_i} \in F^k.$$

Man hat nun einerseits

$$S_n(f^*) = \sum_{i=1}^{n} a_i f^*(x_i) = 0$$

und andererseits

$$S(f^*) = \int_a^b f^*(x)\,dx = \delta \sum_{i=1}^{n+1} \varepsilon_i^{k+1}.$$

Also gilt

$$\Delta_{\max}(S_n) \ge |S(f^*) - S_n(f^*)| \ge \delta \sum_{i=1}^{n+1} \varepsilon_i^{k+1}.$$

Die Funktion

$$(\varepsilon_1, \ldots, \varepsilon_{n+1}) \mapsto \sum_{i=1}^{n+1} \varepsilon_i^{k+1}$$

ist konvex und erreicht auf der Menge $\sum \varepsilon_i = (b-a)/2$, wobei $\varepsilon_i \ge 0$ für alle i, ihr Minimum für

$$\varepsilon_1 = \varepsilon_2 = \cdots = \varepsilon_{n+1} = \frac{b-a}{2(n+1)}.$$

Damit hat man also

$$\Delta_{\max}(S_n) \ge (n+1) \cdot \delta \cdot \left(\frac{b-a}{2(n+1)}\right)^{k+1} = \delta \left(\frac{b-a}{2}\right)^{k+1} \left(\frac{1}{n+1}\right)^k \ge c_1 \cdot n^{-k}.$$

Bemerkungen.

1) Beim Beweis der unteren Schranke

$$\Delta_{\max}(S_n) \ge c_1 \cdot n^{-k}$$

haben wir zu gegebenen Knoten $x_1 \le x_2 \cdots \le x_n$ eine Funktion f^* definiert mit $f^* \in F^k$ und

$$N_n(f^*) = (f^*(x_1), \ldots, f^*(x_n)) = 0 \in \mathbb{R}^n.$$

Ebenso folgt natürlich $N_n(-f^*) = 0$ und $-f^* \in F^k$. Daher gilt $S_n(f^*) = S_n(-f^*)$ sogar für jede Abbildung $S_n : F^k \to \mathbb{R}$ der Form $S_n = \varphi \circ N_n$. Hieraus folgt

$$\Delta_{\max}(S_n) \ge S(f^*)$$

für jedes dieser S_n. Damit ist gezeigt, daß die Aussage von Satz 31 richtig bleibt, wenn man beliebige (nichtlineare) Quadraturformeln der Form $S_n = \varphi \circ N_n$ zur Konkurrenz zuläßt.

2) Noch allgemeiner könnte man auch *adaptive Algorithmen* betrachten, bei denen die Wahl der Stützstelle x_{k+1} in Abhängigkeit der schon berechneten Funktionswerte $f(x_1), \ldots, f(x_k)$ erfolgen kann. Wiederum kann man zeigen, daß die Aussage von Satz 31 auch für adaptive Verfahren gilt. Für manche andere Probleme der Numerik sind adaptive Verfahren viel besser als nichtadaptive. Siehe [18] für eine Übersicht.

3) Bakhvalov war einer der ersten, der sich für *optimale numerische Methoden* interessiert hat und Sätze von diesem Typ bewiesen hat. In seiner Arbeit [1] betrachtet er neben deterministischen Algorithmen auch Monte-Carlo-Methoden. Die Ergebnisse von Bakhvalov sind auch in [16] enthalten.

4) Der Begriff der Konvergenzordnung wird hier etwas anders gebraucht als in Kapitel 2. Dies liegt vor allem daran, daß jetzt der Fehler typischerweise wie $n^{-\alpha}$ gegen 0 geht (und man nennt dann α die „Konvergenzordnung“) während in Kapitel 2 selbst das einfache Bisektionsverfahren viel schneller konvergiert.

Beispiele für zusammengesetzte Quadraturformeln. Bei zusammengesetzten Quadraturformeln (4.5) braucht man höchstens $n = k \cdot m$ Funktionswerte. Diese Zahl verringert sich auf $n = k \cdot m - m + 1$, falls die Funktionswerte an beiden Intervallenden benutzt werden. Die folgenden Fehlerabschätzungen sind Spezialfälle der Formel (4.6).

Mittelpunktverfahren. Für das zusammengesetzte Mittelpunktverfahren ($n = m$) gilt die Fehlerabschätzung

$$\left| \int_a^b f(x)\,dx - \sum_{i=1}^{n} \frac{b-a}{n} f\left(a + \left(i - \frac{1}{2}\right)\frac{b-a}{n}\right) \right| \le \frac{(b-a)^3}{24}\frac{1}{n^2}\|f''\|_\infty$$

für alle $n \in \mathbb{N}$.

Trapezverfahren. Für das zusammengesetzte Trapezverfahren ($n = m + 1$) gilt mit der Abkürzung

$$h = \frac{b-a}{n-1}$$

die Fehlerabschätzung

$$\left| \int_a^b f(x)\,dx - h\left(\frac{f(a)}{2} + \sum_{i=1}^{n-2} f\left(a + ih\right) + \frac{f(b)}{2}\right) \right| \le \frac{(b-a)^3}{12}\frac{1}{(n-1)^2}\|f''\|_\infty$$

für alle $n \ge 2$.

Simpsonverfahren. Für das zusammengesetzte Simpsonverfahren gilt mit den Abkürzungen

$$m \in \mathbb{N}, \qquad n = 2m + 1, \qquad h = \frac{b-a}{n-1}, \qquad x_i = a + (i-1)h$$

die Fehlerabschätzung

$$\left| \int_a^b f(x)\,dx - \frac{h}{3}\left(f(a) + 4\sum_{\mu=1}^{m} f(x_{2\mu}) + 2\sum_{\nu=1}^{m-1} f(x_{2\nu+1}) + f(b) \right) \right| \leq \frac{(b-a)^5}{180}\frac{1}{(n-1)^4}\|f^{(4)}\|_\infty.$$

Der allgemeine Satz ergibt zunächst

$$|S_{2m+1}(f) - S(f)| \leq \frac{1}{2880} \cdot (b-a)^5 \cdot m^{-4} \cdot \|f^{(4)}\|_\infty.$$

Wegen $m = (n-1)/2$ ergibt sich

$$\frac{1}{m^4} = \frac{16}{(n-1)^4}$$

und daraus folgt die behauptete Abschätzung.

Für genügend glatte Funktionen konvergiert also der Fehler dieser drei Methoden wie n^{-2} bzw. n^{-4} gegen 0.

4.4 Universelle Quadraturverfahren

Problemstellung. Wir haben gesehen, daß für die Funktionenmengen F^k eine Folge $(S_n)_{n\in\mathbb{N}}$ von Quadraturformeln mit optimaler Konvergenzordnung n^{-k} existiert. Bisher haben wir jedoch diese Folge nur in Abhängigkeit von k konstruiert. Jetzt wollen wir nach der Existenz einer Folge $(S_n)_n$ von Quadraturformeln fragen, die für jedes F^k optimale Konvergenzordnung hat, d.h.

$$\Delta_{\max}(S_n) \leq c_k \cdot n^{-k}$$

für jedes F^k mit $k \in \mathbb{N}$ und genügend große n.

Definition (Universelle Quadraturverfahren)**.** Eine Folge $(S_n)_{n\in\mathbb{N}}$ von Quadraturformeln heißt ein universelles Quadraturverfahren (zu den F^k mit $k \in \mathbb{N}$), falls für alle k ein $c_k > 0$ und ein $n_k \in \mathbb{N}$ existieren, so daß für $n \geq n_k$

$$\Delta_{\max}(S_n) = \sup_{f\in F^k} |S(f) - S_n(f)| \leq c_k \cdot n^{-k}$$

gilt.

Die Gauß-Formeln. Wir werden in Satz 33 zeigen, daß es ein universelles Quadraturverfahren gibt und daß speziell die *Gauß'schen Quadraturformeln* so eine Folge von S_n sind. Dazu definieren wir zunächst die Gauß-Formeln und zeigen ihre wichtigsten Eigenschaften.

Sei $S(f) = \int_a^b f(x)\,dx$ und seien die $x_i \in [a,b]$ für $i = 1, \ldots, n$ paarweise verschiedene Knoten. Wir haben gesehen, daß es genau ein S_n mit den Knoten $x_1, \ldots, x_n$ gibt, so daß $S(p) = S_n(p)$ für alle Polynome $p \in P_n$. Für dieses S_n gilt

$$a_i = \int_a^b \prod_{j \neq i}^{n} \frac{x - x_j}{x_i - x_j}\,dx.$$

Wir zeigen nun, daß bei geeigneter Wahl der Knoten $x_1, \ldots, x_n$ sogar alle $p \in P_{2n}$ exakt integriert werden.

Satz 32 (Eigenschaften der Gauß-Formel)**.** *Sei $q \in P_{n+1}$ mit $q \neq 0$ und*

$$\int_a^b q(x)p(x)\,dx = 0$$

für alle Polynome $p \in P_n$. So ein q existiert und ist bis auf eine multiplikative Konstante eindeutig bestimmt. Weiter gilt:

a) das Polynom q hat n verschiedene Nullstellen $x_1, \ldots, x_n$, die alle reell sind und im Intervall $[a,b]$ liegen;

b) die interpolatorische Quadraturformel

$$S_n(f) = \sum_{i=1}^{n} a_i f(x_i)$$

(mit den Nullstellen x_i von q) ist exakt für alle $p \in P_{2n}$; die Gewichte a_i sind positiv mit

$$\sum_{i=1}^{n} a_i = b - a;$$

c) Es gibt kein S_n, das für alle $p \in P_{2n+1}$ exakt ist, d.h. die bei b) angegebene Gauß-Formel hat maximalen Exaktheitsgrad.

Beweis: Statt $\int_a^b q(x)p(x)dx = 0$ für alle $p \in P_n$, wobei $q \in P_{n+1} \setminus P_n$, könnten wir auch schreiben $P_{n+1} = P_n \oplus \langle q \rangle$ im Sinne einer orthogonalen direkten Summe. Also existiert so ein q.

a) Wegen $1 \in P_n$ gilt $\int_a^b q(x)dx = 0$ und daher wechselt q sein Vorzeichen mindestens einmal. Ein Widerspruchsbeweis zeigt, daß es tatsächlich n Vorzeichenwechsel im Inneren von $[a,b]$ geben muß.

b) Sei nun $f \in P_{2n}$. Dividiert man mit Rest durch q, so erhält man

$$f = qp + r \qquad \text{mit } p, r \in P_n.$$

Damit folgt

$$f(x_i) = r(x_i).$$

Also

$$\int_a^b f(x)dx = \int_a^b q(x)p(x)dx + \int_a^b r(x)dx = \int_a^b r(x)dx = \sum_{i=1}^{n} a_i r(x_i) = \sum_{i=1}^{n} a_i f(x_i).$$

Zur Positivität der a_i betrachte die Funktionen

$$p_i(x) = \prod_{j \neq i}^{n} (x - x_j)^2 \in P_{2n-1}.$$

Damit hat man

$$0 < S(p_i) = S_n(p_i) = \prod_{j \neq i}^{n} (x_i - x_j)^2 \cdot a_i.$$

Also sind die a_i positiv. Durch Einsetzen des Polynoms $p = 1$ erhält man die Summe der Gewichte a_i.

c) Sei nun eine beliebige Quadraturformel S_n gegeben, $S_n(f) = \sum_{i=1}^{n} a_i f(x_i)$. Für

$$f(x) = \prod_{i=1}^{n} (x - x_i)^2 \in P_{2n+1}$$

gilt

$$S(f) \neq 0 = S_n(f).$$

Daher kann S_n nicht für alle $f \in P_{2n+1}$ exakt sein.

Bezeichnung. Die in Satz 32 betrachtete Quadraturformel heißt Gauß'sche Quadraturformel S_n^G.

Für die Zahlen

$$E_n(f) := \inf_{p \in P_n} \|f - p\|_\infty$$

ist folgendes bekannt. Ist $f \in C([a, b])$, so gilt

$$\lim_{n \to \infty} E_n(f) = 0.$$

Dies ist lediglich eine andere Schreibweise für den Approximationssatz von Weierstraß. Für $f \in C^k([a, b])$ und $n \geq k$ gilt, wie früher schon erwähnt, sogar die Fehlerabschätzung

$$E_n(f) \leq c_k \cdot \|f^{(k)}\|_\infty \cdot n^{-k}.$$

Dies ist der Satz von Jackson, siehe Satz 19. Dabei ist $c_k > 0$ nicht von f abhängig.

Satz 33 (Fehlerabschätzung für die Gauß-Formel)**.** *Sei $f \in C([a, b])$ und sei S_n^G die Gauß-Formel. Dann gilt*

$$|S(f) - S_n^G(f)| \leq 2(b - a) E_{2n}(f).$$

Mit dem Satz von Jackson folgt insbesondere, daß die Folge der S_n^G ein universelles Quadraturverfahren für die Funktionenklassen F^k ist.

Beweis: Für alle Polynome $p \in P_{2n}$ gilt

$$|S(f) - S_n^G(f)| = |S(f-p) - S_n^G(f-p)| \leq |S(f-p)| + |S_n^G(f-p)| \leq 2(b-a)\|f-p\|_\infty.$$

Hieraus folgt

$$|S(f) - S_n^G(f)| \leq 2(b-a)E_{2n}(f).$$

Das Romberg-Verfahren. Der Vorteil der Gauß-Formeln ist die Universalität für die F^k. Ein Nachteil ist jedoch die relativ aufwendige Bestimmung der Knoten und Gewichte. Daher werden oft andere Verfahren betrachtet, die ebenfalls universell sind (mit i.a. etwas größeren Fehlern im Vergleich zu den Gauß-Formeln) und leichter zu programmieren sind. Wir beginnen mit dem Romberg-Verfahren.

Für $m \in \mathbb{N},\ n = 2^m + 1,\ f : [a,b] \to \mathbb{R},\ k \in \{0,\ldots,m\}$ und $l \in \{0,\ldots,k\}$ definiert man induktiv:

$$\begin{aligned} R(0,0) &= \frac{1}{2} \cdot (b-a)\,(f(a) + f(b))\,, \\ R(k+1,0) &= \frac{1}{2} \cdot R(k,0) + \frac{b-a}{2^{k+1}} \sum_{i=1}^{2^k} f\left(a + \frac{(2i-1)(b-a)}{2^{k+1}}\right) \quad \text{und} \\ R(k,l) &= R(k,l-1) + \frac{1}{4^l - 1} \cdot (R(k,l-1) - R(k-1,l-1))\,. \end{aligned}$$

Dann heißt $R(m,m)$ die Romberg-Summe für $n = 2^m + 1$ Funktionswerte. Dabei sind die $R(k,0)$ gerade die Trapezsummen mit 2^k Teilintervallen. Hier gelten die Fehlerabschätzungen

$$\begin{aligned} S(f) - R(k-1,0) &= -\frac{1}{12}\frac{(b-a)^3}{2^{2k-2}} f''(\xi_1) \quad \text{und} \\ S(f) - R(k,0) &= -\frac{1}{12}\frac{(b-a)^3}{2^{2k}} f''(\xi_2). \end{aligned}$$

Wenn die betrachteten Intervalle klein sind, so hofft man, daß $f''(\xi_1) \approx f''(\xi_2)$ ist. Dann kann man aus den beiden Gleichungen einen neuen, besseren Schätzwert für $S(f)$ ermitteln. Dieser ist dann $R(k,1)$. Durch Wiederholen dieser Extrapolationsmethode erhält man die Romberg-Summen.

Die Tatsache, daß man ein universelles Verfahren erhält, wird hier nicht gezeigt. Einen Beweis findet man in [22].

Die Verfahren von Polya, Filippi und Clenshaw-Curtis. Für jedes $n \in \mathbb{N}$ sei ein S_n definiert mit folgenden Eigenschaften:

a) S_n sei interpolatorisch, d.h. $S(p) = S_n(p)$ für $p \in P_n$;

b) die Gewichte a_i in der Formel S_n seien alle positiv.

Dann gilt für jedes f und jedes $p \in P_n$ die Abschätzung

$$|S(f) - S_n(f)| = |S(f-p) - S_n(f-p)| \leq |S(f-p)| + |S_n(f-p)| \leq 2(b-a)\|f-p\|_\infty.$$

Hieraus folgt

$$|S(f) - S_n(f)| \leq 2(b-a)E_n(f).$$

Insbesondere folgt wie bei den Gauß-Formeln, daß die Folge $(S_n)_n$ universell für die F^k ist. Man sucht daher nach solchen S_n, die zusätzlich Knoten und Gewichte haben sollen, die sich einfach beschreiben lassen. Wir definieren die wichtigsten derartigen Verfahren durch Angabe ihrer Knoten, wobei wir das Standardintervall $[a,b] = [-1,1]$ zugrundelegen. Wir verzichten aber auf den Beweis der Tatsache, daß die Gewichte alle positiv sind. Dies kann im Buch von Brass [4] über Quadraturverfahren nachgelesen werden.

a) Polya-Verfahren: Die Knoten von S_n sind hier gegeben durch

$$x_i = -\cos\frac{(2i-1)\pi}{2n},$$

d.h. man wählt die Tschebyscheff-Knoten wie bei Satz 12. Hier und bei den folgenden Formeln gilt $i = 1, \ldots, n$.

b) Filippi-Verfahren: Die Knoten von S_n sind hier gegeben durch

$$x_i = -\cos\frac{i\pi}{n+1}.$$

c) Clenshaw-Curtis-Verfahren: Die Knoten von S_n sind hier gegeben durch

$$x_i = -\cos\frac{(i-1)\pi}{n-1},$$

wobei $n > 1$.

Zusätze und Bemerkungen.

- Auch in diesem Kapitel haben wir nur den eindimensionalen Fall, $f : [a,b] \to \mathbb{R}$, diskutiert. Im Mittelpunkt der Forschung zur Numerischen Integration steht heute die Berechnung mehrdimensionaler Integrale, etwa

  $$S(f) = \int_{[0,1]^d} f(x)\,dx.$$

  Bei wichtigen Anwendungen, etwa in der Finanzmathematik oder der Physik, ist die Dimension d groß, zum Beispiel 360. (Die Zahl 360 ergibt sich bei gewissen Finanzderivaten, die auf Hypotheken mit der typischen Laufzeit von 30 Jahren beruhen; der Zinssatz in jedem Monat entspricht einer Variablen.) Wir haben bewiesen, daß die optimale Konvergenzordnung bei der Integration von C^k-Funktionen auf einem Intervall n^{-k} ist, siehe Satz 31. Diese Konvergenzordnung beträgt für $d > 1$ nur $n^{-k/d}$ und diese Rate ist extrem langsam im Fall $d \gg k$. Man spricht hier vom *Fluch der Dimension* oder *curse of dimension*, siehe [19] für eine Übersicht.

  Da sich diese Rate auf *beliebige* (deterministische) Algorithmen bezieht, ist das Problem prinzipiell schwer. Eine wesentliche Verbesserung läßt sich allerdings mit *randomisierten Algorithmen* (sog. *Monte-Carlo-Methoden*) erzielen. Dann ist die optimale Konvergenzordnung $n^{-k/d-1/2}$, also größer als $1/2$, siehe [16]. Noch schneller geht es mit *Quantencomputern*, die es aber z.Z. nur auf dem Papier gibt. Damit würde man die Rate $n^{-k/d-1}$ bekommen, siehe [20].

Aufgaben

4.1. Stellen Sie die Funktion

$$S(x) = \int_0^x e^{-t^2}\, dt$$

für $x \geq 0$ in der Form einer unendlichen Reihe dar. Wieviele Terme dieser Reihe benötigt man, wenn man $S(1)$ bzw. $S(10)$ auf diese Weise berechnen will mit einem Fehler, der kleiner ist als 10^{-8}?

4.2. Gesucht ist eine Quadraturformel

$$S_4(f) = \sum_{i=1}^{4} a_i f(x_i)$$

mit $x_i = (i-1)/3$ zur Approximation von $S(f) = \int_0^1 f(x)dx$. Berechnen Sie die Gewichte so, daß

$$S_4(f) = S(\widetilde{f}),$$

wobei

a) $\widetilde{f} \in P_4$ ist das Polynom mit $\widetilde{f}(x_i) = f(x_i)$;

b) $\widetilde{f}$ ist der natürliche kubische Spline mit (Knoten x_i und) $\widetilde{f}(x_i) = f(x_i)$.

4.3. Das Integral $S(f) = \int_{-1}^1 f(x)\, dx$ soll durch eine Quadraturformel

$$S_2(f) = \sum_{i=1}^{2} a_i f(x_i)$$

approximiert werden, wobei $x_1 \neq x_2$.

a) Wie müssen die a_i definiert werden, damit (bei gegebenen x_i) Polynome $p \in P_2$ exakt integriert werden?

b) Jetzt sei $x_2 = -x_1$. Schätzen Sie $\Delta_{\max}(S_2)$ für

$$F^2 = \{f \in C^2([-1,1]) \mid \|f''\|_\infty \leq 1\}$$

mit Hilfe von Satz 27 der Vorlesung ab. Hierbei sei S_2 die in a) gefundene Formel.

4.4. Fortsetzung von Aufgabe 4.3: Berechnen Sie $\Delta_{\max}(S_2)$ mit Hilfe von Satz 29 der Vorlesung.

4.5. Es sei S_{2m+1} eine interpolatorische Quadraturformel für $S(f) = \int_a^b f(x)\, dx$, d.h. es gelte $S_{2m+1}(p) = S(p)$ für alle $p \in P_{2m+1}$. Weiter seien die paarweise verschiedenen Knoten x_i von S_{2m+1} symmetrisch angeordnet, d.h. es gelte $x_i - a = b - x_{2m+2-i}$ für alle i. Zeigen Sie:

a) Für die Gewichte gilt dann $a_i = a_{2m+2-i}$ für alle i.

b) Die Formel S_{2m+1} ist sogar exakt für alle $p \in P_{2m+2}$.

Folgern Sie, daß die Simpson-Formel Polynome vom Grad kleiner gleich 3 exakt integriert.

Hinweis: Sie können den Fall $[a,b] = [-1,1]$ betrachten, dann wird es einfacher.

4.6. Berechnen Sie näherungsweise die Integrale

$$\int_0^1 \frac{1}{1+x}\,dx$$

und

$$\int_0^1 \sqrt{x}\,dx.$$

Benützen Sie dazu die Trapezregel, die Simpsonregel und das Romberg-Verfahren, jeweils mit $n = 2, 3, 5, 9$ und 17 Knoten. Kommentieren Sie die Ergebnisse hinsichtlich ihrer Genauigkeit.

4.7. Die folgenden Integrale sollen näherungsweise bestimmt werden, der Fehler soll garantiert kleiner sein als ein beliebig vorgegebenes $\varepsilon > 0$:

$$\int_0^\infty \frac{e^{-x}}{1+x^3}\,dx \qquad \text{bzw.} \qquad \int_0^1 \sqrt{e^x - 1}\,dx.$$

Beachten Sie beim zweiten Integral, daß der Integrand für $x = 0$ nicht differenzierbar ist.

Gefragt ist hier nach einem Lösungsweg, keine Programmieraufgabe. Warum macht es Sinn, beim zweiten Integral die Umkehrfunktion des Integranden zu betrachten?

4.8. Das Integral

$$S(f) = \int_0^1 f(x) \cdot x\,dx$$

soll so durch ein S_2 der Form

$$S_2(f) = a_1 f(x_1) + a_2 f(x_2)$$

approximiert werden, daß $S(f) = S_2(f)$ für alle Polynome vom Grad kleiner als 4. Wie müssen die Knoten und die Gewichte gewählt werden? Bestimmen Sie dazu zunächst ein $q \in P_3$ mit $q \neq 0$ und

$$\int_0^1 p(x) \cdot q(x) \cdot x\,dx = 0$$

für alle $p \in P_2$.

Kapitel 5

Lineare Gleichungssysteme

Vorbemerkung. Bisher haben wir uns auf die Diskussion von *Verfahrensfehlern* konzentriert, die durch unvollständige Information entstanden. *Rundungsfehler* waren bisher meist relativ leicht zu durchschauen.

Jetzt, bei linearen Gleichungssystemen, ist die Information in der Regel vollständig, denn der Raum ist ja nur endlichdimensional. Daher ist es möglich (und üblich), Verfahren zu verwenden, die keinen Verfahrensfehler haben. Das bekannteste dieser Verfahren ist das *Gauß-Verfahren*.

Daneben werden allerdings auch Verfahren betrachtet, die einen positiven Verfahrensfehler haben, etwa iterative Verfahren und die Tikhonov-Regularisierung. Damit kann man manchmal schnellere Algorithmen konstruieren und/oder erhält unter gewissen Bedingungen Ergebnisse mit kleineren *Gesamtfehlern*.

5.1 Das Gauß'sche Eliminationsverfahren

Wir betrachten Gleichungssysteme der Form

$$Ax = b \qquad \text{mit} \quad A \in \mathbb{R}^{n\times n} \text{ und } \quad b \in \mathbb{R}^n,$$

wobei wir meist voraussetzen, daß A regulär ist. Wir nennen zwei Gleichungssysteme dieser Gestalt äquivalent, wenn sie die gleichen Lösungen haben. Man versucht, ein gegebenes System durch einfache Umformungen in ein dazu äquivalentes, aber leicht lösbares, zu verwandeln.

Elementare Umformungen. Die folgenden Operationen heißen elementare Umformungen von Gleichungssystemen: Vertauschen von zwei Gleichungen eines Systems. Multiplizieren einer Gleichung mit einer Konstanten ungleich Null. Addieren eines Vielfachen einer Gleichung zu einer anderen.

Ähnlich sind *elementare Umformungen* von Matrizen definiert: Vertauschen von zwei Zeilen einer Matrix. Multiplizieren einer Zeile mit einer Konstanten ungleich Null. Addieren eines Vielfachen einer Zeile zu einer anderen. Es gilt folgendes Lemma.

Lemma. *Erhält man das System $Bx = c$ aus dem System $Ax = b$ durch endlich viele elementare Umformungen, so sind die beiden Systeme äquivalent. Ist eine Matrix*

$A \in \mathbb{R}^{n\times n}$ regulär, so läßt sie sich durch elementare Umformungen auf die Einheitsmatrix transformieren. Jeder elementaren Umformung einer Matrix entspricht eine Multiplikation der Matrix von links mit einer sogenannten Elementarmatrix.

Invertierbarkeit. Im Folgenden sei $I = I_n$ die Einheitsmatrix. Wenn die in dem System vorkommende Matrix A invertierbar ist, so hat man die eindeutige Lösung $x = A^{-1}b$. Die folgenden Aussagen sind aus der Linearen Algebra bekannt.

Lemma (Reguläre Matrizen, symmetrische Matrizen)**.** *Für $A \in \mathbb{R}^{n\times n}$ sind äquivalent:*
– *A ist invertierbar.*
– $\det A \neq 0$.
– *$A : \mathbb{R}^n \to \mathbb{R}^n$ ist surjektiv.*
– *$A : \mathbb{R}^n \to \mathbb{R}^n$ ist injektiv.*
– *Spaltenrang $A = n$.*
– *Zeilenrang $A = n$.*
– *A ist Produkt von Elementarmatrizen (zu elementaren Umformungen).*
– *0 ist kein Eigenwert von A.*

Für eine selbstadjungierte oder symmetrische Matrix A sind äquivalent:
– *A ist positiv definit, d.h. $x^tAx > 0$ für alle $x \neq 0$.*
– *Alle Eigenwerte von A sind positiv.*
– *Alle Hauptminoren A_k haben positive Determinante.*

Eine symmetrische Matrix A ist diagonalisierbar und es gibt eine Orthonormalbasis aus Eigenvektoren.

Einfach zu lösende Gleichungssysteme. Der einfachste Fall ist natürlich, daß A eine Diagonalmatrix ist. Für $a_{ii} \neq 0$ erhält man die eindeutige Lösung $x_i = \frac{b_i}{a_{ii}}$. Andernfalls hat man entweder keine oder unendlich viele Lösungen. Aber auch wenn A eine (untere oder obere) Dreiecksmatrix ist, ist das Gleichungssystem noch leicht lösbar.

Die LU-Zerlegung. Sei $A = LU$, wobei L untere Dreiecksmatrix und U obere Dreiecksmatrix. Dann ist das System $Ax = b$ äquivalent zu den zwei Systemen $Lz = b$ und $Ux = z$. Diese beiden Systeme können leicht gelöst werden. Aber nicht jede Matrix A läßt sich so zerlegen und wenn so eine Zerlegung existiert, so ist sie nicht eindeutig.

Existenz der LU-Zerlegung. Wenn $A = LU$ in Form einer LU-Zerlegung dargestellt werden kann, so gilt für alle i, j die Gleichung

$$a_{ij} = \sum_{s=1}^{\min\{i,j\}} l_{is}u_{sj}. \tag{5.1}$$

Wir besprechen die Lösbarkeit von (5.1), wobei wir zusätzlich fordern, daß für alle i $u_{ii} = 1$ und $l_{ii} \neq 0$ gelten soll.

Um induktiv eine Lösung zu finden, gehen wir davon aus, daß für $k \in \{1, \dots, n\}$ die ersten $k-1$ Zeilen von U und die ersten $k-1$ Spalten von L bereits bekannt sind.

Dann gelten die Beziehungen

$$a_{kk} = \sum_{s=1}^{k-1} l_{ks} u_{sk} + l_{kk} u_{kk},$$

$$k+1 \le j \le n: \qquad a_{kj} = \sum_{s=1}^{k-1} l_{ks} u_{sj} + l_{kk} u_{kj},$$

$$k+1 \le j \le n: \qquad a_{jk} = \sum_{s=1}^{k-1} l_{js} u_{sk} + l_{jk} u_{kk}.$$

Man sieht, daß die erste Gleichung – Lösbarkeit des ganzen Systems vorausgesetzt – sofort die neuen Diagonalelemente liefert. Den Rest der Zeile von U und der Spalte von L kann man dann mit den beiden anderen Gleichungen bestimmen.

Satz 34 (Existenz der LU-Zerlegung)**.** *Wenn für eine Matrix A alle Hauptminoren A_k, d.h. alle Untermatrizen der Form*

$$A_k = \begin{pmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{pmatrix}$$

$(k = 1, \ldots, n)$ nichtsingulär sind, dann hat A genau eine LU-Zerlegung mit den oben geforderten Eigenschaften.

Beweis: Analog zu A_k definieren wir die Untermatrizen U_k und L_k. Wir nehmen an, daß alle A_k nichtsingulär sind. Dann sind, wenn man im Induktionsbeweis die Existenz der Zerlegung für k voraussetzt, wegen $A_k = L_k U_k$ auch die Untermatrizen L_k und U_k nichtsingulär. Insbesondere gilt also $l_{kk} u_{kk} \neq 0$. Dann sind aber die oben aufgestellten Gleichungssysteme allesamt eindeutig lösbar und man erhält U_{k+1} und L_{k+1}.

Satz 35 (Cholesky-Zerlegung)**.** *Ist A eine reelle, symmetrische, positiv definite Matrix, so gibt es eine eindeutig bestimmte Faktorisierung der Form $A = LL^t$ mit einer unteren Dreiecksmatrix L, deren Diagonalelemente alle positiv sind.*

Beweis: Für ein solches A gilt $x^t A x > 0$ für $x \neq 0$. Betrachtet man Vektoren der Form $(x_1, \ldots, x_k, 0, \ldots, 0)$, so folgt, daß für alle k die Untermatrix A_k nichtsingulär ist. Es existiert also eine LU-Zerlegung. Wir erhalten dieselben Gleichungen wie oben, allerdings wählen wir jetzt die Normierung $l_{kk} = u_{kk}$. Dies ist möglich, weil

$$l_{kk} u_{kk} > 0$$

für alle k. Das folgt aus der Tatsache

$$\det A_k = \det L_k \det U_k > 0,$$

für alle k. Mit der neuen Normierung ergeben sich wegen $a_{kj} = a_{jk}$ für u_{kj} und für l_{jk} jeweils dieselben Gleichungen, die wegen der Eindeutigkeit auch zu denselben Lösungen führen.

Bemerkung. Beim Cholesky-Verfahren ergibt sich

$$a_{kk} = \sum_{s=1}^{k} l_{ks}^2,$$

insbesondere $|l_{ks}| \leq \sqrt{a_{kk}}$. Die Elemente von L sind also nicht zu groß. Dies wirkt sich günstig auf Rundungsfehler aus.

Gauß-Elimination und LU-Zerlegung. Das Gauß'sche Eliminationsverfahren für lineare Gleichungssysteme dürfte wohl aus der linearen Algebra bekannt sein. Im grundlegenden Algorithmus arbeitet man ohne Zeilenvertauschungen: man addiert ein Vielfaches einer Zeile i_1 zu einer anderen Zeile i_2, wobei jedesmal $i_1 < i_2$. Dies entspricht einer Multiplikation der Matrix von links mit einer Elementarmatrix, die zugleich eine untere Dreiecksmatrix ist. Auf diese Weise konstruiert man eine obere Dreiecksmatrix U, es gilt schließlich

$$L_m \dots L_1 A = U$$

oder

$$A = (L_1^{-1} \dots L_m^{-1})U = LU,$$

wobei $L = (L_1^{-1} \dots L_m^{-1})$ eine untere Dreiecksmatrix ist. Daher ist das einfache Gauß-Verfahren (ohne Zeilenvertauschungen) äquivalent zur LU-Zerlegung und wie diese nicht immer möglich.

Pivotsuche. Das Gauß-Verfahren in seiner einfachsten Form ist nicht befriedigend, wie das folgende Beispiel zeigt. Ist $a_{11} = 0$ so ist der Algorithmus nicht anwendbar. Betrachte das System

$$\begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Die Lösung ist

$$x_1 = \frac{1}{1-\varepsilon} \approx 1, \qquad x_2 = \frac{1-2\varepsilon}{1-\varepsilon} \approx 1.$$

Das Gauß-Verfahren ergibt jedoch

$$x_2 = \frac{2 - \frac{1}{\varepsilon}}{1 - \frac{1}{\varepsilon}} \approx 1, \qquad x_1 = \frac{1 - x_2}{\varepsilon} \approx 0.$$

Hier liegt der Fehler daran, daß $|a_{11}|$ zu klein ist im Verhältnis zu $|a_{12}|$. Die numerischen Schwierigkeiten verschwinden bei diesem Beispiel, wenn man die Reihenfolge der Gleichungen vertauscht.

Bei einem guten Algorithmus müssen die Gleichungen unter gewissen Umständen vertauscht werden. Oder, anders formuliert, die Pivotzeile muß geeignet gewählt werden. Es gibt verschiedene Pivotstrategien die darin übereinstimmen, daß das sogenannte

Pivotelement relativ (im Vergleich zu anderen Elementen derselben Zeile oder Spalte) groß ist. Man überlegt sich leicht, daß das Gauß-Verfahren bei jeder regulären Matrix A zur Lösung von $Ax = b$ führt, sofern geeignete Zeilenvertauschungen durchgeführt werden.

Diagonaldominanz. Bei gewissen Matrizen ist eine Pivotsuche nicht erforderlich, zum Beispiel bei diagonaldominanten Matrizen. Eine Matrix A heißt *diagonaldominant*, wenn

$$|a_{ii}| > \sum_{j \neq i}^{n} |a_{ij}|$$

für alle i gilt.

Wenn A diagonaldominant ist, so gilt:
– Die Diagonaldominanz bleibt bei der Gauß-Elimination erhalten.
– A ist regulär und hat eine LU-Zerlegung.

Nachiteration. Auch die Gauß-Elimination mit Pivotsuche ist nicht immer numerisch stabil, das heißt auch hier können kleine Fehler (die während der Rechnung durch Runden entstehen) eine große Wirkung (gemessen an den Fehlern, die man aufgrund der Kondition der Matrix mindestens zu erwarten hat) haben.

Es sei x^0 der Vektor, den man bei der Lösung des Systems $Ax = b$ numerisch erhält. x^* sei die exakte Lösung und r^0 der Fehler. Dann gilt

$$A(x^0 + r^0) = b \qquad \Longrightarrow \qquad Ar^0 = \underbrace{b - Ax^0}_{\text{bekannt}}.$$

Wenn nun $b - Ax^0$ nicht 0 ist, kann man r^0 näherungsweise berechnen und damit einen (vielleicht) besseren Näherungswert für x^* erhalten. Dieses Verfahren heißt Nachiteration.

Bemerkung. Das Gauß-Verfahren mit geeigneter Pivotsuche und einmaliger Nachiteration ergibt einen „einigermaßen stabilen“ Algorithmus. Dies wurde von Skeel in einem präzisen Sinn gezeigt, siehe etwa [7].

Man sollte aber genau auf die Bedeutung dieser Aussage achten: Wenn das Problem $(A, b) \mapsto x = A^{-1}b$ schlecht konditioniert ist, so führen winzige Änderungen bei A oder b i.a. bereits zu großen Änderungen der wahren Lösung x. Berechnet man diese Lösung mit einem stabilen Algorithmus, so ist lediglich gewährleistet, daß sich kleine Rundungsfehler während der Rechnung nicht viel mehr auf das Ergebnis auswirken als kleine Fehler bei der Eingabe – die sich aber extrem auf das Ergebnis auswirken können!

Mit Hilfe der Tikhonov-Regularisierung kann man Algorithmen konstruieren, die auch bei schlecht konditionierten Problemen häufig noch zu brauchbaren Lösungen führen. Dazu benötigt man aber zusätzliche Informationen über die Lösung. Wir kommen hierauf zurück.

5.2 Zur Kondition von linearen Gleichungssystemen

Bemerkung. Wir betrachten das Problem, wie sich kleine Änderungen bei A und bei b auf die wahre Lösung des Gleichungssystems $Ax = b$ auswirken können. Als Abbildung betrachten wir also

$$S : (A, b) \mapsto x = A^{-1}b \qquad S : \{A \in \mathbb{R}^{n\times n} \mid A \text{ regulär}\} \times \mathbb{R}^n \to \mathbb{R}^n.$$

Daneben werden aber auch die einfachen Abbildungen betrachtet, wo wir A oder b festlassen, d.h.

$$S : b \mapsto x = A^{-1}b \qquad S : \mathbb{R}^n \to \mathbb{R}^n,$$
$$S : A \mapsto x = A^{-1}b \qquad S : \{A \in \mathbb{R}^{n\times n} \mid A \text{ regulär}\} \to \mathbb{R}^n,$$

wenn wir die Abhängigkeit der Lösung entweder nur von A oder nur von b studieren wollen.

Ist $S : F \to G$ beliebig, so kann man die relative normweise Konditionszahl von S definieren durch

$$K_{\text{rel}}(f_1) = \lim_{\varepsilon\to 0} \sup_{0<\|f_1-f_2\|<\varepsilon} \frac{\|S(f_1) - S(f_2)\|}{\|S(f_1)\|} : \frac{\|f_1 - f_2\|}{\|f_1\|}.$$

Die Zahl $K_{\text{rel}}(f_1)$ gibt an, mit welchem Verstärkungsfaktor der relative Fehler $\frac{\|f_1-f_2\|}{\|f_1\|}$ bei den Daten verstärkt werden kann zu einem relativen Fehler $\frac{\|S(f_1)-S(f_2)\|}{\|S(f_1)\|}$ im Ergebnis. Hierbei wird angenommen, daß der Fehler „sehr klein“ ist.

Im Fall endlichdimensionaler Räume F und G ist der vorgegebene Fehler bei $f = (f_1, \ldots, f_m) \in \mathbb{R}^m$ oft koordinatenweise relativ klein (etwa beim Rechnen mit fester Stellenzahl). Für den Näherungswert $\widetilde{f} = (\widetilde{f_1}, \ldots, \widetilde{f_m})$ gilt

$$\frac{|\widetilde{f_i} - f_i|}{|f_i|} \le \delta$$

für alle i. Dem Rechnen mit fester Stellenzahl entsprechen daher die komponentenweisen relativen Konditionszahlen (siehe Kapitel 1) besser. Dennoch betrachten wir die normweisen relativen Konditionszahlen, da sie einfacher zu berechnen und in der Numerik üblich sind.

Normen für Vektoren und Matrizen. Die wichtigsten Normen im $\mathbb{R}^n$ sind

$$\|x\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2} \qquad \text{und} \qquad \|x\|_\infty = \max_{i\in\{1,\ldots,n\}} |x_i|.$$

Ebenfalls wichtig ist die Norm $\|x\|_1 = \sum_{i=1}^n |x_i|$. Allgemein gelten für Normen die folgenden Eigenschaften.

$\|x\| \ge 0$

$\|x\| = 0 \iff x = 0$
$\|\alpha x\| = |\alpha|\|x\|$
$\|x + y\| \leq \|x\| + \|y\|$.

Wir betrachten nun eine lineare Abbildung $L : \mathbb{R}^n \to \mathbb{R}^n$. Legen wir im Bild- und Urbildbereich jeweils die Standardbasis zugrunde, so entspricht der Abbildung L eine Matrix $A \in \mathbb{R}^{n\times n}$. Es gilt dann $L(x) = Ax$ und wir identifizieren L mit A.

Lemma (Operatornorm)**.** *Gegeben sei eine beliebige Norm im $\mathbb{R}^n$. Weiter sei $\mathbb{R}^{n\times n}$ der Raum der linearen Abbildungen (oder Matrizen). Dann ist auf $\mathbb{R}^{n\times n}$ eine Norm gegeben durch*

$$\|A\| = \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|.$$

Diese Norm heißt die zur Vektornorm im $\mathbb{R}^n$ gehörige Operatornorm oder Matrixnorm. Für den nächsten Satz benötigen wir eine Bezeichnung. Ist $B \in R^{n\times n}$ eine quadratische Matrix und sind $\lambda_1, \ldots, \lambda_n$ die komplexen Eigenwerte von B, so heißt

$$\varrho(B) = \max_i |\lambda_i|$$

Spektralradius von B. Mit seiner Hilfe läßt sich die zur euklidischen Norm gehörige Operatornorm charakterisieren.

Satz 36 (Beschreibung wichtiger Operatornormen)**.** *Die zur $\| \ \|_2$-Norm gehörige Operatornorm ist die Spektralnorm der Matrix A*

$$\|A\|_2 = \sqrt{\varrho(A^t A)}.$$

Die zur $\| \ \|_\infty$-Norm gehörige Operatornorm ist für eine Matrix $A = (a_{ij})$ gegeben durch die Zeilensummennorm

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|.$$

Die zur $\| \ \|_1$-Norm gehörige Operatornorm ist für eine Matrix $A = (a_{ij})$ gegeben durch die Spaltensummennorm

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|.$$

Beweis: Wir führen den Beweis nur für die $\| \ \|_2$-Norm, der Beweis für die übrigen Normen ist einfacher. Wegen $(A^t A)^t = A^t A$ ist $A^t A$ symmetrisch. Weiter gilt

$$x^t A^t A x = \|Ax\|_2^2 \geq 0.$$

Also ist $A^t A$ selbstadjungiert und positiv semidefinit. Bekanntlich läßt sich eine solche Matrix orthogonal diagonalisieren. Das heißt, es existiert $U \in \mathbb{R}^{n\times n}$ mit

$$U^t A^t A U = D \text{ diagonal und}$$

$$U^t U = I.$$

Daraus folgt

$$\|A\|_2^2 = \sup_{\|x\|_2=1} \|Ax\|_2^2 = \sup_{\|Uy\|_2=1} (AUy)^t(AUy)$$

$$= \sup_{\|y\|_2=1} y^t(U^tA^tAU)y = \sup_{\|y\|_2=1} y^tDy = \sup_{\|y\|_2=1} \sum_{i=1}^n y_i^2 d_{ii} = \max_{1\le i\le n} d_{ii} = \varrho(A^tA).$$

Bemerkung. Aus geometrischen Gründen ist die 2-Norm am schönsten. So gilt für orthogonale Matrizen U und T

$$\|Ux\|_2 = \|x\|_2 \qquad \text{und} \qquad \|UAT\|_2 = \|A\|_2.$$

Die zweite Gleichung folgt aus

$$\|UAT\|_2^2 = \varrho(T^tA^tU^tUAT) = \varrho(T^tA^tAT) = \varrho(A^tA) = \|A\|_2^2.$$

Für die anderen Normen gilt das nicht. Andererseits sind die Normen $\|\ \|_\infty$ und $\|\ \|_1$ leichter berechenbar, da eine Berechnung von Eigenwerten nicht nötig ist.

Fehlerfortpflanzung bei linearen Gleichungssystemen. Wir setzen stets voraus, daß $A \in \mathbb{R}^{n\times n}$ regulär ist und $b \in \mathbb{R}^n$. Das System $Ax = b$ hat also genau eine Lösung $x \in \mathbb{R}^n$. Zunächst betrachten wir nur Änderungen bei der rechten Seite von $Ax = b$. Hier geht das System $Ax = b$ in das System $Ax = b + \delta_b$ über, wobei $\delta_b \in \mathbb{R}^n$ der Eingabefehler ist. Dann gilt für die Lösung $(x + \delta_x) \in \mathbb{R}^n$ des „gestörten Systems“

$$A(x + \delta_x) = b + \delta_b, \qquad A(\delta_x) = \delta_b, \qquad \delta_x = A^{-1}\delta_b.$$

Man kann nun mit einer beliebigen Norm $\|\ \|$ und der zugehörigen Operatornorm $\|\ \|$ abschätzen

$$\|\delta_x\| \le \|A^{-1}\| \cdot \|\delta_b\|.$$

Wegen der Definition der Operatornorm kann bei dieser Abschätzung Gleichheit eintreten. Damit folgt

$$K_{\text{rel}}(b) = \lim_{\varepsilon\to 0} \sup_{0<\|\delta_b\|<\varepsilon} \frac{\|\delta_x\|}{\|x\|} : \frac{\|\delta_b\|}{\|b\|} = \|A^{-1}\| \frac{\|b\|}{\|A^{-1}b\|}.$$

Andererseits gilt aber auch

$$\|b\| \le \|A\| \cdot \|x\|$$

und auch hier tritt Gleichheit (bei einem gewissen $b \neq 0$) ein. Damit folgt

$$\sup_{b\neq 0} K_{\text{rel}}(b) = \|A^{-1}\| \cdot \|A\|.$$

Jetzt betrachten wir kleine Änderungen bzw. Fehler bei A. Beachte zunächst, daß die Inversenbildung stetig ist und deswegen

$$\lim_{\|\delta_A\|\to 0} \|(A + \delta_A)^{-1}\| = \|A^{-1}\|$$

gilt. In Analogie zu oben hat man

$$(A+\delta_A)(x+\delta_x)=b,$$

$$\delta_A(x+\delta_x)=-A(\delta_x),$$

$$\delta_x=(A+\delta_A)^{-1}(-(\delta_A)x)$$

und daher

$$\|\delta_x\|\le\|(A+\delta_A)^{-1}\|\cdot\|\delta_A\|\cdot\|x\|.$$

Diese Überlegungen gelten natürlich nur unter der Voraussetzung, daß δ_A so klein ist, daß die Matrix $(A+\delta_A)$ noch invertierbar ist. Man kann sich überlegen, daß man die letzte Ungleichung nicht verbessern kann. Damit erhält man für $\delta_A\to 0$

$$K_{\mathrm{rel}}(A)=\lim_{\varepsilon\to 0}\sup_{0<\|\delta_A\|<\varepsilon}\frac{\|\delta_x\|}{\|x\|}:\frac{\|\delta_A\|}{\|A\|}=\|A^{-1}\|\cdot\|A\|.$$

Wegen dieser Aussagen nennt man $\mathrm{cond}(A)=\|A^{-1}\|\cdot\|A\|$ die Konditionszahl von A. Es gilt

$$\|x\|=\|A^{-1}Ax\|\le\|A^{-1}\|\cdot\|A\|\cdot\|x\|\le\mathrm{cond}(A)\cdot\|x\|$$

und daher $\mathrm{cond}(A)\ge 1$.

Durch Zusammensetzen der bisher gewonnenen Ergebnisse erhält man das folgende Ergebnis.

Satz 37 (Fehlerfortpflanzung bei linearen Gleichungssystemen)**.**

$$\lim_{\|\delta_b\|,\ \|\delta_A\|\to 0}\frac{\|\delta_x\|}{\|x\|}:\left(\frac{\|\delta_b\|}{\|b\|}+\frac{\|\delta_A\|}{\|A\|}\right)\le\mathrm{cond}(A).$$

Beispiel: Hilbertmatrix. Sei $H_{10}\in\mathbb{R}^{10\times 10}$ die Hilbertmatrix. Dann ist

$$\mathrm{cond}_\infty(H_{10})=35.357.439.251.992,$$

wobei hier die $\|\ \|_\infty$-Norm zugrundegelegt wird. Die Konditionszahl ist also sehr groß. Dies erklärt die großen Fehler bei der Aufgabe 5.4.

5.3 Orthogonalisierungsverfahren

Sei $\{x_1,\dots,x_n\}\subset\mathbb{R}^m$ linear unabhängig, also insbesondere $n\le m$. Dann ist die Schmidt'sche Folge $v_1,\dots,v_n$ definiert durch

$$v_k=\|x_k-\sum_{i<k}(x_k,v_i)\cdot v_i\|_2^{-1}\cdot(x_k-\sum_{i<k}(x_k,v_i)\cdot v_i).$$

Das folgende Ergebnis ist aus der linearen Algebra bekannt.

Satz 38 (Schmidt'sches Orthonormierungsverfahren)**.** *Die Folge $v_1, \ldots, v_n$ ist eine ON-Folge und es gilt*

$$\langle \{v_1, \ldots, v_k\} \rangle = \langle \{x_1, \ldots, x_k\} \rangle$$

für alle $k = 1, \ldots, n$.

Jetzt nehmen wir an, daß die Vektoren $x_1, \ldots, x_n \in \mathbb{R}^m$ die n Spalten einer Matrix $A \in \mathbb{R}^{m \times n}$ mit m Zeilen bilden. Die Skalarprodukte, die beim ON-Verfahren berechnet werden, merken wir uns in einer Matrix T. Wir erhalten also den folgenden Algorithmus:

for $k = 1, \ldots, n$ do
for $i = 1, \ldots, k-1$ do
$t_{ik} = (A_k, B_i)$
end
$C_k = A_k - \sum_{i<k} t_{ik} B_i$
$t_{kk} = \|C_k\|_2$
$B_k = t_{kk}^{-1} \cdot C_k$
end

Setze dann noch $t_{ik} = 0$ für $i > k$.

Satz 39 (Schmidt-Verfahren als Matrixzerlegung)**.** *Wendet man diesen Algorithmus auf eine Matrix $A \in \mathbb{R}^{m \times n}$ mit Rang n an, so erhält man eine Faktorisierung*

$$A = B \cdot T.$$

Hierbei ist B eine $m \times n$-Matrix mit orthonormalen Spalten $B_1, \ldots, B_n$ und $T \in \mathbb{R}^{n \times n}$ ist eine obere Dreiecksmatrix mit positiven Diagonalelementen.

Bemerkungen. Ergänzt man die Spalten B_i zu einer ON-Basis $B_1, \ldots, B_m$ des $\mathbb{R}^m$, so erhält man eine sog. QR-Zerlegung

$$A = B \cdot T = Q \cdot R.$$

Dabei sei $Q \in \mathbb{R}^{m \times m}$ die Matrix mit den Spalten B_i und $R \in \mathbb{R}^{m \times n}$ sei die Matrix, die aus T entsteht, wenn man unten $(m-n)$ Reihen hinzufügt, die ganz aus Nullen bestehen. Das Schmidt-Verfahren führt also zu einer QR-Zerlegung. Da es aber numerisch sehr unstabil ist, wird es nicht verwendet. Stattdessen benutzt man das Verfahren von Householder.

Ein Gleichungssystem $Ax = b$ ist äquivalent zu $QRx = b$ oder $Rx = Q^t b$. Daher ist die Kenntnis einer QR-Zerlegung nützlich zum Lösen von Gleichungssystemen. Ist $Ax = b$ ein sog. überbestimmtes Gleichungssystem mit $A \in \mathbb{R}^{m \times n}$ und $m > n$, so existiert i.a. keine Lösung. Die „Lösung im Sinne der kleinsten (Fehler-) Quadrate“ x ist dann gegeben durch

$$\|Ax - b\|_2 = \min!$$

Dieses Optimierungsproblem nennt man auch (lineares) *Ausgleichsproblem*. Wir setzen voraus, daß der Rang von A gleich n sei. Weiter seien $A = QR = BT$ Zerlegungen wie oben. Wir setzen $Q^t b = (c, d)^t$ mit $c \in \mathbb{R}^n$ und $d \in \mathbb{R}^{m-n}$. Dann gilt

$$\|Ax - b\|_2^2 = \|Q^t Ax - Q^t b\|_2^2 = \|Rx - Q^t b\|_2^2 = \|Tx - c\|_2^2 + \|d\|_2^2.$$

Daher gilt für die eindeutige Lösung des *Ausgleichsproblems*

$$Tx = c.$$

Dabei ist T die Matrix, die ensteht, wenn man bei der Matrix R von $A = QR$ die unteren $(m - n)$ Zeilen streicht. Also sind auch Ausgleichsprobleme leicht zu lösen, wenn eine QR-Zerlegung von A bekannt ist.

Beim Gauß-Verfahren werden Dreiecksmatrizen L von links an A heranmultipliziert, es gilt

$$\mathrm{cond}(LA) \leq \mathrm{cond}(L) \cdot \mathrm{cond}(A)$$

und man muß damit rechnen, daß die Kondition von LA schlechter ist als die von A. Da dies zu einem unstabilen Verfahren führen kann, möchte man nur solche Transformationen verwenden, die die Kondition nicht schlechter machen. Dies leisten orthogonale Matrizen. Ist Q orthogonal, so gilt

$$\mathrm{cond}_2(QA) = \mathrm{cond}_2(Q) \cdot \mathrm{cond}_2(A).$$

Es gilt $\mathrm{cond}_2(Q) = 1$ und daher stimmen die Konditionszahlen von QA und von A überein. Beim Householder-Verfahren wird A durch Linksmultiplikation mit orthogonalen Matrizen auf Dreiecksgestalt gebracht. Die genaue Beschreibung erfolgt in der Vorlesung und kann in jedem Numerik-Buch nachgelesen werden.

Satz 40 (Householder-Verfahren zur QR-Faktorisierung)**.** *Wendet man das Verfahren von Householder auf eine beliebige Matrix $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ an, so erhält man eine QR-Faktorisierung*

$$A = QR.$$

Hierbei ist $Q \in \mathbb{R}^{m \times m}$ eine orthogonale Matrix und $R \in \mathbb{R}^{m \times n}$ ist eine obere (oder rechte) Dreiecksmatrix.

Bemerkung. Bei der Tikhonov-Regularisierung will man (im Fall von Gleichungssystemen) ein Problem der Gestalt

$$\|Ax - b\|_2^2 + \alpha \|Bx - z\|_2^2 = \min!$$

lösen. Hierbei ist $Ax = b$ das Gleichungssystem mit der evtl. ungenauen rechten Seite und $\|Bx - z\| \leq E$ die zusätzliche a priori Information, siehe Abschnitt 3.4. Besonders häufig wird der einfachste Fall

$$\|Ax - b\|_2^2 + \alpha \|x\|_2^2 = \min!$$

betrachtet. Dabei seien die Matrizen A und B quadratisch. Diese Probleme lassen sich leicht in der Form

$$\|\widetilde{A}x - \widetilde{b}\|_2^2 = \min!$$

mit $\widetilde{A} \in \mathbb{R}^{2n \times n}$ schreiben und können daher ebenfalls mit dem Householder-Verfahren gelöst werden.

5.4 Lineare Ausgleichsprobleme

Der folgende Satz charakterisiert die Lösung eines linearen Ausgleichsproblems durch die sog. *Normalgleichung*. Man sollte allerdings die Normalgleichung nicht benutzen, um Ausgleichsprobleme zu lösen, da die Kondition von A^tA häufig sehr schlecht ist, siehe Aufgabe 5.5. Wir haben bereits gesehen, daß man das Ausgleichsproblem numerisch mit dem Householder-Verfahren behandeln kann.

Satz 41 (Satz zur Normalgleichung)**.** *Sei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ mit beliebigen $m, n \in \mathbb{N}$. Dann ist $x \in \mathbb{R}^n$ Lösung von*

$$\|Ax - b\|_2 = \min!$$

genau dann, wenn

$$A^tAx = A^tb.$$

Diese sog. Normalgleichung hat stets mindestens eine Lösung x^*. Das Residuum $r = b - Ax^*$ ist eindeutig bestimmt und es gilt

$$A^tr = 0.$$

Wir diskutieren die Lösung von linearen Gleichungssystemen und Ausgleichsproblemen bei schlecht konditionierter Matrix A und (Rundungs-) Fehlern bei der rechten Seite. Wir betrachten drei Fälle:

1. Fall: $A \in \mathbb{R}^{n \times n}$ ist symmetrisch und regulär,
2. Fall: $A \in \mathbb{R}^{n \times n}$ ist regulär,
3. Fall: $A \in \mathbb{R}^{m \times n}$ ist beliebig.

1. Fall: $A \in \mathbb{R}^{n \times n}$ ist symmetrisch und regulär. Dann existiert eine ON-Basis $\{e_1, \ldots, e_n\}$ des $\mathbb{R}^n$ aus Eigenvektoren von A,

$$Ae_i = \lambda_i e_i.$$

Da A regulär ist, gilt $\lambda_i \neq 0$ für alle i. Also gilt für jedes $x \in \mathbb{R}^n$

$$Ax = \sum_{i=1}^{n} \lambda_i \, (x, e_i) e_i.$$

Für die Lösung von $Ax = b$ erhält man damit

$$x = \sum_{i=1}^{n} \frac{1}{\lambda_i} \, (b, e_i) e_i.$$

Wir wollen die Auswirkung von kleinen Fehlern bei b diskutieren und nehmen der Einfachheit an, daß

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0.$$

Die Hilbert-Matrix ist z.B. positiv definit und es gilt für $n = 10$ ungefähr

$$\lambda_1 = 1.75 \qquad \text{und} \qquad \lambda_{10} = 1.86 \cdot 10^{-13}.$$

Sei $\|\widetilde{b} - b\|_2 \leq \varepsilon$. Dann folgt

$$\|\widetilde{x} - x\|_2 \leq \frac{1}{\lambda_n} \cdot \|\widetilde{b} - b\|_2$$

und diese Abschätzung läßt sich nicht verbessern, das heißt es gilt

$$K_{\text{abs}} = \frac{1}{\lambda_n}.$$

Ähnlich erhält man

$$K_{\text{rel}} = \text{cond}_2(A) = \frac{\lambda_1}{\lambda_n}.$$

Man muß aber im Fall $\|\widetilde{b} - b\|_2 \leq \varepsilon$ nicht mit beliebigen Fehlern mit Norm kleiner als $\varepsilon \cdot K_{\text{abs}}$ rechnen, aus $\|\widetilde{b} - b\|_2 \leq \varepsilon$ folgt nämlich $|(\widetilde{b}, e_i) - (b, e_i)| \leq \varepsilon$ und damit

$$|(\widetilde{x}, e_i) - (x, e_i)| \leq \frac{\varepsilon}{\lambda_i}.$$

Manche Koordinaten lassen sich also viel genauer berechnen als andere, bei der i-ten Koordinate (Richtung von e_i) muß man mit einem Fehler der Größenordnung ε/λ_i rechnen. Die Idee der Regularisierung besteht darin, daß man statt der Formel

$$x = \sum_{i=1}^{n} \frac{1}{\lambda_i} (b, e_i) e_i,$$

die zu großen Rundungsfehlern führt, eine „Ersatzformel“ benutzt. Man kann etwa Koordinaten ignorieren, bei denen λ_i sehr klein ist. In Abhängigkeit von $\alpha > 0$ kann man also die „regularisierte Lösung“

$$x_\alpha = \sum_{|\lambda_i| > \alpha} \frac{1}{\lambda_i} (b, e_i) e_i$$

betrachten. Man spricht vom Abschneiden kleiner Eigenwerte. Man nimmt also einen Verfahrensfehler $\|x - x_\alpha\|$ in Kauf, kann dafür aber x_α ohne große Rundungsfehler berechnen. Diese Methode ist empfehlenswert, kann aber nur angewendet werden, wenn die λ_i und die e_i bekannt sind.

Bei der Tikhonov-Regularisierung wählt man ebenfalls einen Parameter $\alpha > 0$ und betrachtet die eindeutig bestimmte Minimalstelle x_α des Funktionals

$$\|Ax - b\|_2^2 + \alpha \|x\|_2^2 = \min!$$

Dieses x_α ist auch Lösung der Normalgleichung

$$(A^tA + \alpha I)x = A^tb.$$

Mit der ON-Basis aus Eigenvektoren erhält man

$$x_\alpha = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda_i^2 + \alpha} (b, e_i)e_i.$$

Daher hat die Tikhonov-Regularisierung ähnliche Eigenschaften wie das Abschneiden der kleinen Eigenwerte: Ist λ_i^2 groß im Vergleich zu α, so ist (x_α, e_i) ungefähr gleich (x, e_i). Ist dagegen λ_i^2 sehr klein im Vergleich zu α, so gilt $(x_\alpha, e_i) \approx 0$. Wir haben bereits gesehen, daß sich die Tikhonov-Regularisierung mit dem Householder-Verfahren realisieren läßt. Insbesondere muß man keine Eigenwerte kennen.

Bemerkung. Bei beiden Regularisierungsmethoden werden solche $x \in \mathbb{R}^n$ bevorzugt, die eine kleine Norm haben. Diese Methoden passen also zu einer a priori Information vom Typ $\|x\|_2 \leq E$. Wir wissen bereits, wie eine allgemeinere Information vom Typ $\|Bx - z\| \leq E$ berücksichtigt werden kann. Wie in Abschnitt 3.4 beschrieben, betrachtet man eine allgemeinere Tikhonov-Regularisierung und minimiert das Funktional

$$\|Ax - b\|_2^2 + \alpha\|Bx - z\|_2^2 = \min!$$

2. Fall: $A \in \mathbb{R}^{n\times n}$ ist regulär. Im ersten Fall war wesentlich, daß eine ON-Basis aus Eigenvektoren von A existiert. Jetzt könnte man versuchen, die Matrix zu diagonalisieren, d.h. zu einer Zerlegung

$$A = S^{-1}DS$$

überzugehen. Hier seien $A, S, D \in \mathbb{R}^{n\times n}$ mit einer regulären Matrix S und einer Diagonalmatrix D. Diese Zerlegung ist für Fragen der Kondition und Regularisierung jedoch nicht relevant – beim ersten Fall war wesentlich, daß

$$A = Q^tDQ$$

mit einer orthogonalen Matrix Q gilt, da orthogonale Matrizen längentreu sind und die Kondition nicht verändern. Deshalb suchen wir nach einer Zerlegung vom Typ

$$A = PDQ,$$

wobei sowohl P als auch Q orthogonal sind. Da der Fall $A \in \mathbb{R}^{n\times n}$ nicht wesentlich einfacher ist als der allgemeine Fall $A \in \mathbb{R}^{m\times n}$, betrachten wir gleich den allgemeineren Fall.

3. Fall: $A \in \mathbb{R}^{m\times n}$ ist beliebig. Die Singulärwertzerlegung hat viele Anwendungen, besonders im Zusammenhang mit Kondition, Regularisierung und Ausgleichsproblemen.

Satz 42 (Existenz der Singulärwertzerlegung)**.** *Sei $A \in \mathbb{R}^{m\times n}$ beliebig mit beliebigen $m, n \in \mathbb{N}$. Dann existieren orthogonale Matrizen $P \in \mathbb{R}^{m\times m}$ und $Q \in \mathbb{R}^{n\times n}$, so daß*

$$A = PDQ.$$

Hierbei ist $D \in \mathbb{R}^{m\times n}$ eine Diagonalmatrix mit nichtnegativen Diagonalelementen

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq 0.$$

Die Matrix D ist eindeutig bestimmt. Die Quadrate σ_i^2 der sog. singulären Werte σ_i stimmen überein mit den Eigenwerten von A^tA.

Ist $D \in \mathbb{R}^{m\times n}$ eine Diagonalmatrix wie in Satz 42 mit

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

und $\sigma_i = 0$ für alle anderen i, so definiert man die sog. Pseudoinverse D^+ von D als die Diagonalmatrix $D^+ \in \mathbb{R}^{n\times m}$ mit den nichtverschwindenden Diagonalelementen $\sigma_1^{-1}, \ldots, \sigma_r^{-1}$. Ist $A \in \mathbb{R}^{m\times n}$ eine beliebige Matrix und $A = PDQ$ eine Zerlegung wie in Satz 42, so definiert man die Pseudoinverse von A durch

$$A^+ = Q^tD^+P^t.$$

Es gilt: Die Minimallösung von $Ax = b$ (im Sinne von Aufgabe 5.9) ist gegeben durch $x = A^+b$. Mit diesem Thema (Stichwort: Lösung von unter- bzw. überbestimmten Gleichungen) beschäftigt sich auch die Aufgabe 5.9. Die Pseudoinverse heißt auch verallgemeinerte Inverse.

Bemerkung über Konditionszahlen. Ist $A \in \mathbb{R}^{m\times n}$ beliebig mit $A \neq 0$ und den positiven singulären Werten $\sigma_1 \geq \cdots \geq \sigma_r > 0$, so heißt

$$\mathrm{cond}_2(A) = \frac{\sigma_1}{\sigma_r}$$

die Konditionszahl von A bezüglich der Spektralnorm. Für die Nullmatrix $A = 0$ definiert man $\mathrm{cond}_2(A) = 0$. Die Formeln

$$\mathrm{cond}_2(A) = \|A\|_2 \cdot \|A^+\|_2$$

und

$$\mathrm{cond}_2(A^tA) = \mathrm{cond}_2(A)^2$$

gelten dann allgemein, sie hatten bisher nur für reguläre Matrizen einen Sinn. Allgemein gilt auch

$$\|A\|_2 = \sigma_1 \qquad \text{und für } A \neq 0 \qquad \|A^+\|_2 = \frac{1}{\sigma_r}.$$

Ist speziell $A \in \mathbb{R}^{n\times n}$ regulär, so erhält man natürlich $r = n$ und

$$\mathrm{cond}_2(A) = \frac{\sigma_1}{\sigma_n}.$$

Insbesondere stimmt in diesem Fall die neue Definition mit der alten überein. Mit Hilfe der Singulärwertzerlegung kann man jetzt leicht die besprochenen Regularisierungsmethoden verallgemeinern: aus dem „Abschneiden kleiner Eigenwerte“ wird jetzt das „Abschneiden kleiner singulärer Werte“, d.h. man definiert

$$x_\alpha = A_\alpha^+ b = Q^t D_\alpha^+ P^t b,$$

wobei für die Einträge d_i der Diagonalmatrix D_α^+ gilt: ist $\sigma_i > \alpha$ so $d_i = \sigma_i^{-1}$, ist $\sigma_i \leq \alpha$, so gilt $d_i = 0$.

Auch bei der Tikhonov-Regularisierung muß man die Eigenwerte (im symmetrischen Fall) ersetzen durch die singulären Werte und man erhält ebenfalls eine Darstellung der Form

$$x_\alpha = A_\alpha^+ b = Q^t D_\alpha^+ P^t b,$$

wobei jetzt die Diagonalelemente d_i von D_α^+ gegeben sind durch

$$d_i = \frac{\sigma_i}{\sigma_i^2 + \alpha}.$$

Das sich ergebende x_α ist dann wieder Minimalstelle vom Funktional

$$\|Ax - b\|_2^2 + \alpha \|x\|_2^2 = \min!$$

5.5 Iterative Verfahren

Iterative Verfahren sind sehr wichtig, insbesondere bei großen dünnbesetzten Gleichungssystemen, die häufig bei der Lösung von partiellen Differentialgleichungen auftreten. Man nimmt einen Verfahrensfehler in Kauf (wie ja schon bei den Regularisierungsverfahren), um schnell brauchbare Näherungen der Lösung zu erhalten.

Gegeben sei das lineare Gleichungssystem

$$Ax = b$$

mit einer regulären Matrix $A \in \mathbb{R}^{n \cdot n}$ und $b \in \mathbb{R}^n$. Verschiedene *Iterationsverfahren* beruhen auf einer Zerlegung von A in der Form $A = B + (A - B)$ mit einer regulären Matrix B. Dabei soll B so beschaffen sein, daß sich Gleichungssysteme mit der Koeffizientenmatrix B leicht lösen lassen. Das System $Ax = b$ ist äquivalent zu $x = (I - B^{-1}A)x + B^{-1}b$ und es liegt nahe, das Iterationsverfahren

$$x_{k+1} = (I - B^{-1}A)x_k + B^{-1}b \quad \text{bzw.} \quad Bx_{k+1} = (B - A)x_k + b$$

zu betrachten. Die Matrix $(I - B^{-1}A)$ heißt *Iterationsmatrix*. Man erhält x_{k+1} durch Lösen eines Gleichungssystems mit der Koeffizientenmatrix B und der rechten Seite $(B - A)x_k + b$. Der folgende Satz beantwortet die Frage, für welche Iterationsmatrizen das Verfahren konvergiert.

Satz 43 (Konvergenz von Iterationsverfahren bei linearen Gleichungssystemen)**.** *Gegeben sei das lineare Gleichungssystem $Ax = b$ mit regulärer Matrix $A \in \mathbb{R}^{n\cdot n}$. Das Iterationsverfahren*

$$Bx_{k+1} = (B - A)x_k + b$$

(mit regulärem $B \in \mathbb{R}^{n\cdot n}$) konvergiert genau dann für jeden Startvektor $x_0 \in \mathbb{R}^n$ gegen die Lösung x^* von $Ax = b$, wenn der Spektralradius der Iterationsmatrix kleiner ist als 1,

$$\varrho(I - B^{-1}A) < 1.$$

Mit Hilfe dieses Satzes kann man die gängigen Verfahren analysieren: Gesamtschrittverfahren, Einzelschrittverfahren.

5.6 Eigenwertaufgaben

In vielen Anwendungen interessiert man sich für die Eigenwerte einer Matrix $A \in \mathbb{R}^{n\times n}$. Besonders wichtig ist der Fall symmetrischer Matrizen. Wir besprechen das Jacobi-Verfahren zur Berechnung aller Eigenwerte einer symmetrischen Matrix A. Dazu ist es hilfreich, die sog. *Frobenius-Norm* von A zu betrachten, die definiert ist durch

$$\|A\|_F = \left(\sum_{i,j=1}^{n} a_{ij}^2\right)^{1/2}.$$

Zunächst zeigen wir (evtl. als Hausaufgabe, siehe Aufgabe 5.12) das folgende Lemma.

Lemma. *Ist $A = A^t \in \mathbb{R}^{n\cdot n}$ eine symmetrische Matrix und $Q \in \mathbb{R}^{n\cdot n}$ orthogonal, so gilt*

$$\|Q^t AQ\|_F = \|A\|_F.$$

Sind $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ die Eigenwerte von A, so gilt insbesondere

$$\sum_{i=1}^{n} \lambda_i^2 = \sum_{i,j=1}^{n} a_{ij}^2.$$

Wir definieren nun

$$N(A) = \sum_{i\neq j} a_{ij}^2$$

und erhalten

$$\sum_{i=1}^{n} \lambda_i^2 = \sum_{i=1}^{n} a_{ii}^2 + N(A).$$

Durch geeignete orthogonale Transformationen

$$A \mapsto Q^t AQ$$

versucht man, die Größe $N(A)$ zu verkleinern, die Eigenwerte bleiben dabei unverändert. Hat man schließlich ein Q gefunden mit

1. die Matrix Q^tAQ ist „fastdiagonal“ im Sinne von $N(Q^tAQ) < \varepsilon$;

2. die Diagonalelemente von Q^tAQ sind $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_n$,

so gilt:
Die $\widetilde{\lambda}_i$ sind gute Approximationen der Eigenwerte λ_i von A, es gilt (nach einer evtl. nötigen Umsortierung der Indizes)

$$|\lambda_i - \widetilde{\lambda}_i| \leq \sqrt{\varepsilon}$$

für alle i. Dies folgt aus dem folgenden Lemma.

Lemma. *Sind A und B zwei symmetrische Matrizen in $\mathbb{R}^{n\cdot n}$ mit Eigenwerten*

$$\lambda_1(A) \geq \cdots \geq \lambda_n(A) \quad \textit{und} \quad \lambda_1(B) \geq \cdots \geq \lambda_n(B),$$

so gilt für alle i

$$|\lambda_i(A) - \lambda_i(B)| \leq \|A - B\|.$$

Hierbei ist $\|\cdot\|$ eine beliebige Operatornorm für Matrizen.

Um $N(A)$ zu verkleinern, wählt man ein $a_{ij} \neq 0$ mit $i \neq j$ und eine orthogonale Transformation in der durch e_i und e_j aufgespannten Ebene, die a_{ij} in 0 überführt. Setzt man diese Transformation als Drehung im $\mathbb{R}^2$ um den Winkel α an, so liefert die Ähnlichkeitstransformation

$$\begin{pmatrix} b_{ii} & b_{ij} \\ b_{ij} & b_{jj} \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{pmatrix} \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

eine Diagonalmatrix, wenn

$$0 = b_{ij} = a_{ij}(\cos^2\alpha - \sin^2\alpha) + (a_{jj} - a_{ii})\cos\alpha\sin\alpha$$

gilt. Mit Hilfe der Additionstheoreme kann man diese Bedingung umformen in

$$b_{ij} = a_{ij}\cos(2\alpha) + \frac{1}{2}(a_{jj} - a_{ii})\sin(2\alpha)$$

beziehungsweise

$$\cot(2\alpha) = \frac{a_{ii} - a_{jj}}{2a_{ij}}.$$

Im Fall $n > 2$ erhält man eine Transformationsmatrix $G_{ij}(\alpha)$, die nur an 4 Stellen von der Einheitsmatrix verschieden ist:

$$G_{ii}(\alpha) = G_{jj}(\alpha) = \cos\alpha \quad \text{und} \quad G_{ij}(\alpha) = -\sin\alpha, \quad G_{ji}(\alpha) = \sin\alpha.$$

Solche Matrizen heißen *Jacobi-Transformation* oder *Givens-Rotation.*

Lemma. *Bildet man, wie beschrieben,*

$$B = G_{ij}(\alpha) A G_{ij}(\alpha),$$

so gilt

$$b_{ij} = b_{ji} = 0 \qquad \textit{und} \qquad N(B) = N(A) - 2|a_{ij}|^2.$$

Iteriert man dieses Verfahren, so erhält man das klassische Jacobi-Verfahren und seine Konvergenz:

- Setze $A^{(1)} = A$ und führe dann für $m = 1, 2, \ldots$ folgende Schritte aus:
- Bestimme $i \neq j$ mit $|a_{ij}^{(m)}| = \max_{\ell \neq k} |a_{\ell k}^{(m)}|$ und setze $G = G_{ij}(\alpha_i)$ mit dem passenden α_i.
- Setze $A^{(m+1)} = G A^{(m)} G^t$.

Es gilt dann

$$N(A^{(m+1)}) \leq \left(1 - \frac{2}{n(n-1)}\right) N(A^{(m)}$$

und insgesamt erhalten wir:

Satz 44 (Jacobi-Verfahren zur Eigenwertbestimmung)**.** *Es gilt*

$$N(A^{(m)}) \leq \left(1 - \frac{2}{n(n-1)}\right)^{m-1} N(A)$$

und damit

$$|\lambda_i - \lambda_i^{(m)}| \leq \left(1 - \frac{2}{n(n-1)}\right)^{(m-1)/2} N(A)^{1/2}$$

für alle i. Hier sind λ_i die (geeignet geordneten) Eigenwerte von A und $\lambda_i^{(m)}$ die Diagonalelemente von $A^{(m+1)}$.

Aufgaben

5.1. Gegeben sei die Matrix

$$A = \begin{pmatrix} 1 & \alpha \\ 1 & 1 \end{pmatrix}$$

mit $\alpha \in \mathbb{R}$, wobei $\alpha \neq 1$.
a) Konstruieren Sie eine LU-Zerlegung von A.
b) Berechnen Sie $\mathrm{cond}_\infty(A)$.

5.2. Zeigen Sie, daß die Hilbertmatrix $H_n \in \mathbb{R}^{n\times n}$, gegeben durch

$$h_{ij} = \frac{1}{i+j-1}$$

positiv-definit (insbesondere also regulär) ist. Berechnen Sie dazu das Integral

$$\int_0^1 \left(\sum_{i=1}^{n} x_i t^{i-1}\right)^2 dt.$$

5.3. Beweisen Sie:
a) Ist U eine invertierbare obere Dreiecksmatrix, so auch U^{-1}.
b) Gilt zusätzlich $u_{kk} = 1$ für alle k, so gilt das entsprechende auch für U^{-1}.
c) Das Produkt von zwei oberen Dreiecksmatrizen ist wieder eine.

5.4. Programmieren Sie ein einfaches Eliminationsverfahren für ein lineares Gleichungssystem

$$Ax = b.$$

Hierbei sei $A \in \mathbb{R}^{n\times n}$ und $b \in \mathbb{R}^n$.

Testen Sie Ihr Programm (für verschiedene n) anhand der Hilbertmatrix

$$a_{ij} = \frac{1}{i+j-1}$$

mit

$$b_i = \frac{1}{i} + \frac{1}{i+1} + \cdots + \frac{1}{i+n-1}.$$

Bemerkung: Die Hilbertmatrix ist sehr schlecht konditioniert, daher ist schon bei relativ kleinen n mit großen Rundungsfehlern (evtl. auch Division durch Null) zu rechnen.

Zum Vergleich: für die exakte Lösung $x \in \mathbb{R}^n$ gilt natürlich $x_1 = \cdots = x_n = 1$.

5.5. Es sei $A \in \mathbb{R}^{n\times n}$ regulär.
a) Beweisen Sie die Gleichung

$$\mathrm{cond}_2(A^tA) = (\mathrm{cond}_2(A))^2.$$

Dabei sei cond_2 die Konditionszahl bezüglich der Spektralnorm.
b) Sei A zusätzlich symmetrisch. Zeigen Sie, daß sich die Konditionszahl $\mathrm{cond}_2(A)$ durch die Eigenwerte von A ausdrücken läßt.

5.6. Es seien $x, y \in \mathbb{R}^n$ mit $x \neq y$ und $\|x\|_2 = \|y\|_2$. Die Matrix $M \in \mathbb{R}^{n\times n}$ sei gegeben durch

$$M = I - vu^t$$

mit $v = x - y$ und $u = 2v/\|v\|_2^2$. Dabei sei I die Einheitsmatrix. Zeigen Sie:
a) M ist orthogonal mit $MM = I$.
b) Es gilt $Mx = y$ und $My = x$.

5.7. Programmieren Sie das Householder-Verfahren zur QR-Zerlegung einer $(m \times n)$-Matrix, wobei $m \geq n$. Lösen Sie damit nocheinmal das Gleichungssystem (mit der Hilbert-Matrix) von Aufgabe 5.4. Dabei soll die Tikhonov-Regularisierung angewendet werden, gesucht ist also das $x \in \mathbb{R}^n$ mit

$$\|Ax - b\|_2^2 + \alpha \|x\|_2^2 = \min!$$

Wählen Sie beispielsweise $n = 20$ und $\alpha = 10^{-k}$ mit $k = 0, 1, 2, \ldots, 10$ und vergleichen Sie die Ergebnisse, auch mit denen, die Sie bei Aufgabe 5.4 erhalten haben.

5.8. Es sei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ und $\alpha > 0$. Weiter sei

$$F(x) = \|Ax - b\|_2^2 + \alpha \|x\|_2^2$$

für $x \in \mathbb{R}^n$. Zeigen Sie:

a) F hat genau eine Minimalstelle x^*.

b) x^* ist die (stets eindeutige) Lösung von

$$(A^t A + \alpha I)x = A^t b.$$

5.9. Es sei $A \in \mathbb{R}^{m \times n}$ eine beliebige Matrix mit $m, n \in \mathbb{N}$. Weiter sei $b \in \mathbb{R}^m$.

a) Zeigen Sie, daß das Ausgleichsproblem

$$\|Ax - b\|_2 = \min!$$

stets genau eine Lösung x_b^* mit minimaler (euklidischer) Norm hat.

b) Es sei

$$A = PDQ$$

eine Singulärwertzerlegung (siehe Satz 42). Beschreiben Sie x_b^* mit Hilfe dieser Matrix-Zerlegung.

c) Zeigen Sie, daß es genau eine Matrix $A^+ \in \mathbb{R}^{n \times m}$ gibt mit

$$A^+ b = x_b^*$$

für alle b. Beschreiben Sie die Matrix A^+ bei gegebener Zerlegung $A = PDQ$. Zeigen Sie, daß im Fall $m \geq n$ mit $\mathrm{Rang}(A) = n$

$$A^+ = (A^t A)^{-1} A^t$$

gilt.

5.10. Gegeben sei ein Gleichungssystem $Ax = b$ mit der Matrix

$$A = \begin{pmatrix} 1 & 0.1 \\ 6 & 1 \end{pmatrix}.$$

Zeigen Sie, daß sowohl das Gesamtschrittverfahren wie auch das Einzelschrittverfahren konvergieren. Wie groß ist jeweils der Spektralradius der Iterationsmatrix?

5.11. Gegeben sei $Ax = b$ mit einer symmetrischen und positiv definiten Matrix $A \in \mathbb{R}^{n\times n}$. Zeigen Sie, daß das Einzelschrittverfahren stets konvergiert.

Hinweis: Warum genügt es, den Fall $a_{ii} = 1$ (für alle i) zu betrachten?

5.12. Es sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und $T \in \mathbb{R}^{n\times n}$ sei orthogonal. Weiter seien $\lambda_1, \ldots, \lambda_n$ die Eigenwerte von A und es sei

$$\widetilde{A} = T^t A T.$$

Zeigen Sie, daß

$$\sum_{i=1}^{n} \lambda_i^2 = \sum_{i,j} a_{ij}^2 = \sum_{i,j} \widetilde{a}_{ij}^2$$

gilt.

Bemerkung: Diese Aussage ist der Ausgangspunkt für das Jacobi-Verfahren zur Bestimmung der Eigenwerte von A.

Liste der Sätze

1. Fehler des Bisektionsverfahrens
2. Lokaler Fehler des Newton-Verfahrens
3. Globale Konvergenz des Newton-Verfahrens
4. Globaler Fehler des Newton-Verfahrens
5. Fehler des Sekantenverfahrens
6. Banach'scher Fixpunktsatz
7. Lage der Nullstellen von Polynomen
8. Existenz und Eindeutigkeit des Interpolationspolynoms
9. Fehlerabschätzung bei der Interpolation durch Polynome
10. Formel für die Tschebyscheff-Polynome
11. Normabschätzung für normierte Polynome
12. Optimalität der Tschebyscheff-Knoten
13. Satz von Faber
14. Konvergenz der Interpolationspolynome bei Tschebyscheff-Knoten und Lipschitz-Funktionen
15. Eigenschaften der dividierten Differenzen
16. Lösbarkeit des Hermite'schen Interpolationsproblems
17. Fehler bei der Hermite-Interpolation
18. Hermite-Interpolation und dividierte Differenzen
19. Satz von Jackson
20. Optimalitätseigenschaft linearer Splines
21. Fehler bei der Interpolation durch lineare Splines
22. Natürlichen kubische Interpolationssplines
23. Natürliche Splines höherer Ordnung
24. Existenz des abstrakten Interpolationssplines
25. Optimalität des Spline-Algorithmus
26. Eigenschaften der Tikhonov-Regularisierung
27. Fehler von interpolatorischen Quadraturformeln
28. Satz von Peano

29. Anwendung des Satzes von Peano
30. Fehler einiger Quadraturformeln
31. Optimale Konvergenzordnung von Quadraturformeln
32. Eigenschaften der Gauß-Formel
33. Fehlerabschätzung für die Gauß-Formel
34. Existenz der LU-Zerlegung
35. Cholesky-Zerlegung
36. Beschreibung wichtiger Operatornormen
37. Fehlerfortpflanzung bei linearen Gleichungssystemen
38. Schmidt'sches Orthonormierungsverfahren
39. Schmidt-Verfahren als Matrixzerlegung
40. Householder-Verfahren zur QR-Faktorisierung
41. Satz zur Normalgleichung
42. Existenz der Singulärwertzerlegung
43. Konvergenz von iterativen Verfahren bei linearen Gleichungssystemen
44. Jacobi-Verfahren zur Eigenwertbestimmung

Literaturverzeichnis

[1] N. S. Bakhvalov (1959): On approximate computation of integrals. Vestnik MGU, Ser. Math. Mech. Astron. Phys. Chem. **4**, 3–18. [Russisch]

[2] L. Blum, M. Shub, S. Smale (1989): On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. Bull. of the AMS **21**, 1–46.

[3] L. Blum (1990): Lectures on a theory of computation and complexity over the reals (or an arbitrary ring). In: 1989 lectures in complex systems, 1–47, Santa Fe, Addison-Wesley, Redwood City.

[4] H. Brass (1977): Quadraturverfahren. Vandenhoeck und Ruprecht, Göttingen.

[5] J. C. P. Bus, T. J. Dekker (1975): Two efficient algorithms with guaranteed convergence for finding a zero of a function. ACM Trans. Math. Software **1**, 330–345.

[6] N. J. Cutland (1980): Computability. Cambridge Univ. Press, Cambridge.

[7] P. Deuflhard, A. Hohmann (2002): Numerische Mathematik 1. de Gruyter.

[8] J. Edmonds (1967): Systems of distinct representatives and linear algebra. J. Res. of the Nat. Bureau of Standards (B) **71**, 241–245.

[9] M. Hanke-Bourgeois (2009): Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens. Teubner Verlag, 3. Auflage.

[10] L. G. Khachiyan (1979): A polynomial algorithm in linear programming. Soviet Math. Doklady **20**, 191–194.

[11] D. Kincaid, W. Cheney: Numerical Analysis. Brooks/Cole. 3. Auflage 2002. (Nachdruck bei der American Math. Soc. erhältlich.)

[12] R. Kress: Numerical Analysis. Springer Verlag 1998.

[13] G. D. Maistrovskii (1972): On the optimality of Newton's method. Soviet Math. Dokl. **13**, 838–840.

[14] K. Miller (1970): Least squares methods for ill-posed problems with a prescribed bound. SIAM J. Math. Anal. **1**, 52–74.

[15] M. W. Müller (1978): Approximationstheorie. Akademische Verlagsgesellschaft, Wiesbaden.

[16] E. Novak (1988): Deterministic and Stochastic Error Bounds in Numerical Analysis. Lecture Notes in Mathematics **1349**, Springer.

[17] E. Novak (1995): The real number model in numerical analysis. J. Complexity **11**, 57–73.

[18] E. Novak (1996): On the power of adaption. J. Complexity **12**, 199–237.

[19] E. Novak (1999): Numerische Verfahren für Hochdimensionale Probleme und der Fluch der Dimension. Jber. d. Dt. Math.-Verein. **101**, 151–177.

[20] E. Novak (2000): Quantum complexity of integration. J. Complexity **16**, 2–16.

[21] E. Novak, K. Ritter (1993): Some complexity results for zero finding for univariate functions. J. Complexity **9**, 15–40.

[22] T. von Petersdorff (1993): A short proof for Romberg integration. Amer. Math. Monthly **100**, 783–785.

[23] L. Plaskota (1996): Noisy Information and Computational Complexity. Cambridge University Press.

[24] T. J. Rivlin (1969): An Introduction to the Approximation of Functions. Dover Publications, New York.

[25] H.-G. Roos, H. Schwetlick: Numerische Mathematik. Teubner Verlag 1999.

[26] R. Schaback, H. Wendland (2005): Numerische Mathematik. Springer-Verlag.

[27] G. Schmeißer, H. Schirmeier (1976): Praktische Mathematik. Walter de Gruyter Verlag.

[28] S. Smale (1990): Some remarks on the foundation of numerical analysis, SIAM Review **32**, 211–220

[29] J. Stoer (1994): Numerische Mathematik 1. Springer-Verlag.

[30] J. F. Traub, H. Woźniakowski (1991): Information-based complexity: new questions for mathematicians. Math. Intell. **13**, 34–43.

[31] J. F. Traub, G. W. Wasilkowski, H. Woźniakowski (1988): Information-Based Complexity. Academic Press.

[32] M. Vianello, R. Zanovello (1992): On the superlinear convergence of the secant method. Amer. Math. Monthly **99**, 758–761.

[33] J. Werner (1992): Numerische Mathematik I. Vieweg, Braunschweig.

Index