Cheatsheet für 🅱️umerik

Paul Brinkmeier

12. Februar 2020

Inhaltsverzeichnis

1 Interpolation und Approximation

Weiherstraßscher Approximationssatz Sei $f : [a, b] \to \mathbb{R}$ stetig. Dann gilt:

$$\forall \epsilon > 0 : \exists N \in \mathbb{N}, p \in \mathbb{P}^N : \max_{x \in [a,b]} |f(x) - p(x)| \leq \epsilon \tag{1}$$

1.1 Polynominterpolation

Lagrange-Polynome Das n-te Lagrange-Polynom zu $N+1$ Stützwerten $f_n(x) = f(x_n)$ ist gegeben durch:

$$L_n(x) = \prod_{m=0, m \neq n}^{N} \frac{x - x_m}{x_n - x_m} \tag{2}$$

$f(x)$ lässt sich dann interpolieren durch $p(x)$ mit:

$$p(x) = \sum_{n=0}^{N} f_n L_n(x) \tag{3}$$

- Für $f_n = 0$ muss man L_n also gar nicht berechnen.
- Ein Lagrange-Polynom hat immer den Grad $N - 1$.
- Lagrange-Interpolation ist für hohe N schlecht konditioniert und liegt außerdem in $O(n)$.

Newton-Darstellung

$$p_{0,N}(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + ... + a_N \prod_{m=0}^{N-1} x - x_m \tag{4}$$

- Lässt einfach durch das Horner-Schema auswerten.
- $p_{0,N}(x) = p_{0,N-1}(x) + a_N w_N(x)$ mit

$$w_n(x) = \prod_{m=0}^{n-1} (x - x_m) \tag{5}$$

Naive Koeffizientenbestimmung

$$f_0 = p(x_0) = a_0 \tag{6}$$
$$f_1 = p(x_1) = a_0 + (x_1 - x_0)a_1 \tag{7}$$
$$f_2 = p(x_2) = a_0 + (x_2 - x_0)a_1 + (x_2 - x_1)a_2 \tag{8}$$

- Problem: $2n$ Additionen, $2(n-1)$ Multiplikationen, 1 Division für a_n.
- $\rightsquigarrow$ Insgesamt $N(N+1)$ Additionen, $N(N-1)$ Multiplikationen, N Divisionen.

Lemma von Aitken

$$p_{n,k}(x) = \frac{(x_n - x)p_{n+1,k}(x) - (x_k - x)p_{n,k-1}(x)}{x_n - x_k} \tag{9}$$

Dividierte Differenzen

$$f_{n,k} = \begin{cases} f_n & n = k \\ \frac{f_{n,k-1} - f_{n+1,k}}{x_n - x_k} & \text{sonst} \end{cases} \tag{10}$$

- $\frac{N(N+1)}{2}$ Divisionen
- $N(N+1)$ Additionen
- $p(x) = \sum_{m=0}^{N} f_{0,m} w_m(x)$

Lebesgue-Konstante Bezüglich der Stützstellen $x_0, ..., X_N$:

$$\Lambda_N = \max \sum_{x \in [a,b]}^{N} |L_n(x)| \tag{11}$$

- Λ_N ist nur von der relativen Lage der Stützstellen zueinander abhängig.

Interpolationsfehler Der maximale Interpolationsfehler eines Interpolationspolynoms $p(x)$ zu $f(x)$ mit $N+1$ Stützstellen (x_0 bis x_N) ist:

$$\max_{x \in [a,b]} |f(x) - p(x)| \leq \max_{x \in [a,b]} |w_{N+1}(x)| \frac{\max_{x \in [a,b]} |f^{(N+1)}(x)|}{(N+1)!} \tag{12}$$

Optimale (Tschebyscheff-) Stützstellen Wählt man die $N+1$ Stützstellen x_n genau als die Nullstellen des Tschebyscheff-Polynoms $T_{N+1}(x)$, wobei

$$T_0(x) = 1 \tag{13}$$
$$T_1(x) = x \tag{14}$$
$$T_{n+1}(x) = 2x \cdot T_n(x) - T_{n-1}(x) \tag{15}$$
$$T_n(x) = \cos(n \cdot \arccos x) \tag{16}$$
$$T_n(\cos\left(\frac{2n+1}{2N+2}\pi\right)) = 0 \tag{17}$$

so wird $\max_{x \in [-1,1]} |w_{N+1}(x)|$ minimal, und zwar 2^{-N}. Mit mehr Stützstellen sinkt also der Fehler.

Es gilt:

- $\forall x \in \mathbb{R} : |T_N(x)| \leq 1$
- $\deg(T_N) = N$
- $\langle T_0, ..., T_N \rangle = \mathbb{P}^N$ (die T_n bilden 1 Basis von $\mathbb{P}^N$)
- T_N hat N verschiedene reelle Nullstellen x_n mit $x_n \in [-1, 1]$

Tschebyscheff-Darstellung

$$p(x) = \frac{c_0}{2} + \sum_{n=1}^{N} c_n T_n(x) \tag{18}$$

- Tschebyscheff-Darstellung ist eindeutig.
- Berechnung von c_n im Fall $N+1 = 2^k$: $O(N * logN)$

Clenshaw-Algorithmus Sei $p(x)$ gegeben als

$$p(x) = \frac{c_0}{2} + \sum_{n=1}^{N} c_n T_n(x) \tag{19}$$

Dann lässt sich $p(x)$ folgendermaßen an der Stelle x auswerten:

$$d_{N+2} = 0 \tag{20}$$
$$d_{N+1} = 0 \tag{21}$$
$$d_n = c_n + 2x d_{n+1} - d_{n+2} \tag{22}$$
$$p(x) = \frac{1}{2}(d_0 - d_2) \tag{23}$$

- $2N$ Additionen, $N + 2$ Multiplikationen

1.2 Kubische Spline-Interpolation

Seien (x_k, y_k) Stützstellen für $k \in \{0, ..., N\}$ mit

$$a = x_0 < x_1 < ... < x_N = b \tag{24}$$

Ziel: Finde $s(x)$ mit

- Interpolation: $s(x_n) = y_n$
- Glattheit: s zweimal stetig diff'bar und

$$\int_a^b |s''(b)|^2 \,\mathrm{d}x \tag{25}$$

minimal. Vorgehen:

$$s(x) = \begin{cases} s_1(x) & x \in [x_0, x_1] \\ s_2(x) & x \in [x_1, x_2] \\ & \vdots \\ s_N(x) & x \in [x_{N-1}, x_N] \end{cases} \tag{26}$$

Wobei aus 25 folgen muss:

$$s'_n(x_n) = s'_{n+1}(x_n) \tag{27}$$
$$s''_n(x_n) = s''_{n+1}(x_n) \tag{28}$$

2 Numerische Integration

2.1 Quadraturformeln

Allgemeine Darstellung

$$\int_a^b f(x) \,\mathrm{d}x \approx \sum_{k=1}^{s} b_k f(a + c_k(b - a)) \tag{29}$$

b_k nennt man Gewichte, c_k Knoten der Formel. Typischerweise: $b_k, c_k \in [0, 1]$.

- Rechteckregel: $s = 1, (1, 0)$

- Mittelpunktregel: $s = 1, (1, \frac{1}{2})$
- Trapezregel: $s = 2, (\frac{1}{2}, 0), (\frac{1}{2}, 1)$
- Simpsonregel: $s = 3, (\frac{1}{6}, 0), (\frac{4}{6}, \frac{1}{2}), (\frac{1}{6}, 1)$

Ein paar Fakten zu QF:

- Eine QF ist durch die Gewichte $b_1, ..., b_s$ und Knoten $c_1 < ... < c_s$ eindeutig bestimmt.

2.1.1 Ordnung von QF

Besitzt eine Quadraturformel die Ordnung p, so gilt liefert sie für alle $f \in \mathbb{P}_{p-1}$ den exakten Wert des Riemann-Integrals

$$\int_a^b f(x)\,\mathrm{d}x \tag{30}$$

Eine QF besitzt genau dann die Ordnung p, wenn für alle $q \in \{1, ..., p\} \subset \mathbb{N}$, aber nicht mehr für $q = p + 1$, gilt:

$$\sum_{k=1}^{s} b_k c_k^{q-1} = \frac{1}{q} \tag{31}$$

2.1.2 Symmetrische QF

Eine QF heißt symmetrisch, falls gilt:

$$c_k = 1 - c_{s+1-k} \tag{32}$$
$$b_k = b_{s+1-k} \tag{33}$$

D.h. die Knoten sind symmetrisch um $\frac{1}{2}$ angeordnet und die Gewichte sind symmetrisch verteilt.

- Symmetrische QF besitzen immer eine gerade Ordnung.

Gauß-Quadraturformeln Es existiert eine eindeutige Quadraturformel der Ordnung $2s$. Sie ist gegeben durch

$$c_k = \frac{1}{2}(1 + \gamma_k) \tag{34}$$

für $k \in \{1, ..., s\}$, wobei die γ_k die Nullstellen des Legendre-Polynoms vom Grad s sind. Die Gewichte sind dann eindeutig bestimmt durch

$$b_k = \int_0^1 L_k(x)\,\mathrm{d}x \tag{35}$$

Dabei ist L_k das k-te Legendre-Polynom, bspw.:

$$L_0(x) = 1 \tag{36}$$
$$L_1(x) = x \tag{37}$$
$$L_2(x) = \frac{1}{2}(3x^2 - 1) \tag{38}$$

- Gauß-Quadraturformeln sind monoton.

3 Eigenwertprobleme

Grundlegendes Problem: berechne Eigenwerte λ zu Eigenvektoren v einer quadratischen Matrix. Anwendungen: bspw. Eigenschwingung einer Membran, spektrale Bisektion.

EW-EV-Gleichung Sei $A \in \mathbb{R}^{N\times N}$. Dann heißen $v \in \mathbb{R}^{\mathbb{N}} \setminus \{0\}$ Eigenvektor und $\lambda \in \mathbb{R}$ dazugehöriger Eigenwert von A, gdw.:

$$Av = \lambda v \tag{39}$$

Die „Wirkungsweise“ von A auf v ist also einfach zu verstehen (Stauchung/Streckung).

- A hat EW $\Leftrightarrow$ A ist singulär $\Leftrightarrow$ $\exists \lambda \in \mathbb{R} : \det(A - \lambda I_N) = 0$.
- $\det(A - \lambda I_N)$ ist ein Polynom in λ, genannt charakteristisches Polynom von A.
- Aber: EW nicht wie bisher über c.P. berechnen, da schlecht konditioniert.

Konditionszahl Seien $u, v \in \mathbb{R}^N$ normiert, $Av = \lambda v$, $u^\top A = u^\top \lambda$ ($\Leftrightarrow A^\top u = \lambda u$). Dann heißt

$$\frac{1}{|u^\top v|} \tag{40}$$

die Konditionszahl von λ.

- $\frac{1}{|u^\top v|} \overset{\text{(CSU)}}{\geq} \frac{1}{\|u\|_2 \|v\|_2} = 1$

Merke: Für $A = A^\top$ gilt: $u \in \{v, -v\}$, d.h.

$$\frac{1}{|v^\top v|} = 1 \tag{41}$$

Symmetrische Matrizen sind also perfekt konditioniert.

3.1 Vektoriteration

3.1.1 Inverse Vektoriteration

$$Av = \lambda v \Leftrightarrow v = \lambda A^{-1} v \Leftrightarrow A^{-1} v = \frac{1}{\lambda} v \tag{42}$$

$\rightsquigarrow$ EV von A und A^{-1} sind identisch, $\operatorname{spec}(A^{-1}) = \frac{1}{\operatorname{spec}(A)}$.

4 Iterative Verfahren von LGS

Löse $Ax = b$, wobei

- N sehr groß,
- A dünn besetzt
- und/oder keine exakte Lösung notig ist.

(Beispielsweise spektrale Bisektion).

Direkte Verfahren (LR, Cholesky, QR) sind hier ungeeignet, da sie aufwändig sind und „fill-in“ aufweisen, d.h. die Zerlegungsmatrizen sind (fast) voll besetzt, selbst wenn A dünn besetzt ist.

Alternative: Iterative Verfahren.

4.1 Allgemeine lineare Iteration

Seien A, „Vorkonditionierer“ B regulär, idealerweise mit

$$\operatorname{cond}(B^{-1}A) \overset{\text{viel}}{<} \operatorname{cond}(A) \tag{43}$$

Dann gilt:

$$Ax = b \Leftrightarrow B^{-1}Ax = B^{-1}b \Leftrightarrow x = (I_N - B^{-1}A)x + B^{-1}b = x + B^{-1}(b - Ax) \tag{44}$$

Solche Gleichungen nennt man Fixpunktgleichungen.

Fixpunktiteration Eine solche Fixpunktgleichung kann durch eine Fixpunktiteration gelöst werden, wenn

$$\rho(I_N - B^{-1}A) < 1 \tag{45}$$

mit dem Spektralradius $\rho(M) = \max|\operatorname{spec}(M)|$. Ausgehend von einem Startwert $x^0 \in \mathbb{R}^N$ berechne für $k = 0, 1, ...$:

$$x^{k+1} = x^k + B^{-1}r^k \tag{46}$$

mit dem Residuum $r^k = b - Ax^k$. Nach der Bedingung ist A regulär und die Fixpunktiteration konvergiert linear gegen eine eindeutige Lösung x_* von $Ax = b$.

4.2 Beispiele für iterative Verfahren

Zerlege $A = L + D + R$, wobei L eine *strikte* untere Dreiecksmatrix, R eine *strikte* obere Dreiecksmatrix und entsprechend D eine Diagonalmatrix ist.

4.2.1 Jacobi-Verfahren

Wähle $B = D$.

- Auch Gesamtschrittverfahren genannt.
- Einfach parallelisierbar wegen komponentenweiser Multiplikation.

4.2.2 Gauß-Seidel-Verfahren

Wähle $B = L + D$ (immer noch untere Dreicksmatrix).

- Auch Einzelschrittverfahren.
- Einfach nacheinander berechenbar, aber nicht parallelisierbar.

4.2.3 Unvollständige LR-Zerlegung

Wähle $B = M \cdot S$, wobei $A \approx MS$ „unvollst. LR-Zerlegung“. Die unvollständige LR-Zerlegung ($A_{j,k} = 0 \Rightarrow (MS)_{j,k} = 0$) vermeidet Fill-In, liefert aber kein genaues Resultat.

4.3 Konvergenz von Jacobi- und Gauß-Seidel-Verfahren

Ist A strikt diagonaldominant, so konvergieren sowohl das Jacobi- als auch das Gauß-Seidel-Verfahren für jeden Startwert $x^0 \in \mathbb{R}^N$ gegen die Lösung x_* von $Ax = b$.

In der Regel konvergiert das Gauß-Seidel-Verfahren schneller als das Jacobi-Verfahren (es ist aber nicht so gut parallelisierbar).

4.4 Algorithmus

1. Wähle $x^0 \in \mathbb{R}^N, \epsilon > 0$ und setze $r^0 = b - Ax^0, k = 0$.
2. Löse $Bc^k = r^k$ nach c^k.
   - $x^{k+1} = x^k + c^k$
   - $r^{k+1} = r^k - Ac^k$
3. Falls $\left\| r^k \right\| < \epsilon \left\| b \right\|$, STOP. Ansonsten erhöhe k und gehe zu Schritt 2.

Merke: Das LGS in Schritt 2 muss leichter zu Lösen sein als $Ax = b$, bspw. ist es trivial beim Jacobi-Verfahren, da B eine Diagonalmatrix ist und einfach beim Gauß-Seidel-Verfahren durch Vorwärtssubstitution.

5 Grundwissen

5.1 Matrizen

Im folgenden: $A \in \mathbb{R}^{N \times N}$.

5.1.1 Symmetrische Matrizen

$$\text{„}A \text{ ist symmetrisch“} \tag{47}$$
$$\Leftrightarrow A = A^\top \tag{48}$$

5.1.2 Reguläre Matrizen

$$\text{„}A \text{ ist regulär“} \tag{49}$$
$$\Leftrightarrow \text{„}A \text{ ist invertierbar“} \tag{50}$$
$$\Leftrightarrow \exists A^{-1} : AA^{-1} = A^{-1}A = I_N \tag{51}$$

5.1.3 Orthogonale Matrizen

5.1.4 Matrixnormen

Spaltensummennorm Größte betragliche Summe einer Spalte:

$$\left\| A \right\|_1 = \max_{m=1,\ldots,N} \sum_{n=1}^{N} |a_{nm}| \tag{52}$$

Zeilensummennorm Größte betragliche Summe einer Zeile:

$$\left\| A \right\|_\infty = \max_{n=1,\ldots,N} \sum_{m=1}^{N} |a_{nm}| \tag{53}$$

Spektralnorm Wurzel des größten EW von $A^\top A$:

$$\left\| A \right\|_2 = \sqrt{\max \operatorname{spec}(A^\top A)} \tag{54}$$

5.1.5 Eigenwerte

Seien $v \in \mathbb{R}^N, \lambda \in \mathbb{R}$. v heißt Eigenvektor zum Eigenwert λ genau dann wenn:

$$Av = \lambda v \tag{55}$$

A^{-1} hat die EW $\frac{1}{\lambda}$ zu denselben EV v:

$$Av = \lambda v \Leftrightarrow v = \lambda A^{-1} v \Leftrightarrow A^{-1} v = \frac{1}{\lambda} v \tag{56}$$

$A - \lambda^* I_N$ hat die EW $\lambda - \lambda^*$ zu denselben EV v:

$$(A - \lambda^* I_N)v = Av - \lambda^* I_N v = \lambda v - \lambda^* v = (\lambda - \lambda^*)v \tag{57}$$