

Algorithmik kontinuierlicher Systeme

Felix Leidl

10. Juli 2024

Inhaltsverzeichnis

Direkte Verfahren	3
LR-Zerlegung	3
Ziel	3
Algorithmus	3
Komplexität	3
Anwendung	3
LRP-Zerlegung	3
QR-Zerlegung	3
Ziel	3
Housholder-Spiegelungen	4
Givens-Rotationen	4
Cholesky-Zerlegung	4
Lineare Ausgleichsrechnung	5
Matrizen	5
Orthogonal	5
Skalarprodukt	5
Tridiagonalmatrix	5
Normen	5
Matrix-Norm bzw. Operator-Norm	5
Konditionszahl	6
Spektralsatz	6
Diskretisierung	6
Quantisierung	6
Interpolation	6
Bezier	6
Bernstein-Polynom	6
Formeigenschaften	7
Auswertung	7
Horner-Bezier	7
de Casteljau	7
Glatter Übergang zwischen benachbarten Kurven	8

Tensor-Produkt-Bezier-Flächen	8
Allgemein	8
Auswertung	8
SVD	8
Informationen	9
Bild	9
Kern	9
Norm	9
Lösungstheorie	9
Pseudo-Inverse	9
Lösen	9
Iterative Verfahren	10

Direkte Verfahren

Direkte Verfahren Lösen ein Problem nach endlich vielen Schritten. Verwendung: kleine, vollbesetzte Matrizen.

LR-Zerlegung

Ziel

$$A = LR$$
$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ * & 1 & & \\ \vdots & & \ddots & \vdots \\ * & & \dots & 1 \end{pmatrix} \begin{pmatrix} * & * & \dots & * \\ 0 & * & & \\ \vdots & & \ddots & \vdots \\ 0 & & \dots & * \end{pmatrix}$$

Algorithmus

1. i -te Zeile in R übertragen
2. i -te Spalte dividiert durch a_{ii} in L über nehmen. Erstes Element der Spalte gleich 1 setzten
3. Mit i -ter Zeile die i -te Spalte eliminieren

Komplexität

$\mathcal{O}(n^3)$

Anwendung

- $\det(A) = \det(L) \times \det(R) = 1 \times \det(R)$
- Lösen mehrerer GLS:
 - $Ly = b$ mit Vorwärtssubstitution $\mathcal{O}(n^2)$
 - $Rx = y$ mit Rückwärtssubstitution $\mathcal{O}(n^2)$

LRP-Zerlegung

$$A = PLR$$

QR-Zerlegung

Ziel

$$A = QR$$

Housholder-Spiegelungen

Mit einer Housholder-Spiegelung in einer Spalte Nullen einfügen (außer Diagonalelement)
→ nach $n - 1$ Schritten erhält man die Dreiecksmatrix R

$$R = H_{n-1} \dots H_2 H_1 A$$
$$Q = (H_{n-1} \dots H_2 H_1)^{-1} = H_1 H_2 \dots H_{n-1}$$

Givens-Rotationen

Mit einer Givens-Rotation ein Element (unterhalb der Diagonalen) zu Null machen
→ nach $n(n - 1)/2$ Schritten erhält man die Dreiecksmatrix R

$$J_{ij}(\varphi) = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & c & & -s & \\ & & & \ddots & & \\ & s & & & c & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}$$

Wobei c_1 an Position jj ist und c_2 an Position ii

$$c = \cos(\varphi) = \frac{\sigma \cdot a_{jj}}{\sqrt{a_{jj}^2 + a_{ij}^2}}$$
$$s = \sin(\varphi) = \frac{-\sigma \cdot a_{ij}}{\sqrt{a_{jj}^2 + a_{ij}^2}}$$
$$\sigma = \text{sign}(a_{ij})$$

Ergebnis:

$$R = J_{m,n^*} \dots J_{2,1} A$$
$$Q = J_{2,1}^T \dots J_{m,n^*}^T$$
$$n^* = \min\{m - 1, n\}$$

Cholesky-Zerlegung

Wenn A symmetrisch und positiv definit ist kann man A faktorisieren in

$$A = LDL^T$$

Wobei L das L der LR-Zerlegung ist und D der Diagonalanteil von R

Lineare Ausgleichsrechnung

Matrizen

Orthogonal

Eine Matrix ist orthogonal, falls eine der Bedingungen erfüllt ist:

- $Q^T Q = Id$
- $Q Q^T = Id$
- Spalten oder Zeilen bilden eine Orthonormalbasis
- Die Abbildung Q ist winkel- und längentreu
- Q erhält das Skalarprodukt: $Qx \circ Qy = x \circ y$

Skalarprodukt

$$x \circ y = \sum_{i=1}^n x_i y_i$$

Tridiagonalmatrix

Die inverse einer tridiagonalen Matrix ist in der Regel voll besetzt

Normen

Eigenschaften:

- definit: $x \neq 0 \Rightarrow \|x\| > 0$
- homogen: $\|\lambda x\| = |\lambda| \cdot \|x\|$
- sub-additiv: $\|x + y\| \leq \|x\| + \|y\|$

Matrix-Norm bzw. Operator-Norm

Erfüllt Normeigenschaften und mehr:

- $\|Id\| = 1$
- sub-multiplikativ: $\|AB\| \leq \|A\| \cdot \|B\|$
- mit der Vektornorm kompatibel: $\|Ax\| \leq \|A\| \cdot \|x\|$
- $\|A\| \geq |\lambda|$

Beispiele:

- Spalten-Summen-Norm: $\|A\|_1$

$$\|A\|_1 = \max_j \{\sum_i |a_{ij}|\}$$

- Zeilen-Summen-Norm: $\|A\|_\infty$

$$\|A\|_\infty = \max_i \{\sum_j |a_{ij}|\}$$

- Spektral-Norm: $\|A\|_2$

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$$

- Frobenius-Norm: $\|A\|_F$

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}$$

Konditionszahl

$$\kappa(A) = \frac{\max_{x \in \mathbb{R}^n, \|x\|=1} \|Ax\|}{\min_{x \in \mathbb{R}^n, \|x\|=1} \|Ax\|}$$

Spektralsatz

Es sei $A \in \mathbb{R}^{m \times m}$ eine reelle symmetrische Matrix. Dann gibt es eine Orthonormalbasis aus Eigenvektoren bzw. $A = V D V^T$, wobei D die Diagonalmatrix aller EW ist und die Spalten von V die normierten EV sind.

Diskretisierung

Quantisierung

Interpolation

Bezier

Bernstein-Polynom

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

Bildet das Polynom vom Grad n

$$\binom{n}{i} = \frac{n!}{i! \cdot (n-i)!}$$

Es gilt:

- $0 \leq B_i^n(t) \leq 1$ für $t \in [0, 1]$
- $B_i^n(t)$ hat eine i -fache Nullstelle in $t = 0$
- $B_i^n(t)$ hat eine $(n-i)$ -fache Nullstelle in $t = 1$
- $\sum_{i=0}^n B_i^n(t) = 1 \quad \forall t$

Formeigenschaften

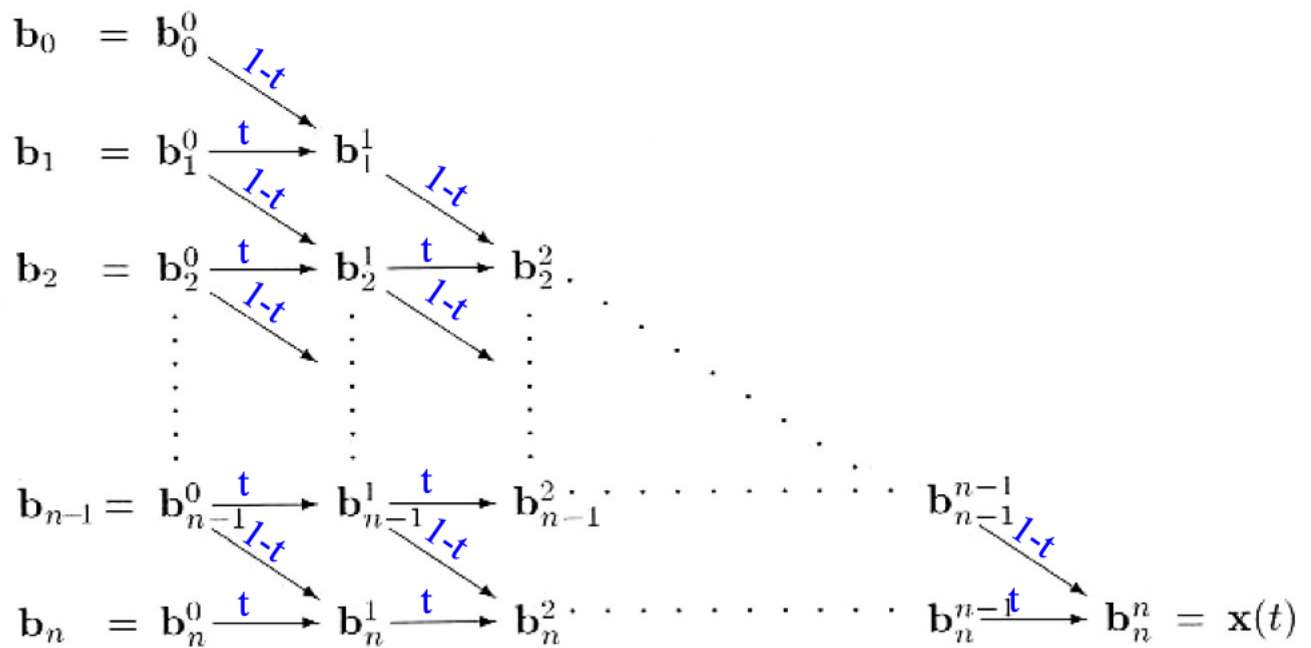
1. Interpolation der Endpunkte
2. In den Endpunkten tangential an das Kontrollpolygon
3. Bezier-Kurve liegt in der konvexen Hülle der Kontrollpunkte
4. Affine Invarianz
5. Variationsreduzierend

Auswertung

Horner-Bezier

$$\begin{aligned}
 C(t) &= \sum_{i=0}^n b_i B_i^n(t) \\
 &= \sum_{i=0}^n b_i \binom{n}{i} (1-t)^{n-i} t^i \\
 &= (1-t)^n (\tilde{b}_0 + \tilde{b}_1 (\frac{t}{1-t})^1 + \cdots + \tilde{b}_n (\frac{t}{1-t})^n) \\
 &= t^n (\tilde{b}_0 (\frac{1-t}{t})^n + \cdots + \tilde{b}_{n-1} (\frac{1-t}{t})^1 + \tilde{b}_n)
 \end{aligned}$$

de Casteljau



Durch die Methode der Subdivision mit Hilfe von de Casteljau können Kurven durch kleine Linien sehr effizient dargestellt werden

Durch Rekursion mit Startpunkt in der Mitte konvergiert de Casteljau sehr schnell

Glatte Übergang zwischen benachbarten Kurven

- Stetigkeit falls: $b_n = c_0$
- Tangenten im Punkt $b_n = c_0$ sind gegeben durch
 - $C'(1) = n(b_n - b_n 1)$
 - $B'(0) = n(c_1 - c_0)$

Glatt, falls $b_{n-1}, b_n = c_0, c_1$ kollinear sind

Tensor-Produkt-Bezier-Flächen

Allgemein

$$F(s, t) = \sum_{k=0}^m \sum_{i=0}^n b_{ik} B_i^n(s) B_k^m(t) = \sum_{i=0}^n \left(\sum_{k=0}^m b_{ik} B_k^m(t) \right) B_i^n(s)$$

Die Kontrollpunkte hängen von t ab

$$d_i(t) = \sum_k b_{ik} B_k^m(t)$$

Kurven mit $s = \text{const}$ sind Bezier-Kurven in t

$$F(s, t) = \sum_k c_k(s) B_k^m(t) \text{ mit } c_k(s) = \sum_i b_{ik} B_i^n(s)$$

Alle Formeigenschaften, außer der Variationsreduktion übertragen sich von den Bezier-Kurven

Auswertung

1D Version: Zuerst eindimensional in erste Richtung mit de Casteljau, dann in die andere.

Verallgemeinerung der bilinearen Interpolation: Coons-Patch

$$P_1 = C_W(0) = C_S(0)$$

$$P_2 = C_O(0) = C_S(1)$$

$$P_3 = C_N(0) = C_W(1)$$

$$P_4 = C_N(1) = C_O(1)$$

$$F_{st}(s, t) = (1-s)(1-t)P_1 + s(1-t)P_2 + (1-s)tP_3 + stP_4$$

SVD

$$A = U \Sigma V^T$$

- Σ ist Diagonalmatrix, $\sigma_{11} \geq \sigma_{22} \geq \dots \geq 0$
- U und V sind orthogonal

- Die Spalten von U bzw. V sind EV von AA^T bzw. $A^T A$
- $\sigma_{kk} = \sqrt{\lambda_k}$ von A
- $U \in \mathbb{R}^{m \times m}, \Sigma \in \mathbb{R}^{m \times n}, V \in \mathbb{R}^{n \times n}$

Informationen

$$\text{rang}(A) = r$$

Bild

$$\text{im}(A) = \langle u_1, \dots, u_r \rangle$$

Kern

$$\ker(A) = \langle v_{r+1}, \dots, v_n \rangle$$

Norm

$$\|A\|_2 = \sigma_{11}$$

Lösungstheorie

- $n = m$ und $\det(A) \neq 0$: eindeutige Lösung
- $n = m$ und $\det(A) = 0$ oder $n \neq m$:
 - nur lösbar, falls $b \in \text{im}(A)$
 - alle Lösungen: $x_0 + \ker(A)$ wobei x_0 eine spezielle Lösung ist

Pseudo-Inverse

$$A^{\sim 1} = V \Sigma^{\sim 1} U^T$$

wobei

$$\Sigma^{\sim 1} = \begin{pmatrix} \frac{1}{\sigma_1} & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & \frac{1}{\sigma_r} & \dots & 0 \\ \vdots & & \vdots & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

Lösen

- A hat maximalen Rang ($\text{rank}(A) = \min\{n, m\}$)
 - überbestimmtes System ($n < m$)
 $x = A^{\sim 1} b$ löst $\|Ax - b\| = \min$
 - unterbestimmtes System ($n > m$)
 $x = A^{\sim 1} b$ löst $Ax = b$ und erfüllt $\|x\|_2 = \min$
- $\text{rank}(A) < \min\{n, m\}$

- $x = A^{\sim}b$ minimiert $\|Ax - b\|_2 = \min$ das Residuum und
- ist unter allen diesen Lösungen diejenige mit der kleinsten Norm $\|x\|_2 = \min$

Iterative Verfahren