

Wichtige Formeln für Numerik

Sebastian Reichelt

28. Juli 2004

Dies ist eine Zusammenstellung von Formeln und Verfahren, die man zum Bestehen einer *bestimmten* Klausur „Numerische Mathematik für Informatiker und Ingenieure“ von Prof. Dr. R. Scherer benötigt. Sie ist bewusst kurz und *unvollständig* gehalten. Es sei auch darauf hingewiesen, dass man eine ähnliche Klausur mit Sicherheit *nicht* bestehen kann, ohne diese Formeln zu kennen, dass man sie aber nicht unbedingt besteht, wenn man *nur* diese Formeln kennt.

Jegliche Haftung für den Inhalt dieser Zusammenstellung ist ausgeschlossen. Bei Fehlern bitte Mail an SebastianR@gmx.de.

1 Fixpunktiteration

Fixpunktiteration: $x_{n+1} = \varphi(x_n)$ mit $x_i \in \mathbb{R}$. (φ liest sich direkt aus Aufgabenstellung ab.)

Gilt analog auch im \mathbb{R}^n .

1.1 Konvergenzkriterium

Suche abgeschlossenes Intervall I , so dass $\varphi(I) \subseteq I$ und $L_{\min} = \sup_{x \in I} |\varphi'(x)| < 1$. Zeige dann, dass alle relevanten Startwerte in dem Intervall landen.

1.2 Fehlerabschätzung

a-priori: $|x_n - \tilde{x}| \leq \frac{L^n}{1-L} \cdot |x_1 - x_0|$ (\tilde{x} ist der eigentliche Fixpunkt; x_1 aus Startwert x_0 berechnen)

a-posteriori: $|x_n - \tilde{x}| \leq \frac{L}{1-L} \cdot |x_n - x_{n-1}|$

2 Matrizen und Lineare Gleichungssysteme

Aussagen über Matrix A ; Lösung der Gleichung $Ax = b$.

2.1 Kreissatz von Gerschgorin

Für Zeile j sei r_j die Summe der Beträge der Elemente außer a_{jj} , also $r_j = \sum_{k \in \{1, \dots, n\} \setminus \{j\}} |a_{jk}|$. Dann gilt für alle Eigenwerte $\lambda_i \in \mathbb{C}$ von A : $\exists j \in \{1, \dots, n\} : |\lambda_i - a_{jj}| \leq r_j$.

2.1.1 Anwendung

Falls alle Eigenwerte reell sind (z.B. A symmetrisch), bestimme Intervall(e) für die Eigenwerte wie folgt:

1. Berechne die r_j für jede Zeile.
2. Berechne Grenzen für Intervall $[a_{jj} - r_j, a_{jj} + r_j]$.
3. Bilde Vereinigung aller Intervalle.

2.2 Gesamt- und Einzelschrittverfahren

Fixpunktiteration (1) mit $\varphi(x) = Tx + g$, wobei es für T und g zwei Möglichkeiten gibt (s.u.). Verfahren konvergiert genau dann, wenn alle Eigenwerte von T betragsmäßig kleiner als 1 sind ($\rho(T) < 1$). Hinreichend ist auch $N(T) < 1$ für beliebige Norm N .

2.2.1 Gesamtschrittverfahren

Teile A auf in Diagonale D und Rest R_G . D^{-1} ist einfach zu berechnen: Jedes Diagonalelement invertieren.

$$T_G = -D^{-1} \cdot R_G, g_G = D^{-1} \cdot b$$

Zeilensummenkriterium: r_j wie im Kreissatz (2.1). Falls $r_j < |a_{jj}| \forall j \in \{1, \dots, n\}$, konvergiert das Gesamtschrittverfahren auf jeden Fall.

Spaltensummenkriterium: Analog für jede Spalte.

2.2.2 Einzelschrittverfahren

Teile A auf in untere Dreiecksmatrix B und Rest R_E .

$$T_E = -B^{-1} \cdot R_E, g_E = B^{-1} \cdot b$$

T_E lässt sich mit Formel für einzelne Komponenten direkt berechnen, aber in der Klausur darf man B invertieren.

Nur Zeilensummenkriterium möglich; im Zweifelsfall auf allgemeines Kriterium (Norm, Spektralradius) zurückgreifen.

2.2.3 Anmerkung

Im Skript wird A direkt in Diagonale D , untere Hälfte L (lower?) und obere Hälfte U (upper?) zerlegt. Schreibe also $L + U$ statt R_G , $L + D$ statt B , U statt R_E .

2.3 LR-Zerlegung

2.3.1 Zerlegung einer Matrix

Zerlege A in Produkt von bestimmter unterer und oberer Dreiecksmatrix. Z.B. für $n = 3$:

$$A = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix}}_{=:L} \cdot \underbrace{\begin{pmatrix} r_{11} & r_{21} & r_{31} \\ 0 & r_{22} & r_{32} \\ 0 & 0 & r_{33} \end{pmatrix}}_{=:R}$$

Bestimme l_{jk} und r_{jk} mittels normaler Matrixmultiplikation. Dies ist einfach, wegen der vielen Nullen. Fange oben links an (Multiplikation der ersten Zeile von L mit der ersten Zeile von R). Mache dann nach unten oder rechts weiter. Man bekommt für jede Komponente von A eine Gleichung für ein l_{jk} oder r_{jk} .

2.3.2 Lösen eines LGS

Löse statt $Ax = b$ zunächst $Ly = b$, d.h. man erhält einen neuen Vektor y . Dies ist wegen der Diagonalform von L sehr einfach. Löse dann $Rx = y$ auf die gleiche Art.

2.4 Cholesky-Zerlegung

$A = L \cdot L^T$ mit unterer Dreiecksmatrix L . Berechnung analog zur LR-Zerlegung (2.3). Existiert genau dann, wenn A symmetrisch und positiv definit ist.

3 Polynome

Wichtige Schreibweise: \mathcal{P}_n : Menge (Vektorraum) der Polynome vom Grad $\leq n$.

3.1 HornerSchema

- Bestimmung der Werte aller Ableitungen an einem bestimmten Punkt x_0 (inklusive Funktion selbst)
- Abspalten eines Faktors $x - x_0$
- Entwicklung nach Potenzen davon (vollständiges HornerSchema)
- Abspalten eines quadratischen Faktors $x^2 - sx - t$ (mehrzeiliges HornerSchema)

Unbedingt üben! Generell wird ein Koeffizient immer nach rechts oben übertragen und dabei mit x_0 multipliziert; bei doppelzeiligem HornerSchema an die beiden symmetrischen Stellen (siehe Pfeile in Musterlösungen). Vorsicht bei Bestimmung der Werte der Ableitungen: Vorfaktor aus Taylor-Reihe nicht vergessen!

3.2 Sturmsche Kette

Bestimmung der Anzahl der Nullstellen eines Polynoms p in einem Intervall $[\alpha, \beta]$. Achtung: Nur einfache Nullstellen erlaubt!

Setze $p_0 := p$, $p_1 := p'$. Für alle weiteren p_{n+1} teile p_{n-1} durch p_n (Polynomdivision) mit Restpolynom r ; setze $p_{n+1} := -r$. Führe so lange fort, bis $r = 0$. Betrachte dann jeweils die gesamten $p_i(\alpha)$ und $p_i(\beta)$ und zähle die Vorzeichenwechsel w_α und w_β . Anzahl der Nullstellen in $[\alpha, \beta]$ ist $w_\alpha - w_\beta$ ($w_\alpha \geq w_\beta$ gilt immer!).

3.3 Polynomielle Interpolation

Punkte (x_i, y_i) ($i \in \{0, \dots, n\}$), evtl. Funktion f mit $y_i = f(x_i)$. Gesucht: Polynom $p \in \mathcal{P}_n$ durch alle Punkte (eindeutig!).

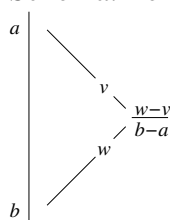
3.3.1 Lagrange

Grundpolynome für die Knoten x_i : $l_i(x) = \prod_{k \in \{0, \dots, n\} \setminus \{i\}} \frac{x - x_k}{x_i - x_k}$ (vorausberechenbar für gegebene x_i)

Dann: $p(x) = \sum_{i=0}^n y_i \cdot l_i(x)$ (also einfach Werte y_i einsetzen)

3.3.2 Newton

Schema: Zeichne senkrechte Wertetabelle. Erweitere dann nach rechts:



(für zwei benachbarte Werte v und w)

Dann sind die Werte in der obersten (schrägen) Zeile die Koeffizienten für die Grundpolynome

$w_i(x) = \prod_{k=0}^{i-1} (x - x_k)$, wobei i von 0 bis n läuft, wenn man die Zahlen von links nach rechts abliest.

Hinzufügen eines neuen Knotens am besten am unteren Rand. Beachte: Die Knoten müssen nicht in einer bestimmten Reihenfolge sein. In einer Klausuraufgabe sind dann meistens zwei verschiedene Interpolationspolynome gefordert, man erhält aber das zweite einfach durch Hinzufügen eines weiteren Terms.

3.3.3 Fehlerabschätzung

Wird eine $n + 1$ mal stetig differenzierbare Funktion $f : [a, b] \rightarrow \mathbb{R}$ polynomiell interpoliert, beträgt

der Fehler maximal $\frac{1}{(n+1)!} \cdot \max_{x \in (a,b)} |f^{(n+1)}(x)| \cdot \max_{x \in [a,b]} \underbrace{\left| \prod_{k=0}^n (x - x_k) \right|}_{=w_{n+1}(x)}$.

3.4 Splines

Definition (unbedingt auswendig lernen): Eine Funktion $f : [a, b] \rightarrow \mathbb{R}$ heißt Spline vom Grad n , wenn sie stückweise polynomiell (mit Polynomen in \mathcal{P}_n) und $n - 1$ mal stetig differenzierbar ist.

3.4.1 Kubische Spline-Interpolation

Ein Spline vom Grad 3 heißt kubisch (genau wie die einzelnen Polynome); dies ist der Standardfall. Für kubische Splines gibt es ein Interpolationsverfahren im Skript; ggf. üben. Wird damit eine 4-mal stetig differenzierbare Funktion f mit äquidistanten Knoten (Distanz h) interpoliert, ist der Fehler maximal $2 \cdot h^4 \cdot \max_{x \in [a,b]} |f''''(x)|$.

3.5 Bézier-Darstellung

3.5.1 Bernstein-Grundpolynome

Für ein Polynom vom Grad n benötigt man die Grundpolynome $b_{i,n}$, $i = 0, \dots, n$:

$$b_{i,n}(t) = \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i} = (1-t) \cdot b_{i,n-1}(t) + t \cdot b_{i-1,n-1}(t)$$

3.5.2 Berechnung der Koeffizienten

Koeffizienten β_i für Grundpolynome aus a_k des Ursprungspolynoms berechnen:

$$\beta_i = \sum_{k=0}^i \frac{\binom{i}{k}}{\binom{n}{k}} \cdot a_k \quad (\text{oder mit Koeffizientenvergleich})$$

Dies liefert die Bézier-Punkte $(\frac{i}{n}, \beta_i)$.

3.5.3 Auswertung (Algorithmus von de Casteljau)

Berechnung von $p(x)$: Schreibe alle Koeffizienten β_i untereinander, addiere dann jeweils zwei übereinanderliegende Zahlen zu einer neuen zusammen, mit der Gewichtung $(1-x) : x$.

3.6 Trigonometrische Polynome

Diskrete Fourier-Polynome, d.h. Teilsummen von Fourier-Reihen:

$$\tau(z) = \frac{\alpha_0}{2} + \sum_{k=1}^m (\alpha_k \cdot \cos(k \cdot z) + \beta_k \cdot \sin(k \cdot z))$$

3.6.1 Trigonometrische Interpolation

Knoten x_i ($i \in \{0, \dots, n\}$) müssen äquidistant sein mit $x_0 = 0$ und $x_{n+1} = 2\pi$.

$$m = \frac{n}{2}$$

$$\alpha_k = \frac{2}{n+1} \cdot \sum_{i=0}^n y_i \cdot \cos(k \cdot x_i) \quad (k \in \{0, \dots, m\})$$

$$\beta_k = \frac{2}{n+1} \cdot \sum_{i=0}^n y_i \cdot \sin(k \cdot x_i) \quad (k \in \{1, \dots, m\})$$

4 Integrale (Quadratur)

Näherungsformel für ein Integral von a bis b mit Hilfe von n Teilintervallen.

4.1 Trapez-Regel

Werte an den beiden Randpunkten werden halbiert; der Rest zählt normal. Funktionsauswertungen:

$$n + 1. \text{ Fehler: } \frac{1}{12} \cdot \frac{(b-a)^3}{n^2} \cdot \max_{x \in [a,b]} |f''(x)|.$$

4.1.1 Romberg-Verfahren

Führe mehrere Berechnungen der Trapezsummen aus, mit $n = 2^j$ Intervallen, $j = 0, 1, 2, \dots$ (Funktionsauswertungen wiederverwendbar). Kombiniere dann jeweils die Ergebnisse von zwei nebeneinanderliegenden j zu einem neuen, wobei das Ergebnis vom größeren j mit $4^k : (-1)$ gewichtet wird ($k = 1, 2, \dots$).

4.2 Mittelpunkt-Regel

Funktion nicht an den Teilintervallgrenzen auswerten, sondern in der Mitte. Funktionsauswertungen:

$$n. \text{ Fehler: } \frac{1}{24} \cdot \frac{(b-a)^3}{n^2} \cdot \max_{x \in [a,b]} |f''(x)|.$$

4.3 Kepler-Simpson-Regel

Interpolation der Teilstücke durch quadratische Polynome. Dafür Auswertung an Grenzen *und* Mittelpunkten nötig. Ergebnis: Randpunkte zählen $\frac{1}{6}$, Punkte an Grenzen zählen $\frac{1}{3}$, Mittelpunkte zählen $\frac{2}{3}$. Funktionsauswertungen: $2n + 1$. Fehler: $\frac{1}{2880} \cdot \frac{(b-a)^5}{n^4} \cdot \max_{x \in [a,b]} |f'''(x)|$.