Numerische Mathematik für die Fachrichtungen Informatik und Ingenieurswesen

Lernzusammenfassung

Lukas Bach

21. Oktober 2017

am Karlsruher Institut für Technologie

21. Oktober 2017

Inhaltsverzeichnis

1 Lineare Gleichungssysteme

1.1 LR-Zerlegung

Gesucht wird eine Lösung x von $Ax = b$, dazu wird A zerlegt in $A = LR$, daraus folgt $Ax = LRx = Ly = b$ für $Rx = y$.

Dafür löst man zuerst $Ly = b$ nach y, dann $Rx = y$.

Ein Gleichungssystem $Lx = b$ mit linker unteren Dreiecksmatrix kann mit Vorwärtssubstitution gelöst werden, ein Gleichungssystem $Rx = b$ mit rechter oberer Dreiecksmatrix mit Rückwärtssubstitution.

1.1.1 Existenz einer LR-Zerlegung

Für eine invertierbare Matrix A existiert eine nicht eindeutige Permutationsmatrix P, sodass eine LR-Zerlegung $PA = LR$ möglich ist, und sodass L auf der Hauptdiagonalen nur 1en hat.

1.1.2 Vorgehensweise

1. Bestimme Matrizen P, L, R
   - Stelle Matrix auf: $A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$
   - In jeder $k$ten Iteration wählt man das Spaltenpivot: Tausche die (k)te Spalte mit der Spalte, deren $k$tes Element $a_{j,k}$ (also das Element am weitesten links, das nicht null ist), größer ist als alle Elemente $a_{n,k}$, also größer als alle anderen Elemente aus derselben Spalte.
   - In jeder $k$ten Iteration, nach der Spaltenpivotwahl, führe Eliminierungen mittels der $k$ten Zeile durch, sodass in der $k$ten Spalte unterhalb der Hauptdiagonalen nur Nullen stehen. Danach, ersetze diese Nullen durch die die negativen Eliminierungsoperatoren (z.B. wenn man auf die letzte Zeile die $k$te Zeile $\frac{3}{2}$ mal draufmultipliziert, wird in dieser Zeile an $k$ter Stelle $-\frac{3}{2}$ notiert.
2. Am Ende erhält man:

- P: Wenn in dem Permutationsarray in Zeile k die Ziffer i steht, so hat die Permutationsmatrix in Zeile k eine eins bei Spalte i.

1.2 Einschub: Gleitkommazahlen

Eine Gleitkommazahl a nimmt die Form an: $a = v \sum_{i=1}^{l} a_i d^{-i}$ und besteht aus:

- $v \in \{1, -1\}$ Vorzeichen
- $a = 0$ oder $d^{-1} \leq |a| < 1$ Mantisse
- $d = 2$ Basis
- $a_i \in \{0, ..., d-1\}$ (Normierung fordert $a_1 \neq 0$)
- l Mantissenlänge

1.3 Kondition von Gleichungssystemen

Die Konditionszahl $cond(A)$ einer Matrix A ist $cond_k(A) = ||A||_k \cdot ||A^{-1}||_k$.

Es gilt $cond(\alpha I) = 1$, $cond(U) = 1$ falls $U^T U = I$. Für $A \in \mathbb{R}^{m \times n}$ mit maximalem Rang $n \leq m$ gilt $cond_2(A^T A) = (cond_2(A))^2$.

1.4 Cholesky Verfahren

Für symmetrische positiv definite Matrizen A existiert eine Cholesky-Zerlegung $A = LL^T$, wobei L eine untere Dreiecksmatrix ist. Auf der Hauptdiagonalen von L müssen nicht notwendigerweise nur 1en stehen.

Zur Lösung von $Ax = b$, also $Ax = LL^T x = Ly = b$ geht man wie folgt vor: Löse $Ly = b$ durch Vorwärtssubstitution, dann $L^T x = y$ durch Rückwärtssubstitution.

Existiert eine solche Zerlegung $A = LL^T$ für eine invertierbare Matrix A, so gilt:

- A ist symmetrisch, da $A^T = (LL^T)^T = (L^T)^T L^T = LL^T = A$.
- A ist positiv definit ($x^T Ax > 0 \quad \forall x \in \mathbb{R}^n \backslash \{0\}$), da $x^T Ax = x^T LL^T x = (L^T x)^T L^T x = y^T y > 0$.

Eine invertierbare Matrix besitzt genau dann eine Cholesky-Zerlegung, wenn sie symmetrisch positiv definit ist.

1.5 QR-Zerlegung

Zu einer Matrix $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ (also eine „hohe“ Matrix) existiert eine Zerlegung $A = QR$, wobei $Q \in \mathbb{R}^{m \times m}$ orthogonal ($QQ^T = I$) und quatratisch ist und $R = \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix} \in \mathbb{R}^{m \times n}$ mit $\tilde{R} \in \mathbb{R}^{n \times n}$ als obere Dreiecksmatrix.

Vorgehensweise zur Lösung von $Ax = b$: Bestimme Q und R mittels Householder-Transformation, löse $Qc = b$ mittels $c = Q^{-1}b = Q^Tb$, dann löse $Rx = c$ durch Rückwärtssubstitution.

1.6 Lineare Ausgleichssysteme

Man versucht ein überbestimmtes Gleichungssystem $Ax = b$, das i.A. keine Lösung zu besitzt, durch $||Ax - b||_2 = min$ zu approximieren.

x ist genau dann eine Lösung zu einem linearen Ausgleichssystem, wenn $A^TAx = A^Tb$ gilt.

1.6.1 Algorithmus mittels QR-Zerlegung

Zu Lösen ist $||Ax - b||_2 = min$. Bestimme eine Zerlegung $A = QR = Q \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}$ mittels Householder-Transformation, also gilt dann $Ax - b = Q \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix} x - b$. Berechne dann c durch $Q^Tb = \begin{pmatrix} c \\ d \end{pmatrix}$. Löse dann $\tilde{R}x = c$ durch Rückwärtssubstitution.

1.7 Einschub: Normen

$|| \cdot ||$ heißt Norm $\Leftrightarrow ||x|| > 0$ und $||x|| = 0 \Leftrightarrow x = 0$ (pos. def.), $||x + y|| \leq ||x|| + ||y||$ (Dreiecksungleichung) und $||\lambda x|| = ||\lambda|| \cdot ||x||$ (Homogenität) $\forall x, y \in \mathbb{R}^N, \lambda \in \mathbb{R}$.

Die zu $|| \cdot ||$ gehörige Matrixnorm lautet $||A|| := \sup\limits_{x \neq 0} \frac{||Ax||}{||x||}$.

Weitere Matrizennormen sind:

$$||A||_1 = \max_{m=1,\dots,N} \sum_{n=1}^{N} |a_{nm}| \qquad \text{Spaltensummennorm: Welche ist die größte Spalte?}$$

$$||A||_\infty = \max_{n=1,\dots,N} \sum_{m=1}^{N} |a_{nm}| \qquad \text{Zeilensummennorm: Welche ist die größte Zeile?}$$

$$||A||_2 = \sqrt{\text{größter EW von } A^T A} \qquad \text{Spektralnorm}$$

1.8 Einschub: CCS Format

Eine Matrix $A \in \mathbb{R}^{N \times M}$ kann im CCS Format $v \in \mathbb{R}^k, z \in \mathbb{N}^K, p \in \mathbb{N}^{M+1}$ angegeben werden mit:

v Einträge $\neq 0$ von A, oben nach unten, links nach rechts.

z Zeilennummern der Einträge in v.

p_i Welcher Index j hat Eintrag a_{ki} in Spalte i? $(0 \leq k \leq N)$

Beispiel:

$$A = \begin{pmatrix} 1.1 & 0 & 0 & 2 \\ 3 & 4 & 0 & 5 \\ 0 & 6 & 0 & 0 \\ 0 & 7 & 0 & 8 \end{pmatrix}$$

$$v = (1.1, 3, 4, 6, 7, 2, 5, 8)^T$$

$$z = (1, 2, 2, 3, 4, 1, 2, 4)^T$$

$$p = (1, 3, 6, 6, 9)^T$$

2 Nichtlineare Gleichungssysteme

Problem: Finde ein $x \in D$ für $f(x) = 0$ mit $f : D \subset \mathbb{R}^n \to \mathbb{R}^n$.

2.1 Fixpunktiteration

Problem: Finde ein $x \in D$ für $\tilde{f}(x) = x$ mit $\tilde{f} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ (das sind die Fixpunkte). Das Problem ist äquivalent zum obigen Problem für $\tilde{f}(x) := x - Af(x)$ für eine

invertierbare Matrix $A \in \mathbb{R}^{n \times n}$.

Approximiere dann eine Lösung für $f(x) = 0$ durch $x_{k+1} := F(x_k)$. Diese Folge konvergiert zu dem (einzigen) Fixpunkt von $\tilde{f}$ für jeden Startwert $x_0 \in D$.

(Banach'scher Fixpunktsatz) Eine Abbildung $F : D \to D \subset \mathbb{R}^n$ ist eine Kontraktion auf D, falls $\exists \theta$ mit $0 \leq \theta < 1 : ||F(x) - F(y)|| \leq \theta ||x - y|| \quad \forall x, y \in D$ (oder: $\forall x \in D : ||F'(x)|| < 1$). Wenn D eine abgeschlossene Teilmenge von D ist, existiert genau ein Fixpunkt x^*, und folgende Abschätzungen gelten:

$$||x^* - x_k|| \leq \theta ||x^* - x_{k-1}|| \qquad \text{(lineare Konvergenz)}$$

$$||x^* - x_k|| \leq \frac{\theta^k}{1-\theta} ||x_0 - x_1|| \qquad \text{(A-priori Abschätzung)}$$

$$||x^* - x_k|| \leq \frac{\theta}{1-\theta} ||x_{k-1} - x_k|| \qquad \text{(A-posteriori Abschätzung)}$$

2.2 Newton-Verfahren

Gesucht: Lösung eines Nullstellenproblems $f(x) \stackrel{!}{=} 0$.

Vorgehensweise:

1. Wähle Startwert $x_0 \in D$.
2. while $(||\Delta x_k||) > TOL$):
   - Löse $f'(x_k)\Delta x_k = -f(x_k)$.
   - Berechne $x_{k+1} = x_k + \Delta x_k$.

Dabei sind

- f und f' Matrizen, f' kann man zuvor als Jakobi-Matrix berechnen. Für die Berechnung kann man f' LR-zerlegen.
- Δx_k das Ergebnis aus jeder Iteration und beschreibt den verbleibenden Fehler. Das Endergebnis des Algorithmus ist allerdings x_n.
- TOL die Toleranz für den Fehler.

Das Newton-Verfahren hat lokale quadratische Konvergenz.

Für das vereinfachte Newton-Verfahren verwendet man für $f'(x)$ eine konstante Matrix A.

3 cg-Verfahren

Problem: Löse $Ax = b$ für sehr große dünne Matrizen A. Statt die Gleichung direkt zu lösen, wird ein Funktional $\Phi(x)$ mit $\Phi(x) = \frac{1}{2}x^TAx - x^Tb$ minimiert. Das ist möglich, da gilt: x ist Lösung von $\Phi(x) = \min_{z\in\mathbb{R}^n} \Phi(z) \Leftrightarrow Ax = b$.

Für die Minimierung wird die Energienorm $||x||_A$ von spd-Matrizen $A \in \mathbb{R}^{n\times n}$ definiert durch $||x||_A = \sqrt{x^TAx}, x \in \mathbb{R}^n$. Das entspr. Skalarprodukt lautet $< x, y >_A = x^TAy$.

3.1 Algorithmus

1. Wähle $x_0 \in \mathbb{R}^n$ und definiere $d_0 = r^0 = b - Ax^0$ und $k = 0$.
2. Vorkonditioniert: Löse $Ms^0 = r^0$ und setze $d^0 = s^0$.
3. Solange $||r^k|| > TOL \cdot ||b||$:
   a) $\alpha_k = \frac{<r^k,r^k>}{<d^k,d^k>_A}$, Vorkonditioniert: $\alpha_k = \frac{<r^k,s^k>}{<d^k,d^k>_A}$
   b) $x^{k+1} = x^k + \alpha_k d^k$
   c) $r^{k+1} = r^k - \alpha_k Ad^k$
   d) Vorkonditioniert: löse $Ms^{(k+1)} = r^{(k+1)}$
   e) $\beta_k = \frac{<r^{k+1},r^{k+1}>}{<r^k,r^k>}$, Vorkonditioniert: $\beta_k = \frac{<r^{k+1},s^{k+1}>}{<r^k,s^k>}$
   f) $d^{k+1} = r^{k+1} + \beta_k d^k$, Vorkonditioniert: $d^{k+1} = s^{k+1} + \beta_k d^k$
   g) Erhöhe k um 1.

Der meiste Aufwand kommt durch iterative Berechnung von Ad^k, der Rest läuft linear in n. Bei dünn besetzten Matrizen kann ersteres effizient berechnet werden.

Falls cg eine Langsame Konvergenz aufzeigt, kann $Ax = b$ zu $\overbrace{C^{-1}AC^{-T}}^{=:\tilde{A}}\overbrace{C^Tx}^{=:\tilde{x}} = C^{-1}b =: \tilde{b}$ manipuliert werden, dann wird der Algorithmus entsprechend abgeändert.

3.2 Fehlerabschätzung

Für eine berechnete Approximation x^* gelten folgende Fehlerabschätzungen:

$$||x^* - x^k||_A \leq \max_{\lambda \text{ EW von } A} |q_k(\lambda)| \cdot ||x^* - x^0||_A \quad \forall \quad q_k \text{ Polynom vom Grad } \leq k, q(0) = 1$$

$$||x^* - x^k||_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k ||x^* - x^0||_A \qquad \text{mit } \kappa = cond_2(A)$$

Für die Konditionszahl einer spd Matrix in der 2-Norm gilt $cond_2(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$, wobei λ entsprechend der größte/kleinste EW von A bezeichnet.

4 Interpolation und Approximation

Interpolation: Suche für Stützpunkte (x_i, f_i) ein Polynom $p(x)$ vom Grad $\leq n$ mit $p(x_i) = f_i$.

Approximation: Suche für $f : [a, b] \to \mathbb{R}$ eine möglichst einfach auszuwertende Funktion $p : [a, b] \to \mathbb{R}$ mit $f - p = min$.

Weierstraßscher Approximationssatz: $\forall f : [a, b] \to \mathbb{R}$ stetig $\quad \forall \epsilon > 0 \quad \exists n \in \mathbb{N} \quad \wedge$ Polynom $p : [a, b] \to \mathbb{R}$ vom Grad n mit $\max_{x \in [a,b]} |f(x) - p(x)| \leq \epsilon$ (Also für jede stetige Funktion existiert ein approximierendes Polynom).

4.1 Lagrangesche Interpolationsformel

Zu $n + 1$ Stützpunkte (x_i, f_i) für $i = 0, ..., n$ mit paarweise verschiedenen Stützstellen x_i existiert genau ein Interpolationspolynom $p(x) := \sum_{i=0}^{n} f_i L_i(x)$ vom Grad $\leq n$.

Dabei sind $L_i(x) := \prod_{j=0, j \neq i}^{n} \frac{x - x_j}{x_i - x_j}$ die Lagrange-Polynome der Stützstellen x_i.

Weiter ist $\Lambda_n := \max_{x \in [a,b]} \sum_{i=0}^{n} |L_i(x)|$ die Lebesgue-Konstante bezüglich der Stützstellen x_i auf $[a, b]$ und invariant unter affinen Transformationen und damit nur von der relativen Lage der Stützstellen x_i zueinander abhängig.

Für zwei Interpolationspolynome $p(x), \tilde{p}(x)$ zu den Stützstellen (x_i, f_i) bzw. $(x_i, \tilde{f}_i), i = 0, ..., n$ gilt: $\max\limits_{x\in[a,b]} |p(x) - \tilde{p}(x)| \leq \Lambda_n \cdot \max\limits_{i=0,...,n} |f_i - \tilde{f}_i|$, hierfür ist Λ_n bereits die minimale Konstate.

4.2 Newtonsche Interpolationsformel

Gegeben sind $n+1$ Stützpunkte (x_i, f_i), $i = 0, ..., n$ mit paarweise verschiedenen Stützstellen x_i, dann existiert genau ein Interpolationspolynom $p(x) := f_{0,0} + f_{0,1} \cdot (x - x_0) + ... + f_{0,n} \cdot (x - x_0) \cdots (x - x_{n-1})$.

Dabei sind die dividierten Differenzen gegeben durch $f_{i,i} = f_i$ und $f_{i,i+k} = \frac{f_{i,i+k-1} - f_{i+1,i+k}}{x_i - x_{i+k}}$ für $i = 0, ..., n$ und $1 \leq k \leq n - i$.

Der Gesamtaufwand beträgt $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ Divisionen und $2\sum_{i=1}^{n} i = n(n+1)$ Additionen.

4.3 Interpolationsfehler

Sei $f : [a, b] \to \mathbb{R}$ mindestens $(n+1)$ mal stetig differenzierbar, $p(x)$ das zugehörige Interpolationspolynom vom Grad $\leq n$ an den Stützstellen $x_0, ..., x_n \in [a, b]$. Dann existiert zu jedem $x \in [a, b]$ eine Zwischenstelle $\xi(x) \in (a, b)$ mit: $f(x) - p(x) = w_{n+1} \cdot \frac{f^{(n+1)}(\xi)}{(n+1)!}$.

Die Elemente der Newton-Basis sind dabei definiert als $w_i(x) = \prod\limits_{j=0}^{i-1} (x - x_j)$.

$\max |w_{m+1}(x)|$ wird dabei minimal, wenn x_i genau die Nullstellen des $(n+1)$ten Tschebyscheff-Polynoms T_{n+1} sind, also $x_k = \cos\left(\frac{2k+1}{2n+2}\pi\right)$, der minimale Wert hierfür ist $\max |w_{m+1}(x)| = 2^{-n}$.

Die Tschebyscheff-Polynome sind dabei definiert durch $T_0(x) = 1, T_1(x) = x, T_{n+1} = 2x \cdot T_n(x) - T_{n-1}(x)$.

4.4 Tschebyscheffsche Interpolationsformel

Zu $n+1$ Stützpunkten (x_i, f_i), deren Stützstellen x_i genau den NSTen von T_{n+1} entsprechen, lässt sich das eindeutige Interpolationspolynom $p(x) := \frac{1}{2}c_0 + c_1 T_1(x) + ... + c_n T_n(x)$ darstellen mit $c_k := \frac{2}{n+1} \sum\limits_{i=0}^{n} f_i \cos\left(k \frac{2i+1}{2n+2}\pi\right)$.

Ein Polynom p obiger Form lässt sich durch den Clenshaw-Algorithmus bestimmen: Für $d_{n+2} = d_{n+1} = 0$ und $d_k = c_k + 2x \cdot d_{k+1} - d_{k+2}$ für $k = n, n-1, ..., 0$ gilt $p(x) = \frac{1}{2}(d_0 - d_2)$.

4.5 Spline-Interpolation

4.5.1 Kubische Spline-Interpolation

Problem: Suche eine glatte Funktion $s : [a, b] \to \mathbb{R}$, die durch Punkte (x_i, y_i) (x_i total geordnet und zwischen a und b) geht. Es soll also gelten: $s(x_i) = y_i$ und $\int_a^b |s''(x)|^2 dx$ minimal.

Sei nun $f, s : [a, b] \to \mathbb{R}$ beide zweimal stetig differenzierbar mit $s(x_i) = f(x_i)$, und gelte eine der folgenden Bedinungen:

1. $s'(a) = f'(a), s'(b) = f'(b)$ (Also Steigung an Intervallsgrenzen ist gleich)
   $\Rightarrow s$ heißt eingespannter interpolierender kubischer Spline.
2. $s''(a) = s''(b) = 0$ (Also an Intervallsgrenzen ist die Kurve eine Gerade)
   $\Rightarrow s$ heißt natürlicher interpolierender kubischer Spline.
3. $s^{(k)}(a) = s^{(k)}(b)$ für $k = 1, 2$ und $f^{(k)}(a) = f^{(k)}(b)$ für $k = 0, 1$.
   $\Rightarrow s$ heißt periodischer interpolierender kubischer Spline.

4.5.2 Konstruktion

Man will zu Stützstellen $(x_i, y_i), i = 0, ..., n$ ($a = x_0 < ... < x_n = b$) einen kubischen interpolierenden Spline $s(x)$ konstruieren. Dieser wird in Teilpolynome $s_i(x) := s|_{[x_{i-1}, x_i]}$ aufgeteilt.

1. Löse $\frac{1}{6}\begin{pmatrix} 2h_1 & h_1 & & & \\ h_1 & 2(h_1+h_2) & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-1} & 2(h_{n-1}+h_n) & h_n \\ & & & h_n & 2h_n \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \vdots \\ \gamma_n \end{pmatrix} = \begin{pmatrix} d_0 \\ \vdots \\ d_n \end{pmatrix}$ um $(\gamma_0, ..., \gamma_n)$ zu bestimmen, wobei $d_i := y_{i,i+1} - y_{i-1,i}$ und $y_{i-1,i} = \frac{y_{i-1} - y_i}{x_{i-1} - x_i}$.

Bei äquidistanten Gitter ($h = h_i$ lautet die Matrix $\frac{h}{6}\begin{pmatrix} 2 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 2 \end{pmatrix}$.

2. Bestimme α_i und β_i durch

$$\alpha_i = \frac{1}{6h_i}(\gamma_{i-1} + 2\gamma_i)$$
$$\beta_i = -\frac{1}{6h_i}(2\gamma_{i-1} + \gamma_i)$$

3. Bestimme die Interpolationspolynome durch

$$s_i(x) = y_{i-1} + (x - x_{i-1})y_{i-1,i} + (x - x_{i-1})(x - x_i)[(\alpha_i(x - x_{i-1}) + \beta_i(x - x_i)]$$

4.6 Bezier-Interpolation

4.6.1 Bernstein-Polynome

Das i-te Bernsteinpolynom vom Grad n bezüglich $[0, 1]$ ist $B_i^n(x) = \binom{n}{i}(1 - x)^{n-i}x^i$.

Für $x \in [0, 1]$ gilt: $B_i^n(x) \in [0, 1]$ (nicht negativ), $B_i^n(x) = B_{n-i}^n(1 - x)$ (symmetrisch), $1 = \sum_{i=0}^n B_i^n(x)$ für $x \in \mathbb{R}$ (Partition der Eins), B_i^n hat in $[0, 1]$ sein Maximum bei i/n.

Für eine Polynom $p(x) = \sum_{i=0}^n b_i B_i^n(x)$ heißen $b_i \in \mathbb{R}^d$ Kontrollpunkte von p, der durch sie verlaufende Streckenzug heißt Bézier-Polygon.

Das Bild von $p(x)$ liegt dabei in der konvexen Hülle der Kontrollpunkte b_i.

4.6.2 Algorithmus von Casteljau zur Überführung in Bernstein-Darstellung

Für $p(x) = \sum_{i=0}^n b_i B_i^n(x)$ sind wie folgt die Teilpolynome β_i^k vom Grad k definiert:

$$\beta_i^k(x) = \sum_{j=0}^{k} b_{i+j} B_j^k(x) = (1 - x)\beta_i^{k-1}(x) + x\beta_{i+1}^{k-1}(x)$$
$$\text{für } k = 0, ..., n \text{ und } i = 0, ..., n - k$$

Dann gilt $\beta_0^n(x) = p(x)$ (Wenn man also bis n iteriert, hat man die Bernsteindarstellung von p) und $\beta_i^0 \equiv b_i$ (Das i-te Teilpolynom 0-ten Grades entspricht also dem i-ten Kontrollpunkt b_i).

4.6.3 Bernsteinpolynome zu beliebigen Intervallen

Das i-te Bernsteinpolynom ($i = 0, ..., n$) vom Grad n bzgl. dem Intervall $[a, b]$ lautet $B_i^n(x; a, b) = \frac{1}{(b-a)^n} \binom{n}{i} (b - x)^{n-i}(x - a)^i$.

Wenn p gegeben ist durch $p(x) = \sum_{i=0}^n b_i B_i^n(x; a, b)$, lassen sich alternative Darstellungen $p(x) = \sum_{i=0}^n a_i B_i^n(x; a, \xi) = \sum_{i=0}^n \xi_i B_i^n(x; \xi, b)$ (bzgl. $[a, \xi]$ bzw. $[\xi, b]$ für $\xi \in]a, b[$) berechnen durch $a_k = \beta_0^k(x; a, b), \xi_k = \beta_k^{n-k}(c; a, b)$ für $k = 0, ..., n$.

Bei wiederholter Intervallteilung konvergieren die zu den Kontrollpunkten gehörenden Bézier-Polygone gegen die Kurve.

5 Numerische Integration

5.1 Simple Ansätze

Problem: Berechne für $f : [a, b] \to \mathbb{R}$ durch numerische Näherung $I(f) := \int_a^b f(x)dx$.

Rechteckregel: $I(f) \approx (b - a)f(a)$ $(s = 1, p = 1)$

Mittelpunktregel: $I(f) \approx (b - a)f(\frac{a+b}{2})$ $(s = 1, p = 2)$

Trapezregel: $I(f) \approx (b - a)\frac{f(a)+f(b)}{2}$ $(s = 2, p = 2)$

Simpsonregel: $I(f) \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$ $(p = 4, s = 3)$

($s = \#$Stützstellen, $p =$ Ordnung)

5.2 Kondition

Für stetige $f, \tilde{f} : [a, b] \to \mathbb{R}$ gilt: $\frac{|I(f)-I(\tilde{f})|}{|I(f)|} \leq \text{cond}_1 \frac{||f-\tilde{f}||_1}{||f||_1}$ mit $\text{cond}_1 := \frac{I(|f|)}{|I(f)|}$.

5.3 Quadratur

Die allgemeine Quadraturformel ist gegeben durch $\int_a^b f(x)dx \approx (b-a) \overbrace{\sum_{i=0}^{s} b_i f(\underbrace{a + c_i(b-a)}_{\text{Stützstelle}})}^{\substack{\text{gewichtetes Mittel der Funktions-} \\ \text{werte an den Stützstellen}}}$.

Dabei sind b_i die Gewichte und $c_i \in [a, b]$ die Knoten der Quadraturformel, s beschreibt die Anzahl der Knoten.

5.3.1 Ordnung der Quadraturformel

Eine Quadraturformel $(b_i, c_i)_{i=1,...,s}$ hat die Ordnung p:

- $\Leftrightarrow$ Sie liefert exakte Lösungen für alle Polynome vom Grad $\leq p-1$, wobei p maximal ist.
- $\Leftrightarrow$ $\frac{1}{q} = \sum_{i=0}^{s} b_i c_i^{q-1} \left(= \int_0^1 x^{q-1} dx \right)$ für alle $q = 1, ..., p$.

Die Ordnung charakterisiert die Approximationsgüte der Formel.

5.3.2 Eindeutigkeit der Ordnung der Quadraturformel

Für Knoten $c_1 < ... < c_s$ und eine geforderte Ordnung von s sind die Gewichte b_i und damit die Quadraturformel eindeutig gegeben durch:

$$\underbrace{\begin{pmatrix} c_1^0 & c_2^0 & \cdots & c_s^0 \\ c_1^1 & c_2^1 & \cdots & c_s^1 \\ \vdots & \vdots & & \vdots \\ c_1^{s-1} & c_2^{s-1} & \cdots & c_s^{s-1} \end{pmatrix}}_{=:C} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \vdots \\ \frac{1}{s} \end{pmatrix}$$

Die Vandermonde-Matrix C ist genau dann invertierbar wenn c_i paarweise verschieden sind.

Für $\mathcal{T} := (1, \frac{1}{2}, ..., \frac{1}{s})^T$ ergibt sich $b = C^{-1}\mathcal{T}$ und $b_i = \int_0^1 L_i(x)dx$.

5.3.3 Symmetrische Quadraturformeln

Eine Quadraturformel heißt symmetrisch, falls $c_i = 1 - c_{s+1-i}$ (Knoten) und $b_i = b_{s+1-i}$ (Gewichte). Die Ordnung einer symm. Quadraturformel ist immer gerade.

Wenn s symmetrische Knoten $c_1 < ... < c_s$ vorgegeben sind und für eine Quadraturformel min. Ordnung s gefordert wird, so sind die eindeutigen Gewichte auch symmetrisch.

5.3.4 Quadraturformeln höherer Ordnungen

Die maximale Ordnung für Quadraturformeln beträgt $2s$. Für eine Quadraturformel der Ordnung p mit $s \leq p \leq 2s$, beträgt die Ordnung genau $p = s+m$ wenn $\int_0^1 M(x)g(x)dx = 0$ für $M(x) = (x - c_1)(x - c_2) \cdots (x - c_s)$ für alle Polynome g vom Grad $< m$.

Quadraturformeln der Ordnung $2s$ sind eindeutig gegeben durch $c_i = \frac{1}{2}(1+\gamma_i), i = 1, ..., s$, wobei γ_i die Nullstellen des Legendre-Polynoms vom Grad s sind. Diese Quadraturformeln heißen Gauß-Quadraturformeln.

5.3.5 Quadraturfehler

Für eine Quadraturformel der Ordnung p und eine $q-$mal stetig differenzierbare Funktion $g : [0, 1] \to \mathbb{R}$ gilt $R(g) = \int_0^1 K_q(t)g^{(q)}(t)dt$ falls $2 \leq q \leq p$ wobei der Peano-Kern gegeben ist durch $K_q(t) = \frac{(1-t)^q}{q!} - \sum_{i=1}^{s} b_i \frac{(c_i-t)_+^{q-1}}{(q-1)!}$. Dabei gilt $(x-t)_+^n = (x-t)^n \Leftrightarrow x > t$, sonst $(x-t)^n = 0$.

TODO: $R(f; a, b)$ und dessen Herleitung, siehe Skript 2015 s. 110 und Vorlesungsmitschrieb 2017 #13 letzte Seite.

6 Eigenwertprobleme