Yet another

very 1337 essence of

N00–merik für Informatiker

Dirk Achenbach
<dirk.achenbach@stud.uni-karlsruhe.de>

Sonntag, 13. Februar 2005 20:47

http://www.stud.uni-karlsruhe.de/~uoavy/wahrheit/numerik.(pdf|ps|dvi)

Inhaltsverzeichnis

0.1. Computertomographie . . . 7

1. Numerik **8**
1.1. Nichtlineare Gleichungssysteme . . . 8
1.1.1. NEWTON–Verfahren . . . 8
1.1.2. HERON–Verfahren . . . 8
1.1.3. Fixpunktiteration — BANACHscher Fixpunktsatz . . . 9
1.2. Lineare Gleichungssysteme . . . 10
1.2.1. LR–Zerlegung . . . 10
1.2.2. CHOLESKY–Zerlegung . . . 13
1.2.3. Ausgleichsproblem . . . 15
1.3. Polynome und Splines . . . 16
1.3.1. GERSCHGORIN . . . 16
1.3.2. STURMsche Kette . . . 17
1.3.3. TSCHEBYSCHEFF–Polynome . . . 18
1.3.4. Splines . . . 20
1.4. Algebraische (Polynom–)Interpolation . . . 23
1.4.1. Normal (VANDERMONDE–Matrix) . . . 24
1.4.2. LAGRANGE–Darstellung . . . 24
1.4.3. NEWTON–Gestalt . . . 25
1.4.4. BÉZIER–Technologie . . . 25

2. Referenz **27**
2.0.5. Mittelwertsatz der Differentialrechnung (MWS) . . . 27
2.0.6. EUKLID–Algorithmus . . . 27
2.0.7. HORNER–Schema . . . 28
2.0.8. Definitheit und Symmetrie bei Matrizen . . . 30
2.0.9. Determinanten . . . 31
2.0.10. Metrik und Normen . . . 32

A. Anhang **35**

Hier steht natürlich irgendwann ein brilliantes Vorwort. Voerst nehmen wir aber mit dem folgenden Gelaber, dem Changelog und der TODO–Liste vorlieb:

Diese Kurzzusammenfassung soll als Lernhilfe auf die Numerik–Klausur für Informatiker dienen. Ich habe mich bemüht, die Zusammenhänge so darszustellen, dass die nachvollziehbar sind. Diese Zusammenfassung soll selbstverständlich keinesfalls den Besuch der Vorlesung oder die Bearbeitung der Übungsblätter ersetzen!

Die Inhalte dieser Zusammenfassung entstammen den im Bibliographieindex[1] angegebenen Quellen. Ich habe sie meinem Verständnis nach angepasst und verändert.

Ich habe alles so festgehalten, wie ich es verstehe. Es wird deutlich, dass ich selbstverständlich keine Garantie darauf geben kann, dass hier auch nur irgendetwas sachlich richtig ist. Ich freue mich über Hilfe bei der Veri– oder Falsifikation der Inhalte. Außerdem mag ich Pralinen.

Hier finden sich ein paar Themen, die nicht klausurrelevant sind. Es war mir ein Bedürfnis, sie festzuhalten, da sie einige Zusammenhänge deutlicher machen oder dem Allgemeinwissen zuträglich sind.

Viel Erfolg in der Klausur!

Kallsruh, 07.02.05.

Changelog

13.02.05 Struktur reingebracht, Tschebyscheff vervollständigt, Korrekturen bei Bézier, Splines, Index angelegt, Literaturverzeichnis gezaubert, Polynominterpolation gemacht, diverse Korrekturen

12.02.05 Tschebyscheff, Bézier (schwammig), Permutationsmatrizen

09.02.05 Definitionen und Konventionen

07.02.05 Ausgleichsproblem

07.02.05 Changelog angelegt.

TODO

- Quadraturverfahren

[1] ☞ Bibliographie (A, S. 35)

Lizenz

Definitionen und Konventionen

Eine übliche Symbolkenntnis wird vorausgesetzt. Darüberhinaus verwende ich folgende Symbole:

♪	Der „Blah"–Operator.
$\perp$	Eine Erdung; mit einer Streichung vergleichbar.

0.1. Computertomographie

Ganz große Leistung der Mathematiker und Informatiker. Und der Ingenieure natürlich auch.

Siehe auch: `http://de.wikipedia.org/wiki/Computertomographie`.

1. Numerik

1.1. Nichtlineare Gleichungssysteme

1.1.1. Newton–Verfahren

Die Situation

Wir haben eine Funktion $\varphi : [a, b] \to \mathbb{R}$, die zweimal stetig differenzierbar ist und suchen eine Nullstelle x^*.

Bedingungen

Satz:
Die Funktion $\varphi : I \to \mathbb{R}$ sei zweimal stetig differenzierbar und x^* sei einfache Nullstelle von $\varphi(x)$. Das Newton–Verfahren konvergiert gegen x^*, falls der Startwert x_0 nahe genug bei x^* liegt.

Das Verfahren

$$x_{n+1} = x_n - \frac{\varphi(x_n)}{\varphi'(x_n)}$$

1.1.2. Heron–Verfahren

HERON von Alexandria. ☞ Euklid (2.0.6, S. 27).

Die Situation

Wir wollen $\sqrt{a},\ a \in \mathbb{R}$ berechnen. Mit einem Trick können wir das unter Benutzung des NEWTON–Verfahrens (☞ NEWTON (1.1.1, S. 8)) tun.

Das Verfahren

- Stelle $f(x) = x^2 - a$ auf.
- Bestimme x^* für $f(x^*) = 0$. Am besten mit NEWTON[a].

[a] ☞ NEWTON–Verfahren (1.1.1, S. 8)

1.1.3. Fixpunktiteration — Banachscher Fixpunktsatz

Die Situation

$$\varphi(x) : I \to I, \quad I = [a, b]$$

ist desorientiert und braucht einen Fixpunkt x^* mit $\varphi(x^*) = x^*$.

Satz:
Sei $\varphi : I \to \mathbb{R}$ stetig differenzierbar mit

- $\varphi(I) \subseteq I$ (kontrahierend) und
- $|\varphi'(x)'| \leq L < 1$ für alle $x \in I$

Dann besitzt φ genau einen Fixpunkt x^* in I und die Iteration

$$x_n := \varphi(x_{n_1}) \quad n = 1, 2, \ldots$$

konvergiert für jedes $x_0 \in I$ gegen x^*.

Voraussetzungen

- I abgeschlossen
- φ ist Abbildung in sich:

  $\rightsquigarrow$ Monotonie

- φ kontrahiert: $\exists\, 0 \leq L < 1 \forall x, y \in I : |\varphi(y) - \varphi(x)| \leq L|x - y|$

  ! Wenn φ stetig differenzierbar: φ kontrahiert $\forall x \in I :\Leftrightarrow |\varphi'(x)| \leq L < 1$
  ☞ MWS (2.0.5, S. 27).

Fehlerabschätzungen

A–Priori–Schranke

$$|x_n - x^*| \leq \frac{L^n}{1 - L}|x_1 - x_0|$$

A–Posteriori–Schranke

$$|x_n - x^*| \leq \frac{L}{1 - L}|x_n - x_{n-1}|$$

Durchführung

Völlig unkritisch, da meist $x_{n+1} = \varphi(x_n)$ und $x_0 \in I$.

1.2. Lineare Gleichungssysteme

1.2.1. LR–Zerlegung

Die Situation

Wir haben eine Matrix A und wollen damit das LGS $Ax = b$ lösen. Die direkte Lösung ist zu hardcore, weshalb wir die Matrix in eine untere und obere Dreiecksmatrix zerlegen wollen, damit die Lösung des LGS einfacher wird.

Die Zerlegung

Eine LR–Zerlegung sieht so aus:

$$A = LR = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \text{♪} & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \text{♪} & \dots & \text{♪} & 1 \end{pmatrix} \cdot \begin{pmatrix} \text{♪} & \text{♪} & \dots & \text{♪} \\ 0 & \text{♪} & \dots & \text{♪} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \text{♪} \end{pmatrix}$$

Das Verfahren

Das Lösungsverfahren ist zweigeteilt. Zuerst wird zerlegt, danach wird mit der Zerlegung gelöst.

LR–Zerlegung
Man führt systematisch den Gauss–Algorithmus durch, um A in R–Form zu bringen. Man bringt die Felder spaltenweise von links nach rechts auf Null; eine Spalte pro Schritt. Zusätzlich muss im k–ten Schritt die k–te Zeile dazu verwendet werden, alle Zeilen, die darunter liegen, um ein Feld „kürzer zu machen". Zeilenvertauschungen dürfen nicht durchgeführt werden.
Die einzelnen Schritte des Gauß–Verfahrens werden nicht wie üblich rechts an Pfeilen notiert. Der Faktor, mit dem die im letzten Schritt verwendete Zeile multipliziert wurde, um ein Feld zu nullen, wird negativiert in dieses Feld geschrieben. Zur Unterscheidung zwischen „echten" Matrixfeldern und „Logfeldern" werden die Logfelder mit Linien abgetrennt.
Ist man fertig, so bilden die Logfelder die L–Matrix und die anderen Felder die R–Matrix.
Gilt nach einem Nachrechnen $LR \neq A$, so funktioniert das Verfahren mit A so nicht. Man kann dann eventuell die Zeilen von A so permutieren, dass das Verfahren trotzdem funktioniert.

$$\begin{pmatrix} \text{♪} & \text{♪} & \text{♪} \\ \text{♪} & \text{♪} & \text{♪} \\ \text{♪} & \text{♪} & \text{♪} \end{pmatrix} \Rightarrow \left(\begin{array}{c|cc} \text{♪} & \text{♪} & \text{♪} \\ \hline -x & \text{♪} & \text{♪} \\ -y & \text{♪} & \text{♪} \end{array}\right) \rightsquigarrow$$

$$\begin{pmatrix} \text{♪} & \text{♪} & \text{♪} \\ 0 & \text{♪} & \text{♪} \\ 0 & \text{♪} & \text{♪} \end{pmatrix} \Rightarrow \left(\begin{array}{c|c|c} \text{♪} & \text{♪} & \text{♪} \\ \hline -x & \text{♪} & \text{♪} \\ \cline{2-2} -y & -z & \text{♪} \end{array}\right)$$

Lösung des LGS
Analog wie bei der CHOLESKY–Zerlegung:

$$\begin{aligned} Ax &= b \Rightarrow \\ L\underbrace{Rx}_{=:y} &= b \\ Ly &= b \Rightarrow y \\ Rx &= y \end{aligned}$$

Permutationsmatrizen

Definition:
Eine Permutationsmatrix ist eine Matrix, bei der in jeder Zeile und in jeder Spalte eine 1 steht, sonst Nullen. Eine Linksmultiplikation mit einer Permutationsmatrix ergibt eine zeilenweise, eine Rechtsmultiplikation eine Spaltenweise Vertauschung.

Damit man sich das vorstellen kann, ein einfaches

Beispiel

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix}$$

Man liest das so (von oben nach unten):

1. Erste Zeile wird zur ersten,
2. zweite zur dritten,
3. dritte zur zweiten.

Gelingt eine LR–Zerlegung nicht, ist die Matrix in dieser Form nicht LR–zerlegbar. Eine Zeilenpermutation kann helfen.

1.2.2. Cholesky–Zerlegung

Die Cholesky–Zerlegung findet in Numerik–Klausuren und im Bereich der Computertomographie (☞ (0.1, S. 7)) Anwendung.

Die Situation

Wir haben eine positiv definite und symmetrische (☞ Symmetrie, Definitheit (2.0.8, S. 30)) Matrix $A \in \mathbb{R}^{n\times n}$ und wollen damit das LGS $Ax = b$ lösen. Als ob wir nichts anderes zu tun hätten. Aber so sind wir halt.

Die Grundlagen

Satz:
Eine positiv definite[a] Matrix A lässt sich zerlegen in das Produkt einer unteren Dreiecksmatrix mit ihrer transponierten Matrix $A = LL^\top$.

[a] und symmetrische (Anm. d. Autors)

Vorsicht Schmechz!

Cholesky–Algorithmus
Man berechnet L zeilenweise:

- $l_{11} = \sqrt{a_{11}}$
- $l_{jk} = \frac{1}{l_{kk}}(a_{jk} - \underbrace{\sum_{m=1}^{k-1} l_{jm} l_{km}}_{A_{jk}} \qquad j < k)$
- $l_{jj} = \sqrt{a_{jj} - \underbrace{\sum_{m=1}^{j-1} |l_{jm}|^2}_{B_{jj}}}$

Merkhilfe
Für die bedenklichen Summen A und B gilt:

- $A = l_{j1}l_{k1} + l_{j2}l_{k2} + \cdots + l_{jk-1}l_{kk-1}$. Man multipliziert also für jede Spalte $< k$ den in der k–ten und j–ten Spalte stehenden Wert mineinander und addiert sie auf:

$$\begin{pmatrix} \eighthnote & 0 & 0 & 0 & 0 & 0 \\ \eighthnote & \eighthnote & 0 & 0 & 0 & 0 \\ \times & \times & \eighthnote & 0 & 0 & 0 \\ \eighthnote & \eighthnote & \eighthnote & \eighthnote & 0 & 0 \\ \times & \times & (5,3) & \eighthnote & \eighthnote & 0 \\ \eighthnote & \eighthnote & \eighthnote & \eighthnote & \eighthnote & \eighthnote \end{pmatrix}$$

- $B = |l_{j1}|^2 + |l_{j2}|^2 + \cdots + |l_{jj-1}|^2$. Man quadriert also die Beträge aller voranstehenden Werte in der Zeile und addiert sie auf:

$$\begin{pmatrix} \eighthnote & 0 & 0 & 0 & 0 & 0 \\ \eighthnote & \eighthnote & 0 & 0 & 0 & 0 \\ \eighthnote & \eighthnote & \eighthnote & 0 & 0 & 0 \\ \eighthnote & \eighthnote & \eighthnote & \eighthnote & 0 & 0 \\ |\times|^2 & |\times|^2 & |\times|^2 & (5,4) & \eighthnote & 0 \\ \eighthnote & \eighthnote & \eighthnote & \eighthnote & \eighthnote & \eighthnote \end{pmatrix}$$

Das Verfahren

- Berechne L und daraus $L^\top$.
- Man suhlt sich im Schmerz und überprüft L nochmal (Rechenfehlergefahr hoch $\Leftrightarrow$ da PAIN $\Leftrightarrow$ abartige Formeln).
- Man baut:

$$\begin{aligned} Ax &= b \\ L\underbrace{L^\top x}_{=:y} &= b \end{aligned}$$

- Man rechnet aus:

$$\begin{aligned} Ly &= b \Rightarrow y \\ L^\top x &= y \Rightarrow x \end{aligned}$$

1.2.3. Ausgleichsproblem

Die Situation

Es liegt ein überbestimmtes Gleichungssystem vor. Es soll trotzdem möglichst gut gelöst werden. Diese Situation tritt oft bei der Computertomographie (☞ CT (0.1, S. 7)) auf.

Formal:

$$\begin{aligned} &Ax = b \\ &A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ \operatorname{Rang} A = n,\ m > n \\ &b - Ax^* \text{ soll möglichst klein sein} \end{aligned}$$

$b - Ax$ bezeichnet man auch als das Residuum.

Definition:

$$r(x) := b - Ax$$

heißt Residuum (Restvektor).

Das Problem kann also nochmal neu formuliert werden: „Finde x^* mit $r(x^*)$ minimal."

Definition:
Die Normalgleichung lautet

$$A^\top Ax = A^\top b$$

Satz:
Die Matrix A besitze vollen Rang. Die eindeutig bestimmte Lösung x^* der Normalgleichungen löst das überbestimmte lineare Gleichungssystem $Ax = b$ im Sinne der Euklidischen Norm[a] bestmöglich, d.h. $||r(x^*)||_2$ ist minimal.

[a]☞ Normen (2.0.10, S. 32)

Das Verfahren

- Stelle $r(x)$ auf, falls nicht schon gegeben.
- Berechne $A^\top A$ und $A^\top b$
- Löse $A^\top Ax = A^\top b$

1.3. Polynome und Splines

1.3.1. Gerschgorin

=)

Die Situation

Wir haben ein reelles Polynom p und sollen irgendwas mit einem Kreis und den Nullstellen rausfinden.

Der Satz

Satz von Gerschgorin:
Die Nullstellen $\xi_1, \ldots, \xi_n$ des Polynoms $p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0$ liegen in einem Kreis $\mathcal{K}$ um Null mit Radius r, wobei

$$r \leq max \left\{ 1, \sum_{j=0}^{n-1} |a_j| \right\} \quad \text{und}$$

$$r \leq max \left\{ 1 + |a_{n-1}|, \ldots, 1 + |a_1|, |a_0| \right\}$$

Zu beachten ist, dass

- x_n keinen Koeffizienten hat
- In der zweiten Feststellung kein 1+ vor $|a_0|$ gesetzt wird.

Zu Deutsch

r ist kleiner oder gleich dem Maximum der aufaddierten Beträge der Koeffizienten (oder halt 1) und kleiner/gleich dem Maximum der um 1 erhöhten Beträge der Koeffizienten, wobei der Betrag vom letzten Koeffizienten nicht um 1 erhöht wird.

Sogar noch windiger als die Sturmsche Kette[1].

1.3.2. Sturmsche Kette

Die Situation

Wir haben ein Polynom $p(x) = a_1x_1 + \cdots + a_nx_n$ im Intervall $[\alpha, \beta]$. Uns interessiert die Anzahl der Nullstellen.

[1] ☞ Sturm (1.3.2, S. 17)

Das Verfahren

Was muss man machen?

- Aufbau einer STURMschen Kette[a]:
  - $p_0(x) = p(x)$
  - $p_1(x) = p'(x)$
  - Ab hier: Iteration durch den EUKLID–Algorithmus[b]: $p_{n-2}(x) = ap_{n-1}(x) - p_n(x)$
    Wir hören natürlich auf, wenn p_ν nicht mehr von von x abhängig ist.
- Jetzt interessieren uns die Vorzeichen den Intervallgrenzen ausgewerteten Polynome:

	p_0	...	p_ν	|VZW|
$x = a$	$+/-/0$	...	$+/-/0$	v_1
$x = b$	$+/-/0$	...	$+/-/0$	v_2

- Der Satz von STURM sagt nun über die Anzahl der Nullstellen von p in $[a, b]$:

$$Z = v_1 - v_2$$

[a] windig
[b] ☞ EUKLID (2.0.6, S. 27)

1.3.3. Tschebyscheff–Polynome

Definition:

$$\begin{aligned} x &= \cos\theta, \quad 0 \le \theta \le \pi \\ T_n(x) &:= \cos n\theta \\ &= \cos(n \arccos x) \end{aligned}$$

Rekursionsformel

$$T_{n+1} = 2xT_n(x) - T_{n-1}(x)$$

Beispiele

$$\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
T_5(x) &= 16x^5 - 20x^3 + 5x \\
T_6(x) &= 32x^6 - 48x^4 + 18x^2 - 1
\end{aligned}$$

Mithilfe der Definition lassen sich T_0 und T_1 bequem errechnen. Zu Bestimmung der weiteren Tschebyscheff–Polynome ist die Rekursionsformel praktisch.

Entwicklung von Polynomen nach Tschebyscheff–Polynomen

Da die Tschebyscheff–Polynome eine Basis des $\mathbb{R}[X]$ bilden, lässt sich jedes Polynom als Linearkombination von Tschebyscheff–Polynomen formulieren.

Die Vorteile einer Tschebyscheff–Entwicklung liegen einerseits darin, dass die Darstellung stabil ist[2] und andererseits darin, dass die Koeffizienten von unten nach oben abnehmen.

[2] Störung in den Koeffizienten verursacht kleine Auswirkung

Achenbach–Pajor–Saqe–Verfahren[a] zur Tschebyscheff–Entwicklung eines Polynoms
Dieses Verfahren ist in einer Lerngruppe gefunden worden; es ist mit Sicherheit weder optimal noch unbekannt:
Gegeben sei das Polynom $p(x) = b_n x^n + \cdots + b_0$ vom Grad n.

- Bestimme $T_0(x)$ bis $T_n(x)$.
- Stelle auf: $\overline{p}(x) = a_n T_n(x) + \cdots + a_0 T_0(x) = a_n T_n(x) + \cdots + a_0$.
- Bringe $\overline{p}(x)$ in Polynomnormalform (nach Potenzen von x absteigend sortiert: $\overline{p}(x) = (a_i + a_j - a_k)x^n + \ldots$.
- Stelle LGS auf: $(a_i + a_j - a_k) = b_n, \ldots$.
- Löse LGS nach allen a_ψ.
- Nun gilt: $p(x) = \overline{p}(x)$.

[a] die Reihenfolge ergibt sich aus der lexikographischen Ochdnung

1.3.4. Splines

Definition:
Ein Gitter Δ ist eine Unterteilung eines Intervalls $[a, b]$, für die gilt:

$$a = x_0 < x_1 < \cdots < x_n = b$$

Definition:
Eine Abbildung $s_\Delta : [a, b] \to \mathbb{R}$ heißt Spline vom Grad l bezüglich des Gitters Δ, wennst:

- $s_\Delta|_{[x_{j-1}, x_j]}$ ist Polynom mit Grad $l \quad \forall j \in \{1, \ldots, n\}$
- s_Δ ist in $[a, b]$ $l - 1$ mal stetig differenzierbar

Definition:
Der Raum der Splines vom Grad l bezüglich Δ wird mit

$$S_l(\Delta)$$

bezeichnet. Seine Dimension ist $n + l$.

Prominente Splineklassen

Spline vom Grad 0:	Treppenfunktion	(stückweise stetig)
Spline vom Grad 1:	Polygonzug	(stetig)
Spline vom Grad 2:	Parabelzug	(stetig diff'bar)
Spline vom Grad 3:	Kubischer Spline	

Darstellung von Splines

Definition:
Eine abgeschnittene Potenzfunktion x_+^m ist definiert durch

$$x_+^m := \begin{cases} x^m & , x \geq 0 \\ 0 & , x < 0 \end{cases}$$

Darstellung mit abgeschnittener Potenzfunktion (durch)

$$s(x) = \sum_{j=0}^{2} b_j (x - x_0)_+^j + \sum_{k=0}^{n-1} c_k (x - x_k)_+^3$$

Darstellung intervallweise durch Polynome

$$s(x) = \begin{cases} p_0(x) & \text{in } [x_0, x_1] \\ p_1(X) & \text{in } [x_1, x_2] \\ \vdots & \\ p_{n-1}(x) & \text{in } [x_{n-1}, x_n] \end{cases}$$

Hierbei sind $p_0, \ldots, p_{n-1}$ Polynome vom Grad 3.

Darstellung durch B–Splines
B–Splines bilden eine Basis von $S_l(\Delta)$. Die Darstellung von Polynomen durch B–Splines setzt das Verständnis von B–Splines voraus und wird daher weggelassen.

Kubische Spline–Interpolation

Die Situation ist die, dass wir auf dem Gitter Δ die Daten $y_0, \ldots, y_n$ mit $y_j = f(x_j)$ gegeben haben und einen kubischen Spline so interpolieren wollen, dass er durch alle Stützstellen geht.

Man kann hier zwei zusätzliche sinnvolle Randvorgaben wählen:

- $s'(x_0) = y_0'$ und $s'(x_n) = y_n'$
- $s''(x_0) = s''(x_n) = 0$

Die Interpolationsaufgabe ist mit einer dieser Randvorgaben eindeutig lösbar.

Fehler bei der Interpolation:
Für den interpolierten kubischen Spline s_f zu der in $[a, b]$ 4–mal stetig differenzierbaren Funktion f bezüglich äquidistanter[a] Knoten mit einer der beiden Randvorgaben gilt:

$$||f - s_f||_\infty \leq 2h^4||f^{(4)}||_\infty$$

Weiterhin gilt gleichmäßige Konvergenz mit $O(h^4)$ für $h \to 0$.

[a] !

Erkennen von Splines

Sag mir, sag mir,
Funktion sag mir,
bist du ein Spline?

kubisch musst du sein,
sag mir, sag mir,
ist die Stetigkeit dein?

Sag mir, sag mir,
Funktion sag mir,
bist du ein Spline?

Randvorgaben sollst du erfüllen,
sag mir, sag mir
kann ich dich in die Lösung einhüllen?

Sag mir, sag mir,
Funktion sag mir,
bist du ein Spline?

Interpolationsbedingungen sind gefragt,
sag mir, sag mir,
sind sie erfüllt derart?

Sag mir, sag mir,
Funktion sag mir,
bist du ein Spline?

Erkennen von kubischen Splines

- stückweise kubisches Polynom oder abgeschnittene Potenzfunktion?
- Interpolationsbedingung erfüllt?

$$s(x_i) = f(x_i) \quad \forall i \in \{0, \ldots, n\}$$

- Dreimal stetig?

$$\lim_{\varepsilon \to 0} s(x_i + \varepsilon) = \lim_{\varepsilon \to 0} s(x_i - \varepsilon)$$
$$\lim_{\varepsilon \to 0} s'(x_i + \varepsilon) = \lim_{\varepsilon \to 0} s'(x_i - \varepsilon)$$
$$\lim_{\varepsilon \to 0} s''(x_i + \varepsilon) = \lim_{\varepsilon \to 0} s''(x_i - \varepsilon)$$

- (optional?) Ist eine der beiden sinnvollen Randvorgaben erfüllt?

$$s'(x_0) = y'_0 \wedge s'(x_n) = y'_n$$
$$\vee$$
$$s''(x_0) = s''(x_n) = 0$$

1.4. Algebraische (Polynom–)Interpolation

Der Plan bei der algebraischen Interpolation ist, für eine gegebene Reihe von Zeit–Wert–Paaren ein Polynom zu finden, das diese Stellen durchläuft und sich an die tatsächliche Funktion, die diesen Paaren zugrundeliegt, annähert.

Die Zeitwerte seien im Folgenden mit $x_0, \ldots, x_n$ bezeichnet und im Allgemeinen äquidistant und verschieden. Die Wert-Werte (Daten) seien mit $y_0, \ldots, y_n$ bezeichnet.

1.4.1. Normal (Vandermonde–Matrix)

Definition:
Die Vandermonde–Matrix sieht so aus:

$$V_n = \begin{pmatrix} x_0^0 & x_0^1 & x_0^2 & \dots & x_0^n \\ x_1^0 & x_1^1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \\ x_n^0 & x_n^1 & x_n^2 & \dots x_n^n & \end{pmatrix}$$
$$= \begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix}$$

Darstellung des Interpolationspolynoms in Normalgestalt

- Löse $V_n a = y$
- Formuliere $p(x) = \sum_{k=0}^{n} a_k x^k$

1.4.2. Lagrange–Darstellung

- Formuliere die Lagrange–Grundfunktionen $l_j(x)$:

$$l_j(x) = \prod_{k=0, k\neq j}^{n} \frac{x - x_k}{x_j - x_k}$$

- Formuliere $p(x) = \sum_{j=0}^{n} y_j l_j(x)$

1.4.3. Newton–Gestalt

- Formuliere die Grundfunktionen $w_0, \ldots, w_j$:

$$w_0 := 1$$
$$w_k := w_{k-1} \cdot (x - x_{k-1})$$

- Bestimme $a_0, \ldots, a_n$ nach folgendem abgefahrenen Schema:

$$\begin{array}{|c|ccccc}
x_0 & f(x_k) & \frac{f(x_1)-f(x_0)}{x_1-x_0} =: \alpha_0 & \frac{\alpha_1-\alpha_0}{x_2-x_0} =: \beta_0 & \ldots & \zeta_0 \\
x_1 & f(x_1) & \frac{f(x_1)-f(x_0)}{x_1-x_0} =: \alpha_1 & \ldots & & \\
\vdots & & & & & \\
x_n & \ldots & & & &
\end{array}$$

  Nun gilt: $a_0 = \alpha_0,\ a_1 = \beta_0, \ldots, a_n = \zeta_0$

- Forumuliere $p(x) = \sum_{j=0}^{m} a_j w_j(x)$.

1.4.4. Bézier–Technologie

Bernstein–Polynome

Definition:
Das i–te BERNSTEIN–Grundpolynom auf dem Intervall $[0,1]$ ist definiert durch:

$$B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i$$

Definition:
Die <u>BÉZIER–Koeffizienten</u> b_i lassen sich bezüglich der Knoten $\frac{0}{n}, \frac{1}{n}, \ldots \frac{n}{n}$ darstellen als

$$\begin{pmatrix} 0 \\ b_0 \end{pmatrix}, \begin{pmatrix} \frac{1}{n} \\ b_1 \end{pmatrix}, \ldots, \begin{pmatrix} \frac{n}{n} \\ b_n \end{pmatrix}$$

Satz:
Jedes Polynom P lässt sich unter Zuhilfenahme der Kontrollpunkte $b_i \in \mathbb{R}^d$ darstellen als:

$$P(t) = \sum_{i=0}^{n} b_i B_i^n(t)$$

$P([0,1])$ liegt in der konvexen Hülle der Kontrollpunkte.

Algorithmus von de Casteljau

Wir wollen Bézierkurven auswerten und darstellen. Der Casteljau–Algorithmus bietet sich an, weil das mit ihm prima funktioniert.
b_i bezeichne den i–ten Kontrollpunkt. Es seien n Kontrollpunkte im Intervall $[0,1]$ (äquidistant?) gegeben. Dann gilt:

$$\beta_\nu^{(0)} := b_\nu$$
$$\beta_\nu^{(k)} := (1-t)\cdot\beta_\nu^{(k-1)} + t\cdot\beta_{\nu+1}^{(k-1)}$$
$$p(t) = \beta_0^{(n)}$$

Schematisch:

$\beta_0^{(0)}$	$\nearrow \cdot t + \rightarrow \cdot(1-t)$	$\beta_0^{(1)}$	$\nearrow \cdot t + \rightarrow \cdot(1-t)$	$\beta_0^{(2)}$	$\dots$	$\beta_0^{(n)}$
$\beta_1^{(0)}$	$\nearrow \cdot t + \rightarrow \cdot(1-t)$	$\beta_0^{(1)}$	$\dots$			
$\vdots$						
$\beta_n^{(0)}$						

2. Referenz

2.0.5. Mittelwertsatz der Differentialrechnung (MWS)

Der MWS ist wesentlich bei der praktischen Durchführung (Klausur) von Fixpunktiteration[1]saufgaben.

> **Mittelwertsatz der Differentialrechnung:**
> Sei $f : [a, b] \to \mathbb{R}$ eine auf $[a, b]$ definierte und stetige Funktion und in (a, b) differenzierbar. Dann existiert mindestens ein ξ, so dass gilt:
>
> $$f'(\xi) = \frac{f(b) - f(a)}{b - a}$$

Das heißt:

Die Steigung, die zwischen a und b ist, gibts dazwischen auf der Kurve der Funktion mindestens nocheinmal.

Siehe auch: `http://de.wikipedia.org/wiki/Mittelwertsatz`

2.0.6. Euklid–Algorithmus

Übrigens: Euklid war — wie Heron — von Alexandria.

Euklids Plan bei der Findung seines Algorithmus war, quasi den geometrisch größten gemeinsamen Teiler zweier Strecken zu finden. Dazu zog er iterativ die größere von der kleineren ab, bis beide gleich waren. Und fertig war der ggT. Abgefahrenerweise funktioniert der Algorithmus auch für Polynome und zur Einschüchterung von Studenten.

[1] ☞ Fixpunktiteration (1.1.3, S. 9)

Der eigentliche Algorithmus

```
ggT ggT(a, b) {

  let r be undefined;
  m := a;
  n := b;

  while (r != 0) {
    if (m < n) {
      vertausche(m, n);
    }
    r := m mod n;
    m := n;
    n := r;
  }
  cheer();
  return m;
}
```

Euklid für Polynome

Das Euklidverfahren für Polynome verläuft analog zum „normalen“ Verfahren. Allerdings benutzt man zur Restbildung bei Division die Polynomdivision. Man muss allerdings beachten, dass die Darstellung des Restes bei der Polynomdivision durch eine Subtraktion erfolgt.

Beispiel:

$$\begin{array}{l}
\big(\quad 2x^3 - 3x^2 - 11x \quad + 6\big) \div \big(6x^2 - 6x - 11\big) = \frac{1}{3}x - \frac{1}{6} + \dfrac{-\frac{25}{3}x + \frac{25}{6}}{6x^2 - 6x - 11} \\
\underline{\;-\, 2x^3 + 2x^2 + \frac{11}{3}x\;} \\
\qquad\quad - x^2 - \frac{22}{3}x \quad + 6 \\
\qquad\quad \underline{\quad x^2 \quad - x - \frac{11}{6}} \\
\qquad\qquad\quad - \frac{25}{3}x + \frac{25}{6}
\end{array}$$

Der Rest bei Division ($-$ beachten!) ist also $\left(\frac{25}{3}x - \frac{25}{6}\right)$.

2.0.7. Horner–Schema

Das Hornerschema ist ein Verfahren zur Polynomauswertung, mit dem man

- Den Wert eines Polynoms an einer Stelle errechnen (einfaches Hornerschema)

- Die Werte aller Ableitungen eines Polynoms an einer Stelle errechnen (vollständiges Hornerschema)
- Ein Polynom „umentwickeln“

kann.

Die Anwendung des Hornerschemas ist hochgradig trivial, wenn man's mal kapiert hat:

Einfaches und vollständiges Hornerschema
Gegeben sei das Polynom $p(x) = a_n x^n + \cdots + a_0 x^0$ und eine Stelle $\overline{x}$.

- Mache Rohform der Tabelle:

	a_n	a_{n-1}	$\dots$	a_0
$\overline{x}$				
0				

  Nicht auftretende Koeffizienten müssen natürlich mit 0 eingetragen werden!

- Es wird von links nach rechts die zweite und dritte Zeile ausgefüllt:
  - Der Wert eines Feldes in der zweiten Zeile ergibt sich aus: Feld links unten $\cdot\, \overline{x}$.
  - Der Wert eines Feldes in der dritten Zeile ist die Summe der beiden obenstehenden.

	a_n	a_{n-1}	$\dots$	a_0
$\overline{x}$	$0 \cdot \overline{x}$	$b_n \cdot \overline{x}$	$\dots$	$b_1 \cdot \overline{x}$
0	$a_n + 0 \cdot \overline{x} := b_n$	$\sum$	$\dots$	$\sum = \frac{f(\overline{x})}{0!} = f(\overline{x})$

- Hier wären wir mit dem einfachen Hornerschema fertig. Allerdings wäre das zu einfach für uns. Wir machen also genauso weiter, erden bei jedem folgenden Iterationsschritt allerdings den letzten Koeffizienten:

	a_n	a_{n-1}	$\dots$	a_0
$\overline{x}$	$0 \cdot \overline{x}$	$b_n \cdot \overline{x}$	$\dots$	$b_1 \cdot \overline{x}$
0	$a_n + 0 \cdot \overline{x} := b_n$	$\sum$	$\dots$	$\frac{f(\overline{x})}{0!} = f(\overline{x})$
$\overline{x}$	♪	♪	$\dots$	$\perp$
0	$\sum$	$\sum$	$\frac{f'(\overline{x})}{1!}$	
$\overline{x}$	$\dots$	$\dots$	$\perp$	
0	$\frac{f^{(n)}(\overline{x})}{n!}$	$\frac{f^{(n-1)}(\overline{x})}{n-1!}$		

Wir haben für das Polynom $p(x)$ eine vollständige Hornerentwicklung nach $\overline{x}$ durchgeführt. Damit können wir jetzt sogar noch das Polynom umbauen:

„Umentwicklung“ eines Polynoms mit Horner
Sei b_k der im $k+1$–ten Schritt ermittelte Wert (ohne Fakultätskorrektur). Also $\frac{f^{(k)}}{k!} = b_k$. Dann gilt:

$$p(x) = b_n(x-\overline{x})^n + b_{n-1}(x-\overline{x})^{n-1} + \cdots + b_0$$

Also von unten nach oben.

Beispiel

Wir hornern $2x^4 + 5x^2 - 7x + 21$ bei $\overline{x} = 3$.

$$\begin{array}{c|ccccc} & 2 & 0 & 5 & -7 & 21 \\ 3 & & & & & \\ \hline & 2 & & & & \end{array}$$

$$\begin{array}{c|ccccc} & 2 & 0 & 5 & -7 & 21 \\ 3 & & 6 & & & \\ \hline & 2 & 6 & & & \end{array}$$

$$\begin{array}{c|ccccc} & 2 & 0 & 5 & -7 & 21 \\ 3 & & 6 & 18 & & \\ \hline & 2 & 6 & 23 & & \end{array}$$

$$\begin{array}{c|ccccc} & 2 & 0 & 5 & -7 & 21 \\ 3 & & 6 & 18 & 69 & \\ \hline & 2 & 6 & 23 & 62 & \end{array}$$

$$\begin{array}{c|ccccc} & 2 & 0 & 5 & -7 & 21 \\ 3 & & 6 & 18 & 69 & 186 \\ \hline & 2 & 6 & 23 & 62 & \boxed{207} \end{array}$$

2.0.8. Definitheit und Symmetrie bei Matrizen

Definition:
Matrix A heißt <u>symmetrisch</u> $\Leftrightarrow A = A^\top$.

Wikipedia[2] sagt:

> **Hauptminoren-Kriterium für Definitheit:**
> Die symmetrische Matrix A ist genau dann positiv definit, wenn alle Hauptminoren von A positiv sind.

Hierbei sind die Hauptminoren einer $n \times n$–Matrix alle linken oberen $k \times k$–Matrizen.

Sinnvoll anwendbar ist dieses Kriterium für Matrizen bis zum Format 3×3.

2.0.9. Determinanten

Die Determinante ist eine Abbildung $det : \mathbb{R}^{n \times m} \to \mathbb{R}$, die für ein lineares Gleichungssystem bestimmt, ob es eine eindeutige Lösung besitzt. Außerdem hat sie noch andere Eigenschaften.

Bestimmung von Determinanten

Formeln

$$\begin{aligned} \det(\begin{pmatrix} a \end{pmatrix}) &= a \\ \det(\begin{pmatrix} a & b \\ c & d \end{pmatrix}) &= ac - bd \\ \det(\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}) &= aei + cdh + bfg - ceg - afh - bdi \end{aligned}$$

Für größere Matrizen isses zu abgefahren.

> **Satz:**
> Ist A eine obere/untere Dreiecksmatrix, so folgt aus der Definition der Determinante, dass
>
> $\det A = a_{11} \cdot \dots \cdot a_{nn}$

[2] http://de.wikipedia.org/wiki/Minor_(Mathematik)

Allgemeines Verfahren

- Bestimmte untere/obere Dreiecksmatrix
- Multipliziere die Diagonalelemente miteinander

2.0.10. Metrik und Normen

Das Wissen über Metriken ist für Numerik–Klausuren nicht relevant, hilft aber beim Verständnis.

Definition:
Sei M eine beliebige Menge. Eine Metrik ist eine Abbildung $d : M \to \mathbb{R}$ mit folgenden Eigenschaften $\forall x, y, z \in M$:

1. $d(x, x) = 0$
2. $d(x, y) = 0 \Rightarrow x = y$ (Definitheit)
3. $d(x, y) = d(y, x)$ (Symmetrie)
4. $d(x, y) \leq d(x, z) + d(z, y)$ (Dreiecksungleichung)

(M, d) heißt metrischer Raum.
Sachen, die auf einer Landkarte funktionieren, funktionieren in einem beliebigen metrischen Raum.

Vektornormen

Definition:
Eine Vektornorm ist eine Abbildung $|| \cdot || : \mathbb{R}^n \to \mathbb{R}$ mit folgenden Eigenschaften:

1. $||x|| \geq 0$
2. $||x|| = 0 \Rightarrow x = 0$ (Definitheit)
3. $\alpha x|| = |\alpha| ||x|| \quad (\alpha \in \mathbb{R})$ (Homogenität)
4. $||x + y|| \leq ||x|| + ||y||$ (Dreiecksungleichung)

Eine Vektornorm induziert die Metrik $d(x, y) := ||x - y||$.

Eine allgemeine (p–)Vektornorm ist definiert als

$$||x||_p := \sqrt[p]{\sum_{i=1}^{n} |x_i|^p}$$

spezielle Vektornormen

- $||x||_\infty := \max_j |x_j|$ (max–Norm)
- $||x||_1 := \sum_j |x_j|$ (l_1–Norm, Manhattan–Norm)
- $||x||_2 := \sqrt{\sum_j |x_j|^2}$ (Euklid–Norm)

Matrixnormen

Definition:
Eine Abbildung $N : \mathbb{R}^{n\times n} \to \mathbb{R}$ heißt Matrixnorm, falls gilt:

1. $N(A) \geq 0$
2. $N(A) = 0 \Rightarrow A = o$ (Definitheit)
3. $N(\alpha A) = |\alpha| N(A) \quad (\alpha \in \mathbb{R})$ (Homogenität)
4. $N(A + B) \leq N(A) + N(B)$ (Dreiecksungleichung)
5. $N(AB) \leq N(A) \cdot N(B)$ ((Sub–?)Multiplikativität)

prominente Matrixnormen

- $N_G(A) := n \max_{j,k} |a_{jk}|$ (Gesamtnorm)
- $N_Z(A) := \max_j \sum_{k=1}^{n} |a_{jk}|$ (Zeilensummennorm)
- $N_S(A) := \max_k \sum_{j=1}^{n} |a_{jk}|$ (Spaltensummennorm)

Verträglichkeit von Normen

Definition:
Eine Matrixnorm $N(\cdot)$ und eine Vektornorm $||\cdot||$ heißen verträglich, falls gilt:

$$||Ax|| \leq N(A)\,||x|| \quad \forall x \in \mathbb{R}^n,\ A \in \mathbb{R}^{n\times n}$$

Verträgliche Normen

Die folgenden Normen sind verträglich:

- $||x||_\infty$ verträgt sich mit $N_Z(A)$ und $N_G(A)$.
- $||x||_1$ verträgt sich mit $N_S(A)$ und $N_G(A)$.
- $||x||_2$ verträgt sich mit $N_G(A)$.

A. Anhang

Literaturverzeichnis

[1] Wikipedia, http://de.wikipedia.org

[2] Skriptum zur Vorlesung Numerische Mathematik für die Fachrichtung Informatik und Ingenieurwesen, Prof. Dr. Rudolf Scherer, SS04

[3] Inoffizielles Skriptum zur Vorlesung Numerische Mathematik für Informatiker und Ingenieure, basierend auf Vorlesungen von Prof. Dr. Rieder, SS03. ???

[4] Schatzkiste für Informatiker, Martin Lösch, WS01. ???

[5] Lineare Algebra und Analytische Geometrie, Dr. V. Drumm, Prof. Dr. W. Weil

[6] Höhere Mathematik für Informatiker, Inoffizielles Skriptum, 2000–2003, Daniel Winkler, http://danielwinkler.de/hm/

[7] Repetitorium der Höheren Mathematik, Merziger, Wirth, 4. Auflage, Binomi–Verlag, ISBN 3–923 923–33–3.

[8] Numerik Schemata für Informatiker, Thomas Pajor, 2005, http://www.logn.de/mathe.

Index

A–Posteriori–Schranke, 10
A–Priori–Schranke, 10
Alexandria, 8, 27
Ausgleichsproblem, 15

Bézier, Technologie, 25
Banach, Fixpunktsatz, 9
Bernstein, Polynome, 25

Casteljau, Algorithmus, 26
Cholesky, Zerlegung, 13
Computertomographie, 7

Definitheit, 30
Determinante, 31
Differentialrechnung, MWS, 27

Euklid, Algorithmus, 27

Gerschgorin, 16
ggT, 27
Gitter, 20

Heron, Iteration, 8
Horner, Schema, 28

Interpolation, Polynom–, 23

Kreis um Null, 16

Lagrange, 24
LR–Zerlegung, 10

Matrixnorm, 33
Metrik, 32
Mittelwertsatz, 27

Newton, 25
Newton–Iteration, 8
Norm, 32
Normen, Verträglichkeit, 34
Nullstellen, Polynom, 16

Permutationsmatrix, 12
Polynominterpolation, 23
Polynomnullstellen, 16, 17

Randvorgaben, 22

Spline, 20
Spline–Interpolation, kubisch, 22
Symmetrie (Matrizen, 30

Tschebyscheff, Polynome, 18

Vandermonde, Matrix, 24
Vektornorm, 32