Numerische Mathematik für Informatiker

Manuel Kauers
nach einer Vorlesung von
Prof. Dr. W. NIETHAMMER

21. Oktober 2000

Inhaltsverzeichnis

1 Lösen von Linearen Gleichungssystemen

1.1 Lösen von Dreieckssystemen

Gegeben: $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ sowie die Gleichung $Ax = b$ (Lineares Gleichungssystem, LGS) . Gesucht ist ein $x \in \mathbb{R}^n$, für das die Gleichung $Ax = b$ erfüllt ist. Das LGS heißt *überbestimmt*, falls $m > n$ und *unterbestimmt*, falls $m < n$. Ein überbestimmtes System hat i.A. keine Lösung.

Sei nun speziell $R \in \mathbb{R}^{n \times n}$ obere Dreiecksmatrix ist, d.h. R hat die Form

$$R = \begin{pmatrix} r_{11} & \dots & \dots & r_{1n} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & r_{nn} \end{pmatrix}$$

Weiter gelte

$$\det R \left(= \prod_{i=1}^{n} r_{ij} \right) \neq 0$$

Dann erhält man eine Lösung des LGS durch „Rückwertsrechnen“:

$$\begin{aligned} x_n = \frac{1}{r_{nn}} b_n \quad &\Longrightarrow \quad x_{n-1} = \frac{1}{r_{(n-1),(n-1)}}(b_{n-1} - r_{n-1,n} x_n) \\ &\Longrightarrow \quad \dots \\ &\Longrightarrow \quad x_1 = \frac{1}{r_{11}}(b_1 - r_{12}x_2 - \dots - r_{1n}x_n) \end{aligned}$$

Komplexität: $\sim \frac{n^2}{2}$

1.2 Elimination nach Gauss

Prinzip: Im LGS $Ax = b, A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n, \det A \neq 0$ soll die Matrix A auf Dreiecksform gebracht werden. Schreibe $Ax = b$ in der Form $z_k^\top x = b_k$ ($k = 1, \dots, n$), wobei $z_k^\top$ die k-te Zeile der Matrix A ist. Es gelten folgende *elementere Zeilenumformungen* (EZU), mit denen die Matrix $(A|b)$ modifiziert werden kann, ohne den Lösungsvektor x zu ändern:

1. Vertauschen von zwei Zeilen
2. $z_k^\top$ wird ersetzt durch $\lambda_k z_k^\top x = \lambda_k b_k$
3. $z_k^\top$ wird ersetzt durch $z_k^\top x + \lambda z_k^\top x = b_k + \lambda b_l$

Der Gauss-Algorithmus besteht aus zwei Schritten:

Schritt A: Transformation von $(A|b)$ durch eine Folge von EZUen auf obere Dreiecksform

Schritt B: Rückwärtsrechnen nach 1.1

Man bezeichnet $a_{ii}^{(i)}$ als Pivotelement. Optimierung: Tausche $a_{ii}^{(i)}$ gegen das betragsmäßig größte Element derselben Spalte aus, um Rechenfehler zu minimieren (Spaltenpivotsuche). Komplexität des GAUSS-Algorithmus: $\sim \frac{n^3}{3}$

1.3 Berechnung der Determinante

Sei $A \in \mathbb{R}^{n \times n}$. Führe A durch EZUen in eine Dreicksmatrix R über. Für die Determinante von A gilt dann:

$$\det A = (-1)^v \cdot \prod_{i=1}^{n} a_{ii}^{(i)}$$

wobei v die Anzahl der Zeilenvertauschungen beim Übergang von A nach R bezeichnet.

1.4 Faktorisierung von A

Def. 1 Sei $A = ((a_{ij})) \in \mathbb{R}^{n \times n}$. Gibt es eine untere Dreiecksmatrix $L = ((l_{ij}))$ und eine obere Dreiecksmatrix $R = ((r_{ij}))$ $(L, R \in \mathbb{R}^{n \times n})$ mit $A = L \cdot R$, also

$$\begin{pmatrix} a_{11} & \dots & \dots & a_{1n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \dots & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ l_{n1} & \dots & \dots & l_{nn} \end{pmatrix} \begin{pmatrix} 1 & r_{12} & \dots & r_{1n} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_{n-1,n} \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

so heißt $L \cdot R$ *Faktorisierung* von A

Prinzip: a_{ik} ist Skalarprodukt von Zeile i in L mit Spalte k in R. Man berechnet abwechselnd eine Zeile von L und eine Spalte von R unter Verwendung der vorherigen Ergebnisse und beginnt mit $a_{11} = l_{11}$.

1.5 Verfahren von CRÔUT zur Lösung eines LGS

Schritt A: Faktorisiere $A = L \cdot R$ und definiere $y := Rx$

Schritt B: Löse $Ly = b$ nach 1.1

Schritt C: Löse $Rx = y$ nach 1.1

1.6 Invertierung

Schritt A: Faktorisiere $A = L \cdot R$

Schritt B: Berechne L^{-1} und R^{-1}. Dies sind Dreiecksmatrizen, die wegen $L \cdot L^{-1} = I$ als n lineare Gleichungssysteme aufgefaßt und so nach 1.1 berechnet werden können.

Schritt C: $A^{-1} = R^{-1} \cdot L^{-1}$

1.7 Verfahren von Cholesky

Es sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Dann gibt es eine Dreiecksmatrix L mit $A = L \cdot L^\top$. Die Matrix L kann mit dem Algorithmus aus Abbildung 1 mit einem Aufwand von $\frac{n^3}{6}$ bestimmt werden. Damit läßt ich ein LGS nach folgendem Schema lösen:

Schritt A: Berechne L mit $A = L \cdot L^\top$

Schritt B: Löse $Ly = b$

Schritt C: Löse $L^\top x = y$

1.8 Methode der kleinsten Quadrate

Eine Größe $s(t)$ wird an $t = t_1, \ldots, t_n$ zu $s(t_i) := s_i$ gemessen. Gesucht ist eine Parabel $\hat{s}(t)$ zweiter Ordnung mit

$$\sum_{j=1}^{m} (s_j - \hat{s}(t))^2 \leq \sum_{i=1}^{m} (s_i - s(t))^2$$

$$
\begin{array}{l}
\texttt{for (int k = 1; k <= n; k++) \{} \\
\quad \texttt{for (int i = 1; i < k; i++)} \\
\qquad \texttt{l[k][i] = } \dfrac{\texttt{1}}{\texttt{l[i][i]}} \texttt{ * } \left(\texttt{a[k][i] - } \displaystyle\sum_{j=1}^{i-1} \texttt{l[i][j] * l[k][j]} \right)\texttt{;} \\
\quad \texttt{l[k][k] = } \sqrt{\texttt{a[k][k] - } \displaystyle\sum_{j=1}^{k-1} \texttt{l[k][l]}^2}\texttt{;} \\
\texttt{\}}
\end{array}
$$

Abbildung 1: Algorithmus zur Bestimmung von L

Lösung: Seien $b \in \mathbb{R}^m, \quad A \in \mathbb{R}^{m\times 3}$ mit

$$b = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{pmatrix}, \qquad A = \begin{pmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ \vdots & \vdots & \vdots \\ 1 & t_m & t_m^2 \end{pmatrix}$$

Dann enthält der Lösungsvektor $\hat{x} \in \mathbb{R}^3$ des LGS $(A^\top A)\hat{x} = A^\top b$ die Koeffizienten des gesuchten Polynoms:

$$\hat{s}(t) = \hat{x}_1 + \hat{x}_2 t + \hat{x}_3 t^2$$

2 Lösen von Gleichungen

2.1 Banach-scher Fixpunktsatz (eindimensional)

Def. 2 Sei $f : D \to \mathbb{R}$ eine Funktion. $x_0 \in D$ heißt *Fixpunkt*, falls gilt $f(x_0) = x_0$.

Satz 1 Sei $I \subseteq \mathbb{R}$ ein abgeschlossenes Intervall und $f : I \to I$ eine Funktion mit

$$\exists \quad L < 1 \quad \forall \quad x, y \in \mathbb{R} : \qquad |f(x) - f(y)| \leq L|x - y| \tag{1}$$

Dann besitzt f in I genau einen Fixpunkt s und für die Vorschrift $(x_n)_{n=1}^{\infty}$ mit $x_0 \in I$ beliebig und

$$x_{n+1} = f(x_n) \qquad (n \geq 0) \tag{2}$$

gilt $\lim\limits_{n\to\infty} x_n = s$. Es gilt die Fehlerabschätzung

$$|x_m - s| \leq \frac{L^m}{1-L}|x_1 - x_0| \tag{3}$$

Bemerkungen:

- Eine Funktion f, die (1) genügt, heißt *stark kontrahierend.*
- (2) bezeichnet man als *Allgemeines Iterationsverfahren.*
- Ist f stetig differenzierbar auf I, und gilt

  $$\exists \quad 0 \leq L < 1 \quad \forall \quad x \in I : \qquad |f'(x)| \leq L$$

  so ist f stark kontrahierend auf I. (Dies ist i.A. leichter nachzuprüfen)

2.2 Konvergenzordnung

Def. 3 Ein Fixpunkt s einer Funktion $f : I \to I$ heißt *anziehend* $:\Longleftrightarrow$ Es gibt ein Intervall I mit $s \in I^0$ (I^0: Inneres von I), so daß für $x_0 \in I$ die Folge $(x_m)_{m=0}^{\infty}$ aus Gleichung (2) gegen s konvergiert

Def. 4 Das Verfahren (2) heißt konvergierend bezüglich s von (mindestens) der Ordnung k, wenn s anziehender Fixpunkt ist und es gilt

$$\forall \quad x_0 \in I \quad \exists \quad n_0 \in \mathbb{N} \quad \forall \quad m \geq n_0 : \qquad |x_{m+1} - s| \leq c|x_m - s|^k$$

Ist $k = 2$, so bezeichnet man das Verfahren als *quadratisch* konvergierend, ist $k = 1$ ($\Longrightarrow L < 1$), so heißt das Verfahren *linear* konvergierend.

Satz 2 Sei $f : I \to I$ zweimal stetig differenzierbar und s Fixpunkt von f mit $f'(s) = 0$. Dann konvergiert das Verfahren (2) quadratisch.

2.3 Newton-Verfahren

Das Newton-Verfahren ist eine Iterationsvorschrift zur Näherung einer Nullstelle. Ist φ eine differenzierbare Funktion mit einer Nullstelle s in einem Intervall I und gilt $\varphi'(x) \neq 0$ ($x \in I$), dann kann die Nullstelle s durch die Vorschrift

$$x_{m+1} = x_m - \frac{\varphi(x_m)}{\varphi'(x_m)} \qquad (m \in \mathbb{N}_0) \tag{4}$$

approximiert werden (Startwert $x_0 \in I$ beliebig). Existiert auf I keine Nullstelle s, so divergiert das Verfahren.

Satz 3 Sei $I = [a, b]$ ein Intervall, $\varphi \in C^3(I, \mathbb{R})$, $s \in I$ mit $\varphi(s) = 0$ und $\varphi'(x) \neq 0$ ($x \in I$). Dann konvergiert das Newton-Verfahren (4) quadratisch.

3 Lösen von Gleichungssystemen

3.1 Vektor- und Matrixnormen

Def. 5 Eine Abbildung $||\cdot|| : \mathbb{R}^n \to \mathbb{R}$ heißt *Vektornorm*, falls für alle $\lambda \in \mathbb{R}, x, y \in \mathbb{R}^n$ gilt:

1. $||x|| \geq 0, \qquad ||x|| = 0 \quad \Longleftrightarrow \quad x = 0$
2. $||\lambda x|| = |\lambda| \; ||x||$

3. $||x+y|| \leq ||x|| + ||y||$

Beispiele für Normen auf $\mathbb{R}^n$:

$$||x||_1 := \sum_{i=1}^{n} |x_i|, \quad ||x||_2 := \sqrt{\sum_{i=1}^{n} x_1^2}, \quad ||x||_\infty := \max_{1 \leq i \leq n} |x_i|,$$

$||\cdot||_1$ heißt *Betragssummennorm*, $||\cdot||_2$ heißt *Euklidnorm*, $||\cdot||_\infty$ heißt *Maximumsnorm*.

Def. 6 Eine Abbildung $||\cdot|| : \mathbb{R}^{n\times n} \to \mathbb{R}$ heißt *Matrixnorm*, falls für alle $A, B \in \mathbb{R}^{n\times n}, \lambda \in \mathbb{R}$ gilt:

1. $||A|| \geq 0, \qquad ||A|| = 0 \iff A = 0$
2. $||\lambda A|| = |\lambda| \; ||A||$
3. $||A+B|| \leq ||A|| + ||B||$
4. $||A \cdot B|| \leq ||A|| \cdot ||B||$

Def. 7 Eine Matrixnorm $||\cdot||_M$ heißt *passend* zu einer Vektornorm $||\cdot||_V$, falls für alle $A \in \mathbb{R}^{n\times n}, x \in \mathbb{R}^n$ gilt:

$$||Ax||_V \leq ||A||_M \cdot ||x||_V$$

So ist z.B. die Spaltensummennorm $||\cdot||_1$ passend zur Betragssummennorm $||\cdot||_1$, wobei

$$||A||_1 := \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}|$$

3.2 Der Fixpunktsatz im $\mathbb{R}^n$

Satz 4 Sei $D \subseteq \mathbb{R}^n$ abgeschlossen, $||\cdot||$ eine Vektornorm, $f : D \to D$ eine Funktion mit

$$\exists \quad L < 1 \quad \forall \quad x, y \in D: \qquad ||f(x) - f(y)|| < L||x-y||$$

(d.h. f ist stark kontrahierend). Dann hat f auf D genau einen Fixpunkt s und für $x^{(0)} \in D$ konvergiert die Folge $(x^{(m)})_{m=0}^{\infty}$ mit

$$x^{(m+1)} = f(x^{(m)}) \qquad (m \geq 0) \tag{5}$$

gegen diesen Fixpunkt (also $\lim\limits_{m\to\infty} x^{(m)} = s$). Es gilt die Fehlerabschätzung

$$||x^{(m)} - s|| \leq \frac{L^m}{1-L} ||x^{(1)} - x^{(0)}|| \tag{6}$$

Wie im Eindimensionalen (vgl. 2.1) folgt die starke Kontraktion, wenn nachgewiesen werden kann, daß gilt:

$$\exists \quad L < 1 \quad \forall \quad x \in D: \qquad ||f'(x)|| \leq L$$

Dabei ist die Ableitung durch

$$f'(x) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{pmatrix} \tag{7}$$

definiert. Außerdem muß die verwendete Matrixnorm zur Vektornorm passend sein. Die Größe von L bestimmt die Konvergenzgeschwindigkeit.

3.3 Das Newton-Verfahren im $\mathbb{R}^n$

Def. 8 Sei $(x^{(m)})_{m=0}^{\infty}$ eine Iterationsvorschrift im $\mathbb{R}^n$.

1. Die Menge $U_\varepsilon(a) := \{x \in \mathbb{R}^n : ||x - a|| \leq \varepsilon\}$ heißt *Kugel* um a mit Radius ε. Eine Menge $U(a)$ heißt *Umgebung* um a, falls gilt:

$$\exists \quad \varepsilon > 0: \qquad U_\varepsilon(a) \subseteq U(a)$$

2. Ein Fixpunkt s einer Funktion f heißt *anziehend*, falls gilt

$$\exists \quad \varepsilon > 0 \quad \forall \quad x^{(0)} \in U_\varepsilon(s): \qquad \lim_{m \to \infty} x^{(m)} = s$$

3. Die Vorschrift $(x^{(m)})$ konvergiert (mindestens) von der Ordnung k bezüglich s, falls s anziehender Fixpunkt ist und es gilt:

$$\forall \ x^{(0)} \in U_\varepsilon(s) \ \exists \ n_0 \in \mathbb{N}, c \in \mathbb{R}: \quad ||x^{(m+1)} - s|| \leq c||x^{(m)} - s||^k$$

Satz 5 Sei $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ eine Funktion und $s \in \mathbb{R}^n$ mit $\varphi(s) = 0$. φ sei differenzierbar in einer Umgebung $U(s)$ und es gelte

$$\exists \quad L < 1 \quad \forall \quad u, v \in U(s): \qquad ||\varphi'(u) - \varphi'(v)|| \leq L||u - v||$$

$||\varphi'(u) - \varphi'(v)|| \leq K||u - v||$ für ein $K < 1$ und alle $u, v \in \mathbb{R}^n$. Für alle $x \in U(s)$ sei $\varphi'(x)$ invertierbar. Dann ist s anziehender Fixpunkt der Iteration

$$x^{(m+1)} = x^{(m)} - \left(\varphi'(x^{(m)})\right)^{-1} \cdot \varphi(x^{(m)}) \tag{8}$$

Diese Iteration ist quadratisch.

4 Gleitkommarechnen und Fehler

4.1 Gleitkommazahlen

Darstellung von Zahlen im Rechner:

$$x = \pm a \cdot 10^b \qquad 0.1 \leq a < 1 \text{ für } x \neq 0$$

Dabei heißt a Mantisse und b Exponent. Wenn die Mantisse t und der Exponent e Dezimalstellen hat, dann heißt $n := t + e$ die *Wortlänge* der Zahl x. A sei die Menge aller exakt darstellbaren Zahlen. A ist endlich.

4.2 Rundung

Betrachten wir die Zahl $x = a \cdot 10^b$ mit der Mantisse $a := 0, \alpha_1\alpha_2 \ldots \alpha_t\alpha_{t+1} \ldots$ Damit $x \in A$ muß für alle $m > t$ gelten $\alpha_m = 0$. Ist dies nicht der Fall, so wird x auf $\tilde{x} := \tilde{a} \cdot 10^b$ gerundet, wobei

$$\tilde{a} := \begin{cases} 0, \alpha_1\alpha_2 \ldots \alpha_{t-1}\alpha_t & \text{falls } 0 \leq \alpha_{t+1} \leq 4 \\ 0, \alpha_1\alpha_2 \ldots \alpha_{t-1}(\alpha_t + 1) & \text{falls } 5 \leq \alpha_{t+1} \leq 9 \end{cases}$$

Damit ist gewährleistet, daß $\tilde{x} \in A$ und $\forall\ q \in A :\ |x - \tilde{x}| \leq |x - q|$, d.h. $\tilde{x}$ ist die x am nächsten gelegene Maschinenzahl.

Möglicherweise liegt das nach der angegebenen Vorschrift berechnete $\tilde{x}$ aber nicht mehr in A. Beim „Überlauf“ wird durch die Rundung $b > 10^e$. In diesem Fall wird die Rechnung mit einem Fehler abgebrochen. Beim „Unterlauf“ wird durch die Rundung $b \leq 10^{-e}$. In diesem Fall wird mit $\tilde{x} := 0$ weitergerechnet.

Der durch das Runden in Kauf genommene Fehler ist durch

$$\left|\frac{\tilde{x} - x}{x}\right| \leq \frac{0.5 \cdot 10^{-t}}{0.1} = 5 \cdot 10^{-t} =: \tau$$

beschränkt. τ heißt „Maschinengenauigkeit“.

Beachte: Arithmetrische Grundgesetze gelten in A nicht durchgehend, z.B. führt für alle $x, y \in A$ mit $0 < |y| < \frac{\tau}{10}|x|$, die Rechnung $x + y = x$ zum Widerspruch.

4.3 Fehler bei arithmetrischen Operationen

Def. 9 Seien $x, y \in \mathbb{R}$ mit den (absoluten) Fehlern $\Delta x, \Delta y$. Dann heißen

$$\varepsilon_x := \frac{\Delta x}{x} \qquad \text{und} \qquad \varepsilon_y := \frac{\Delta y}{y}$$

die *relativen Fehler* von x und y.

Für den relativen Fehler von $z := x + y$ gilt

$$\varepsilon_z = \frac{\Delta z}{z} = \frac{\Delta x + \Delta y}{x + y} = \frac{x}{x+y}\varepsilon_x + \frac{y}{x+y}\varepsilon_y$$

Haben x, y dasselbe Vorzeichen, so gilt $\varepsilon_z \leq \max\{\varepsilon_x, \varepsilon_y\}$. Für unterschiedliche Vorzeichen, insbesondere für $x \to -y$ verstärken sich die relativen Fehler, denn $|\frac{x}{x+y}| \longrightarrow \infty \; (x \to -y)$. Diesen Effekt nennt man *numerische Auslöschung.*

4.4 Fehlerfortpflanzung

Satz 6 Sei $F : \mathbb{R}^n \to \mathbb{R}^n$ stetig differenzierbar und $x \in \mathbb{R}^n$ mit einem absoluten Fehler $\Delta x \in \mathbb{R}^n$. Für den absoluten Fehler $\Delta z := F(x+\Delta x) - F(x)$ gilt: Es ex. ein $\xi \in s(x, x + \Delta x)$ mit

$$\Delta z = F(x + \Delta x) - F(x) = \sum_{i=1}^{n} \underbrace{\frac{\partial F(\xi)}{\partial x_i}}_{\substack{\text{Konditions-}\\ \text{zahlen}}} \Delta x_i \approx \sum_{i=1}^{n} \frac{\partial F(x)}{\partial x_i} \Delta x_i$$

Der relative Fehler ergibt sich aus $\varepsilon_z := \frac{\Delta z}{z}$ zu

$$|\varepsilon_z| \approx \sum_{i=1}^{n} \underbrace{\left| \frac{x_i}{F(x_i)} \frac{\partial F(x)}{\partial x_i} \right|}_{\substack{\text{Kond.-Zahlen für}\\ \text{den relativen Fehler}}} |\varepsilon_{x_i}|$$

5 Polynome

5.1 Auswertung von Polynomen

Sei $p_n(x) := a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1} x + a_n$ ein Polynom. Gesucht ist der Funktionswert $p_n(r)$ für ein vorgegebenes $r \in \mathbb{R}$. Schreibe p_n in der Form

$$p_n(r) = (\ldots \underbrace{(\underbrace{(\underbrace{(a_0 r + a_1)}_{=:b_1} r + a_2)}_{=:b_2} r + a_3)}_{=:b_3 \ldots} r + \ldots + a_{n-1}) r + a_n$$

$p_n(r) = b_n$ ergibt sich damit durch die Rekursionsformel („HORNER-Schema“)

$$b_0 = a_0, \qquad b_k = a_k + r b_{k-1} \quad (k = 1, \ldots, n) \tag{9}$$

Aufwand: n Multiplikationen, n Additionen. (Dagegen kostet das Ausrechnen nach Definition $(2n-1)$ Multiplikationen und n Additionen). Zur Handrechnung empfielt sich die in Abb. 2 gezeigte Darstellung.

Das HORNER-Schema läßt sich auch zur Polynomendivision verwenden. Mit $p_{n-1}(x) = b_0 x^{n-1} + b_1 x^{n-2} + \ldots + b_n$ wird $p_n(x) = p_{n-1}(x)(x-r) + b_n$. Ist r eine Nullstelle von p_n, so hat p_{n-1} dieselben Nullstellen wie p_n mit Ausnahme von r, es gilt $p_n(x) = p_{n-1}(x)(x-r)$.

5.2 Vollständiges HORNER-Schema

Prinzip: Wende auf p_{n-1} wieder das HORNER-Schema an

$$\begin{aligned}
p_n(x) &= p_{n-1}(x)(x-r) + \overline{b}_n \\
p_{n-1}(x) &= p_{n-2}(x)(x-r) + \overline{b}_{n-1} \\
p_{n-2}(x) &= p_{n-3}(x)(x-r) + \overline{b}_{n-2} \\
&\vdots \\
p_1(x) &= p_0(x)(x-r) + \overline{b}_1 \\
p_0(x) &= \overline{b}_0
\end{aligned}$$

Zusammengesetzt ergibt sich

$$p_n(x) = \overline{b}_0(x-r)^n + \overline{b}_1(x-r)^{n-1} + \ldots + \overline{b}_{n-1}(x-r) + \overline{b}_n$$

Diese Darstellung heißt *Umentwicklung* von p_n nach Potenzen von $(x-r)$, denn

$$p_n(x) = \sum_{i=0}^{n} a_{n-i} x^i = \sum_{j=0}^{n} \overline{b}_{n-j}(x-r)^j$$

$$p_4(x) = 2x^4 + 3x^3 - 7x^2 - 12x + 1, \quad x_0 = 3$$

$$\begin{array}{c|rrrrr}
 & 2 & 3 & -7 & -12 & 1 \\
 & 0 & 6 & 27 & 60 & 144 \\
\hline
\boxed{x_0 = 3} & 2 & 9 & 20 & 48 & \underline{145}
\end{array} \implies p_4(3) = 145$$

$$p_4(x) = (2x^3 + 9x^2 + 20x + 48)(x-3) + 145$$

Abbildung 2: Beispiel für Polynomauswertung nach HORNER

Die Berechnung der $\overline{b}_j$ heißt *vollständiges* HORNER-*Schema*. Außerdem gilt, daß die $\overline{b}_j$ die gleichen Koeffizienten sind, die auch im TAYLOR-Polynom bei der TAYLOR-Entwicklung an der Stelle r entstehen.

5.3 Lage der Nullstellen eines Polynoms

Für Matrizen $A = ((a_{ij})) \in \mathbb{C}^{n\times n}$ läßt sich zeigen, daß die Eigenwerte von A in bestimmten Kreisen in der komplexen Ebene liegen. Diese Aussage macht der Satz von GERSHGORIN: Ist λ ein Eigenwert von A, so gilt:

$$\lambda \in \bigcup_{i=1}^{n} K_i \quad \text{mit} \quad K_i = \left\{ z \in \mathbb{C} : \; |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \right\}$$

Die Kreise K_i heißen GERSHGORIN-*Kreise*. Dieser Satz läßt sich für Polynome ausnutzen. Zum Polynom p mit

$$p(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

(beachte: $a_n = 1$) betrachtet man die FROBENIUS-sche Begleitmatrix:

$$F(p) := \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \ldots & \ldots & 0 & 1 \\ -a_0 & -a_1 & \ldots & -a_{n-2} & -a_{n-1} \end{pmatrix} \in \mathbb{R}^{n\times n}$$

Mit dem Satz von GERSHGORIN und dieser Matrix läßt sich der folgende Satz beweisen:

Satz 7 Sei $p_n = x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ ein Polynom mit den Nullstellen $x_1, x_2, \ldots, x_n$. Dann gilt für $1 \leq j \leq n$:

1. $|x_j| \leq \max\left\{\sum\limits_{i=0}^{n-1} |a_i|, 1\right\}$
2. $|x_j| \leq \max\{|a_0|, 1 + |a_1|, \ldots, 1 + |a_n|\}$

Die Berechnung von Nullstellen ist schlecht konditioniert. D.h. kleine Änderungen an den Koeffizienten (z.B. durch Rundung) können zu völlig anderen Ergebnissen für die Nullstellen führen.

6 Interpolation

6.1 Das Interpolationspolynom nach Lagrange

Satz 8 Sei $f : I = [a,b] \to \mathbb{R}$ eine Funktion und $x_i \in I$ $(i = 0, \ldots, n)$. Es gelte $x_i \neq x_j$ $(i \neq j)$. Dann ist p_n mit

$$p_n(x) = \sum_{k=0}^{n} f(x_k) \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i} \tag{10}$$

das einzige Polynom mit Grad $p_n \leq n$ und $p_n(x_i) = f(x_i)$ $(i = 0, \ldots, n)$.

Die Produkte

$$l(x) := \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}$$

heißen Lagrange-*Grundpolynome*.

6.2 Fehlerabschätzung

Satz 9 Sei $f : I = [a,b] \to \mathbb{R}$, $f \in C^{(n+1)}([a,b], \mathbb{R})$, p_n sei das Interpolationspolynom nach Satz 8 an den Stützstellen (Knoten) $x_0, \ldots, x_n \in I$. Weiter sei ω_n mit

$$\omega_n(x) := \prod_{i=0}^{n} (x - x_i)$$

(ω_n heißt „Knotenpolynom" zu $x_0, \ldots, x_n$). Dann gilt:

$$\forall \quad x \in I \quad \exists \quad \xi \in I : \qquad f(x) - p_n(x) = \frac{\omega_n(x)}{(n+1)!} f^{(n+1)}(\xi) \tag{11}$$

7 Numerische Integration

7.1 Newton-Cotes-Formeln

Gesucht sind Näherungen für

$$F := \int_a^b f(x)dx \qquad (a, b, \in \mathbb{R}, f \in R([a,b]))$$

Prinzip: Wähle eine äquidistante Zerlegung von $I = [a, b]$ und interpoliere über diese Stützstellen. Nach den Sätzen 8 und 9 gibt es eine Darstellung

$$F = \int_a^b f(x)dx = \underbrace{\int_a^b p_n(x)dx}_{=:F_n} + \underbrace{\int_a^b r_n(x)dx}_{=:R_n}$$

Dabei kann F_n geschlossen ausgewertet und R_n abgeschätzt werden. Auf eine explizite Aufstellung von p_n kann verzichtet werden. Setzt man

$$\tilde{l}_k(s) := \prod_{\substack{i=0\\i\neq k}}^{n} \frac{s-i}{k-i}, \qquad c_k^{(n)} := \frac{1}{n}\int_0^n \tilde{l}_k(s)ds \tag{12}$$

so läßt sich F_n schreiben als

$$F_n = \int_a^b p_n(x)dx = (b-a)\sum_{k=0}^{n} f(x_k)c_k^{(n)} \tag{13}$$

Diese Darstellung heißt Newton-Cotes-*Formel*, die Zahlen $c_k^{(n)}$, die unabhängig von f sind und somit vorausberechnet werden können, heißen Newton-Cotes-*Zahlen*. Für diese Zahlen gilt:

$$\sum_{k=0}^{n} c_k^{(n)} = 1, \qquad c_k^{(n)} = c_{n-k}^{(n)}$$

7.2 Trapezregel

Setzt man in (13) $n = 1$ mit den Stützstellen $x_0 = a, x_1 = b$, so erhält man die *einfache Trapezregel* ($c_0^{(1)} = c_1^{(1)} = \frac{1}{2}$):

$$F_1 = \frac{b-a}{2}\left(f(a) - f(b)\right) \tag{14}$$

Teilt man das Intervall $[a, b]$ in N Teilintervalle, so ergibt sich mit (14) durch Setzung von

$$h := \frac{b-a}{N}, \quad x_0 = a, \quad x_i = x_0 + ih, \quad x_n = b$$

die *allgemeine Trapezregel*:

$$F_{TR}^{(N)} = \frac{h}{2}\left(f(x_0) + 2f(x_1) + 2f(x_2) + \ldots + 2f(x_{N-1}) + f(x_N)\right) \tag{15}$$

Diese Berechnung ist von der Ordnung 2 und es gilt

$$|R_{TR}^{(N)}| \leq M_2 \frac{b-a}{12} h^2 \qquad (|f''(x)| \leq M_2 \text{ f.a. } x \in [a, b])$$

als Fehlerabschätzung.

7.3 Simpson-Regel

Setzt man in (13) $n = 2$, so ergibt sich die *einfache* Simpson-*Regel*, auch Kepler-*sche Faßregel*:

$$F_2 = \frac{b-a}{2}\left(f(a) + 4f(\frac{a+b}{2}) + f(b)\right) \tag{16}$$

Teilt man das Intervall $[a, b]$ in $2N$ Teilintervalle, $h = \frac{b-a}{2N}$ und wendet auf jedes Intervall der Länge $2h$ die obige Gleichung an, so folgt die *allgemeine* Simpson-*Regel*:

$$F_S^{(N)} = \frac{h}{3}\left(f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots + 2f_{2N-2} + 4f_{2N-1} + f_{2N}\right) \tag{17}$$

Dabei sei $f_i := f(x_i)$. Dieses Verfahren ist von der Ordnung 4 und es gilt

$$|R_S^{(N)}| \le M_4 \frac{b-a}{180} h^4$$

für ein M_4 mit $|f^{(4)}(x)| \le M_4$ als Fehlerabschätzung.

7.4 Romberg-Integration

Gesucht ist $F = \int_a^b f(x)dx$, wobei f genügend oft differenzierbar sei. Ohne allzugroßen Fehler gilt die Abschätzung

$$F_S^{(N)} = \frac{4F_{TR}^{(N)} - F_{TR}^{(2N)}}{3} = \frac{2^2 F_{TR}^{(N)} - F_{TR}^{(2N)}}{2^2 - 1}$$

Dadurch wird die Ordung von 2 auf 4 „aufgewertet“. Diese Überlegung läßt sich fortführen:

$$R_N := \frac{2^4 F_S^{(N)} - F_S^{(2N)}}{2^4 - 1}$$

konvergiert mit der Ordnung 6, entsprechend konvergiert P_N mit

$$P_N := \frac{2^6 R_N - R_{2N}}{2^6 - 1}$$

von der Ordnung 8 usw.

Man berechnet also zunächst die Werte für $F_{TR}^{(N)}, F_{TR}^{(2N)}, F_{TR}^{(4N)}$ und $F_{TR}^{(8N)}$. Daraus lassen sich nach den obigen Formeln $F_S^{(N)}, F_S^{(2N)}$ und $F_S^{(4N)}$ berechnen und damit R_N und R_{2N} und schließlich P_N (vgl. Abb. 3).

Die Trapezsummen $F_{TR}^{(N)}$ lassen sich rekursiv nach

$$F_{TR}^{(2N)} = \frac{1}{2} F_{TR}^{(N)} + h \cdot (f(x_1) + f(x_3) + \cdots + f(x_{2^k-1})) \quad (k \in \mathbb{N})$$

berechnen.

8 Spline-Interpolation

8.1 Einführung, Definition

Def. 10 Sei $I := [a, b]$ ein Intervall und $\Delta := \{x_0, x_1, \ldots, x_n\}$ mit $x_0 = a$, $x_n = b$ und $x_i < x_{i+1}$ für alle $0 < i < n$. S_Δ heißt die zu Δ gehörende *Splinefunktion* oder *kubischer Spline*, falls gilt:

1. $S_\Delta \in C^2([a, b])$
2. In jedem Teilintervall $[x_i, x_{i+1}]$ $(i = 0, \ldots, n-1)$ stimmt S_Δ mit einem Polynom dritten Grades überein.

Das heißt: Die Funktion S_Δ ist auf den Intervallen $[x_i, x_{i+1}]$ so definiert, daß sie auch an allen Stellen x_i stetig differenzierbar ist (dazwischen sowieso!).

Def. 11 Sei $y = (y_0, y_1, \ldots, y_n) \in \mathbb{R}^{n+1}$ und $\Delta = \{x_0, \ldots, x_n\}$. $S_\Delta(y, \cdot)$ heißt *interpolierende Splinefunktion*, falls gilt:

1. $S_\Delta(y, \cdot)$ ist kubischer Spline zu Δ
2. $S_\Delta(y, x_i) = y_i \quad (i = 0, \ldots, n)$ (Interpolationseigenschaft)

8.2 Analytische Aussagen

Satz 10 Zu jeder Zerlegung Δ und zugehörigem $y \in \mathbb{R}^{n+1}$ existiert ein $S_\Delta(y, \cdot)$. Dieses $S_\Delta(y, \cdot)$ ist eindeutig bestimmt, falls eine der folgenden Bedingungen erfüllt ist:

1. $S''_\Delta(y, a) = S''_\Delta(y, b) = 0$
2. f und $S_\Delta(y, \cdot)$ sind periodisch

$$
\begin{array}{ccccccc}
F_{TR}^{(8N)} & & & & & & \\
 & \rangle & F_S^{(4N)} & & & & \\
F_{TR}^{(4N)} & & & \rangle & R_{2N} & & \\
 & \rangle & F_S^{(2N)} & & & \rangle & P_N \ldots \\
F_{TR}^{(2N)} & & & \rangle & R_N & & \\
 & \rangle & F_S^{(N)} & & & & \\
F_{TR}^{(N)} & & & & & &
\end{array}
$$

Abbildung 3: Systematische Berechnung von $T_N, S_N, R_N, \ldots$

3. $f'(a) = S'_\Delta(y, a)$ und $f'(b) = S'_\Delta(y, b)$

Satz 11 Sei $S_\Delta(y, \cdot)$ ein interpolierender Spline zu Δ und $y \in \mathbb{R}^{n+1}$ und $g \in C^2([a, b])$ eine beliebige interpolierende Funktion (also $g(x_i) = y_i$ für $(i = 0, \ldots, n)$). Dann gilt:

$$\int_a^b g''(x)^2 dx \geq \int_a^b S''_\Delta(Y, x)^2 dx \tag{18}$$

Satz 12 Sei $f \in C^4([a, b])$ mit $|f^{(4)}(x)| \leq M$ für $x \in [a, b]$. Sei $S_{\Delta_m}(y, \cdot)$ der zu $y := (f(x_1), \ldots, f(x_n))$ und Δ_m gehörende interpolierende Spline mit $S'_{\Delta_m}(y, a) = f'(a)$ und $S'_{\Delta_m}(y, b) = f'(b)$. Dann gilt für beliebiges $x \in [a, b]$

$$\left| f^{(i)}(x) - S^{(i)}_{\Delta_m}(y, x) \right| \leq c_i \cdot M \cdot h_m^{4-i} \qquad (i = 0, 1, 2, 3) \tag{19}$$

Das heißt $f^{(i)}$ wird von der Ordnung h_m^{4-i} approximiert.

8.3 Berechnung

Nach Satz 10 existiert ein eindeutiges $S_\Delta(y, \cdot)$ zu y, wenn eine der drei folgenden Bedingungen erfüllt ist:

1. $S''_\Delta(y, a) = S''_\Delta(y, b) = 0$
2. f und $S_\Delta(y, \cdot)$ sind periodisch
3. $f'(a) = S_\Delta(y, a)$ und $f'(b) = S'_\Delta(y, b)$

$S_\Delta(y, \cdot)$ habe für $j = 0, \ldots, n-1$ auf $I_j := [x_j, x_{j+1}]$ die Form

$$S_\Delta(y, x) = \alpha_j + \beta_j(x - x_j) + \gamma_j(x - x_j)^2 + \delta_j(x - x_j)^3 \tag{20}$$

Gesucht sind nun die Koeffizienten $\alpha_j, \beta_j, \gamma_j, \delta_j,\ (j = 0, \ldots, n-1)$.
Man bezeichnet

$$M_j := S''_\Delta(y, x_j) \qquad (i = 0, \ldots, n)$$

als *Momente* von $S_\Delta(y, \cdot)$. Für $j = 1, \ldots, n-1$ und $h_j := x_j - x_{j-1}$ gilt

$$\frac{h_j}{6} M_{j-1} + \frac{h_j + h_{j-1}}{3} M_j + \frac{h_{j+1}}{6} M_{j+1} = \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j}. \tag{21}$$

Das sind $n-1$ lineare Gleichungen für die $n+1$ Unbekannten $M_0, \ldots, M_n$. Zwei weitere Gleichungen lassen sich aus den Bedingungen ableiten:

1. zu Bed. 1: $M_0 = 0$ sowie $M_n = 0$

2. zu Bed. 2: $M_0 = M_n$ sowie

$$\frac{h_n}{6} M_{n-1} + \frac{h_n + h_1}{6} M_n + \frac{h_1}{6} M_1 = \frac{y_1 - y_n}{h_1} - \frac{y_n - y_{n-1}}{h_n}$$

3. zu Bed. 3: $f'(a) = S'_\Delta(y, a)$ sowie $f'(b) = S'_\Delta(y, b)$

Diese Gleichungen ergeben zusammen ein LGS $Ax = b$ ($A \in \mathbb{R}^{(n+1)\times(n+1)}$, $b \in \mathbb{R}^{n+1}$), wobei $x = (M_0, \ldots, M_n)^\top$ ist. Aus den M_j kann man die Koeffizienten $\alpha_j, \beta_j, \gamma_j, \delta_j$ aus (20) bestimmen zu

$$\begin{aligned} \alpha_j &= y_j \\ \beta_j &= \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{2M_j + M_{j+1}}{6} h_{j+1} \\ \gamma_j &= \frac{M_j}{2} \\ \delta_j &= \frac{M_{j+1} - M_j}{6h_{j+1}}. \end{aligned} \tag{22}$$

Zusammenfassung des Algorithmus

Schritt A: Bestimme die Koeffizienten von A und b aus (21) und den Gleichungen aus den Bedingungen

Schritt B: Löse das Gleichungssystem

$$\begin{pmatrix} 2 & \lambda_1 & 0 & \cdots & 0 \\ \mu_1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \lambda_{n-1} \\ 0 & \cdots & 0 & \mu_{n-1} & 2 \end{pmatrix} \begin{pmatrix} M_1 \\ \vdots \\ \vdots \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} d_0 \\ \vdots \\ \vdots \\ \vdots \\ d_n \end{pmatrix}$$

mit

$$\begin{aligned} \lambda_i &= \tfrac{h_{i+1}}{h_i + h_{i+1}}, \ \mu_i = 1 - \lambda_i & \qquad (i = 0, \ldots, n-1) \\ d_i &= \tfrac{6}{h_i + h_{i+1}} \left(\tfrac{y_{i+1} - y_i}{h_{i+1}} - \tfrac{y_j - y_{j-1}}{h_j} \right) & \qquad (i = 1, \ldots, n-2) \end{aligned}$$

und d_0, d_{n-1} aus der Randbedingung

Schritt C: Berechne $\alpha_j, \beta_j, \gamma_j, \delta_j$ nach (22).

8.4 Ergänzungen

Zu einem vorgegebenen $\Delta := \{a = x_0 < x_1 < \ldots < x_n = b\}$ heißt allgemeiner $S_\Delta \in C^{m+1}([a,b])$ *Splinefunktion* vom Grad m $(m \geq 1)$, falls S_Δ auf $I_k = [x_k, x_{k+1}]$ $(k = 0, \ldots, n-1)$ mit Polynomen vom Grad m übereinstimmt. Für $m = 1$ ist S_Δ ein Polygonenzug.

Als *B-Splines* bezeichnet man Splines, in denen auch „abgeschnittene“ Potenzen $(t^m)_+$ mit

$$(t^m)_+ := \begin{cases} t^m & t \geq 0 \\ 0 & t < 0 \end{cases}$$

verwendet werden. Theorie und Konstruktion sind hier wesentlich schwieriger.

9 BERNSTEIN-Polynome, BEZIER-Darstellung von Polynomen

9.1 BERNSTEIN-Polynome

Def. 12 Es sei $\Pi_n := \{p \in \mathbb{R}[x] : \text{ Grad } p \leq n\}$. Die Polynome $b_{\nu,n}$ $(n \in \mathbb{N}_0)$ mit

$$b_{\nu,n}(t) := \binom{n}{\nu} t^\nu (1-t)^{n-\nu} \qquad (\nu = 0, \ldots, n) \tag{23}$$

heißen BERNSTEIN-*Polynome*.

Satz 13 Für $n \in \mathbb{N}$ und $0 \leq \nu \leq n$ gilt:

1. $b_{\nu,n}$ hat eine ν-fache Nullstelle in $t = 0$ und eine $(\nu - 1)$-fache Nullstelle in $t = 1$.

2. $b_{\nu,n}$ ist strikt positiv für $0 < t < 1$ und hat genau ein Maxiumum in $t = \frac{\nu}{n}$ $(\nu = 0, \ldots, n)$.

3. Die Polynome $b_{\nu,n}$ $(\nu = 0, \ldots, n)$ sind linear unabhängig und bilden eine Basis von Π_n.

Satz 14 Für $t \in \mathbb{R}$ gilt:

$$\sum_{\nu=0}^{n} b_{\nu,n}(t) = 1 \qquad (n \in \mathbb{N}_0) \tag{24}$$

$$\sum_{\nu=0}^{n} \frac{\nu}{n} b_{\nu,n}(t) = t \qquad (n \in \mathbb{N}) \tag{25}$$

9.2 BEZIER-Darstellung eines Polynoms

Da die Polynome $b_{\nu,n}$ eine Basis von Π_n bilden, gibt es für jedes $p \in \Pi_n$ eine Darstellung der Form

$$p(t) = \sum_{\nu=0}^{n} \beta_\nu b_{\nu,n}(t) \tag{26}$$

Diese Darstellung heißt BEZIER-*Darstellung* von p. Die Koeffizienten β_ν ($\nu = 0, \ldots, n$) heißen BEZIER-*Koeffizienten*. Die Punkte $(\frac{\nu}{n}|\beta_\nu) \in \mathbb{R}^2$ heißen BEZIER-*Punkte* und als BEZIER-*Polygon* bezeichnet man die Verbindung dieser Punkte.

Satz 15 Für $t \in [0,1]$ liegt der Graph von p in der konvexen Hülle seiner BEZIER-Punkte.

9.3 Der Algorithmus von CASTELJAU

Zur Auswertung eines Polynoms p an einer Stelle t. p sei in seiner BEZIER-Form, also durch die Koeffizienten $\beta_0, \ldots, \beta_n$ bestimmt. Setze

$$\beta_\nu^{(0)} := \beta_\nu \qquad (\nu = 0, \ldots, n)$$

und berechne rekursiv

$$\beta_\nu^{(k)} := (1-t)\beta_n^{(k-1)} + t\beta_{\nu+1}^{(k-1)} \qquad (\nu = 0, \ldots, k-1) \tag{27}$$

Dann gilt $p(t) = \beta_0^{(n)}$.

10 Lösung von Anfangswert- und Randwertproblemen bei gewöhnlichen Differentialgleichungen

10.1 Anfangswertprobleme

Def. 13 Sei $f : [a,b] \times \mathbb{R} \to \mathbb{R}$ und $y : [a,b] \to \mathbb{R}$ eine (unbekannte) Funktion mit $y(x_0) = y_0$. Eine Gleichung der Form

$$y' = f(x,y)$$

heißt *Anfangswertproblem*. Die Definitionsmenge von f, $[a,b] \times \mathbb{R}$, heißt *Streifen*.

Satz 16 (Existenzsatz von PICARD-LINDELÖF) Es sei $||\cdot||$ eine Norm im $\mathbb{R}^n$,

$$S := \{(x,y) \in \mathbb{R}^{n+1} : \quad a \le x \le b, \quad y \in \mathbb{R}^n\}.$$

$f : S \to \mathbb{R}^n$ sei stetig und genüge der LIPSCHITZ-Bedingung

$$\forall \quad x \in [a,b], y_1, y_2 \in \mathbb{R}^n : \qquad ||f(x,y_1) - f(x,y_2)|| \le L||y_1 - y_2||. \tag{28}$$

Dann gilt: Zu jedem $x_0 \in [a,b], y_0 \in \mathbb{R}^n$ existiert genau eine Lösung $y : [a,b] \to \mathbb{R}$ für das Problem $y' = f(x,y)$ mit

1. y ist stetig differenzierbar in $[a,b]$
2. $y'(x) = f(x,y(x))$
3. $y(x_0) = y_0$

10.2 Polygonzugverfahren nach EULER-CAUCHY

Die Funktion y wird durch einen Polygonenzug approximiert. Wähle dazu eine äquidistante Zerlegung $Z = \{x_0, \ldots, x_n\}$ mit $x_0 = a, x_n = b$ und $x_i = a + ih, h = \frac{b-a}{n}$. Der Startwert $y(x_0) = y_0$ ist vorgegeben. Die Werte an den übrigen Stellen werden berechnet zu

$$y(x_{i+1}) = y(x_i) + hf(x_i, y(x_i)) \qquad (i = 0, \ldots, n-1) \tag{29}$$

Dieses Verfahren konvergiert von der Ordnung 1.

10.3 RUNGE-KUTTA-Verfahren

Über den Ansatz

$$y(x+h) - y(x) = h\frac{1}{h}\int_x^{x+h} f(t, y(t))dt \tag{30}$$

lassen sich die Verfahren der numerischen Integration (vgl. 7, Seite 14) anwenden:

1. Verfahren von HEURE: (Ordnung 2)

$$y(x_{i+1}) = y(x_i) + \frac{h}{2}(k_0 + k_1) \tag{31}$$

mit
$k_0 = f(x_i, y(x_i))$
$k_1 = f(x_{i+1}, y(x_i) + hk_0)$

2. (Ordnung 3)

$$y(x_{i+1}) = y(x_i) + \frac{h}{6}(k_0 + 4k_1 + k_2) \tag{32}$$

mit
$k_0 = f(x_i, y(x_i))$
$k_1 = f(x_i + \frac{h}{2}, y(x_i) + \frac{h}{2}k_0)$
$k_2 = f(x_i + h, y(x_i) + h(2k_1 - k_0))$

3. Klassisches Runge-Kutta-Verfahren: (Ordnung 4)

$$y(x_{i+1}) = y(x_i) + \frac{h}{6}(k_0 + 2k_1 + 2k_2 + k_3) \tag{33}$$

mit
$k_0 = f(x_i, y(x_i))$
$k_1 = f(x_i + \frac{h}{2}, y(x_i) + \frac{h}{2}k_0)$
$k_2 = f(x_i + \frac{h}{2}, y(x_i) + \frac{h}{2}k_1)$
$k_3 = f(x_i + h, y(x_i) + hk_2)$

10.4 Diskretisierung des Randwertproblems

Gesucht ist eine Funktion $y \in C^2([a,b])$ mit

$$-y''(x) + q(x)y(x) = g(x) \tag{34}$$

mit den *Randwerten* $y(a) = \alpha, y(b) = \beta$. Die Funktion y soll an den Stellen x_i mit $x_0 = a, x_i = a + ih, x_n = b, h = \frac{b-a}{n}$ approximiert werden. Dies geschieht durch Lösen des Liearen Gleichungssystems

$$\begin{pmatrix} 2+g_1h^2 & -1 & 0 & \cdots & 0 \\ -1 & 2+g_2h^2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2+g_nh^2 \end{pmatrix} \cdot \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ \vdots \\ q_n \end{pmatrix} \tag{35}$$

wobei $y_i := y(x_i), q_i := q(x_i)$ und $g_i = g(x_i)$ $(i = 0, \ldots, n)$ sein soll. Dieses Verfahren konvergiert mit der Ordnung 2.

11 Randwertprobleme bei partiellen DGLen

11.1 Lineare partielle DGL zweiter Ordnung

Def. 14 Eine Gleichung der Form

$$\gamma + \alpha U_{xx} + \beta U_{yy} = 0 \quad \text{auf} \quad Q := \{(x,y) \in \mathbb{R}^2 : \quad 0 < x, y < 1\} \tag{36}$$

für $\alpha, \beta, \gamma \in \mathbb{R}$ und eine unbekannte Funktion $U = U(x,y)$ heißt *partielle Differentialgleichung* (zweiter Ordnung). Die Gleichung (36) heißt *Laplace-Gleichung.*

Beispiele für partielle Differentialgleichungen mit Randbedingungen: Es seien g, h vorgegebene Funktionen.

1. Elliptische Gleichung:

$$\begin{aligned} -U_{xx} - U_{yy} = 0 &\qquad \text{auf } Q \\ U = g &\qquad \text{auf } \partial Q \end{aligned}$$

2. Hyperbolische Gleichung:

$$\begin{aligned} U_{yy} &= c^2 U_{xx} &\qquad &\text{auf } Q \\ U(x,0) &= g(x) &\qquad &(0 \leq x \leq 1) \\ U_y(x,0) &= h(x) &\qquad &(0 \leq x \leq 1) \end{aligned}$$

3. Parabolische Gleichung:

$$\begin{aligned} U_{yy} &= U_{xx} &\qquad &\text{auf } Q \\ U(x,y) &= h(x) &\qquad &(0 \leq x < 1) \\ U(0,y) &= g(y) &\qquad &(y \geq 0) \\ U(1,y) &= h(y) &\qquad &(y \geq 0) \end{aligned}$$

11.2 Diskretisierung des Randwertproblems

Für das Problem (36) mit der Nebenbedingung

$$U = g \quad \text{auf } \partial Q$$

gibt es eine eindeutige Lösung. Zur Approximation dieser Lösung wird Q „gerastert", d.h. Q wird nur durch die Punkte

$$(x_i, y_j) \in \mathbb{R}^2 \text{ mit } x_i = ih, y_j = jh \text{ und } h = \frac{1}{N+1}(1 \leq i, j \leq N)$$

repräsentiert. Näherungsweise gilt für $1 \le i, j \le N$:

$$U(x_i, y_j) = \frac{1}{4}(U(x_{i-1}, y_j) + U(x_{i+1}, y_j) + U(x_i, y_{j-1}) + U(x_i, y_{j+1}) \qquad (37)$$

(Fünfpunktstern). Diese Gleichung läßt sich als LGS der Ordnung $n = N^2$ für die Unbekannten $U(x_i, y_j)$ $(1 \le i, j \le N)$ auffassen. Definiere

$$u_k := U(x_i, y_j) \qquad k = (j-1)N + i, \quad (1 \le i, j \le N)$$

zur Auflösung der doppelten Indizierung. So hat das LGS die Form $Ax = b$, wobei $b \in \mathbb{R}^n$ die Werte von U auf dem Rand enthält und A die folgende Form hat:

$$A = \begin{pmatrix} \boxed{A_N} & \boxed{-I_N} & 0 & \cdots & 0 \\ \boxed{-I_N} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \boxed{-I_N} \\ 0 & \cdots & 0 & \boxed{-I_N} & \boxed{A_N} \end{pmatrix} \in \mathbb{R}^{N^2 \times N^2}$$

mit

$$A_N = \begin{pmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 4 \end{pmatrix} \in \mathbb{R}^{N \times N}, \ I_N \text{ Einheitsmatrix}$$

A heißt *Blocktridiagonalmatrix*. Zur Lösung solch großer LGS ist das Gauss-Verfahren (vgl. 1.2, Seite 3) nicht geeignet.

11.3 Iterationsverfahren zur Lösung großer LGS

Gesamtschrittverfahren: Sei $A \in \mathbb{R}^{n \times n}$ mit diag $A = I$ und $Ax = b$ ein LGS. Zerlege A in eine untere Dreiecksmatrix L und eine obere Dreiecksmatrix R mit

$$A = I - L - R \qquad (38)$$

Damit läßt sich das LGS $Ax = b$ als

$$x = Lx + Rx + b \qquad (39)$$

schreiben. Auf diese Form läßt sich der Fixpunktsatz (vgl. 3.2, Seite 8) anwenden:

$$x^{(m+1)} = (L + R)x^{(m)} + b \qquad (m \in \mathbb{N}_0) \tag{40}$$

Um die Konvergenz dieser Iteration zu gewährleisten, müssen alle Eigenwerte von $L + R$ dem Betrag nach kleiner als 1 sein. Das *Einzelschrittverfahren* verwendet statt (40) die Iterationsvorschrift

$$x^{(m+1)} = (I - L)^{-1}(Rx^{(m)} + b) \qquad (m \in \mathbb{N}_0). \tag{41}$$

Beide Verfahren konvergieren langsam (nämlich in $O(n^2)$). Neuere Verfahren wie *Relaxionsmethoden* ($O(n)$), das *Verfahren der konjugierten Iteration* ($O(n)$) oder das *Mehrgitterverfahren (Multi-grid)* ($O(1)$) arbeiten deutlich schneller und sind entsprechend komplizierter.

Index