Gleitkomma-Arithmetik

Für $e_{min}, e_{max} \in \mathbb{Z}$, $e_{min} < e_{max}$ ist ein Gleitkommasystem wie folgt definiert:

$$\begin{aligned}\mathcal{F} &= \mathcal{F}(\beta, t, e_{min}, e_{max}) \\ &= \{\pm m\beta^{e-t} | m \in \mathbb{N}, \beta^{t-1} \leq m \leq \beta^t - 1 \vee m = 0, \\ &\qquad e_{min} \leq e \leq e_{max}\}\end{aligned}$$

$x \in \mathcal{F} \setminus \{0\} \Rightarrow \beta^{e_{min}-1} \leq |x| \leq \beta^{e_{max}}(1-\beta^{-1})$.

Normalisierte Darstellung

Für $d_1 \neq 0$, $0 < d_1 \leq \beta - 1$:
$x = \pm\beta^e(\frac{d_1}{\beta^1} + \frac{d_2}{\beta^2} + \cdots + \frac{d_t}{\beta^t}) =: \pm 0.d_1 d_2 \cdots d_t \cdot \beta^e$

Relative Maschinengenauigkeit

$fl(x) \in \mathcal{F}$ ist die $x \in \mathbb{R}$ am nächsten liegende Gleitkommazahl.
Für relative Maschinengenauigkeit $\epsilon := \frac{1}{2}\beta^{1-t}$:
$\frac{|fl(x)-x|}{|x|} < \epsilon$, $\frac{|fl(x)-x|}{|fl(x)|} \leq \epsilon$

Arithmetische Grundoperationen

Für $x, y \in \mathcal{F}$ sind Operationen $o \in \{x, -, *, \div\}$ bzgl. eines Gleitkommasystems definiert:
$\tilde{o}(x,y) := fl(o(x,y))$
Zu beachten ist hier die Ungültigkeit der Assoziativ- und Distributivgesetze.

Kondition mathematischer Probleme

Ein mathematisches Problem (f, x) ist die Auswertung von $f(x)$ an $x \in E$ wobei $f : E \subset X \to R \subset Y$.
X und Y sind normierte Räume, E Menge der Eingaben, R Menge der Resultate.
Ein Algorithmus für f ist Abbildung $\tilde{f} : E \subset X \to Y$ s.d. $\tilde{f}(x)$ in endlich vielen Schritten auswertbar ist und $\tilde{f}(x) \approx f(x)$ gilt.
Die Konditionszahl eines mathematischen Problems (f, x) ist die kleinste Zahl $\kappa_f(x) \geq 0$ mit:

$$\frac{\|f(x+\Delta x) - f(x)\|_Y}{\|f(x)\|_Y} \leq \kappa_f(x)\frac{\|\Delta x\|_X}{\|x\|_X} + o(\|\Delta x\|_X)$$

Für $\|\Delta x\|_X \to 0$.

$$\kappa_f(x) = \limsup_{\delta \to 0}\left\{\frac{\|f(x+\Delta x) - f(x)\|_Y \|x\|_X}{\|f(x)\|_Y \|\Delta x\|_X}\right\}$$

Für $\Delta x \in X$, $x + \Delta x \in E$, $\|\Delta x\|_X \leq \delta$.
Ein Problem (f, x) ist gut konditioniert für *kleine* und schlecht konditioniert für *große* Konditionszahlen $\kappa_f(x)$.
Existiert $\limsup$ nicht wird $\kappa_f(x) = \infty$ gesetzt und das Problem als *schlecht gestellt* bezeichnet.

Kondition stetig differenzierbarer Fkt.

Für $f \in \mathcal{C}^1(E, \mathbb{R}^m)$ in Umgebung $E \subseteq \mathbb{R}^n$ von x:

$$\kappa_f(x) = \frac{\|f'(x)\|_\infty \cdot \|x\|_X}{\|f(x)\|_Y}$$

Stabilität numerischer Algorithmen

Elementaroperationen eines Computers sind mit dem relativen Fehler ϵ behaftet. Zusätzlich existiert ein Eingabefehler derselben Größenordnung. Es ist also in jedem Fall mit einem relativen Fehler $\kappa_f(x)\epsilon$ zu rechnen.

Vorwärtsanalyse

Stabilitätsindikator der Vorwärtsanalyse eines Algorithmus' $\tilde{f}$ zur Lösung von (f, x) ist minimales $\sigma = \sigma(x) \geq 0$ für $\{x^\epsilon\}_{\epsilon>0}$ mit $\|x - x^\epsilon\|_X \leq \epsilon\|x\|_X$:

$$\frac{\|\tilde{f}(x^\epsilon) - f(x^\epsilon)\|_Y}{\|f(x^\epsilon)\|_Y} \leq \sigma\kappa_f(x^\epsilon)\epsilon + o(\epsilon) \text{ für } \epsilon \to 0$$

Algorithmus $\tilde{f}$ ist stabil im Sinne der Vorwärtsanalyse, wenn $\sigma \leq$ #Elementaroperationen.

Rückwärtsanalyse

Stabilitätsindikator der Rückwärtsanalyse ist minimales $\rho = \rho(x) \geq 0$ für $\{x^\epsilon\}_{\epsilon>0}$ s.d. für bel. $\|x^\epsilon - x\|_X \leq \epsilon\|x\|_X$ Schar $\{\hat{x}^\epsilon\}_{\epsilon>0}$ ex. mit $f(\hat{x}^\epsilon) = \tilde{f}(x^\epsilon)$:

$$\frac{\|\hat{x}^\epsilon - x^\epsilon\|_X}{\|x^\epsilon\|_X} \leq \rho\epsilon + o(\epsilon) \text{ für } \epsilon \to 0$$

Vorwärtsstabilität folgt aus Rückwärtsstabilität.

Vektor- und Matrixnormen

Induzierte Matrixnorm / Operatornorm

Für Normen $\|\cdot\|_\circ$, $\|\cdot\|_\star$ auf $\mathbb{K}^n$ bzw. $\mathbb{K}^m$ ist eine Matrixnorm $\|\cdot\| : \mathbb{K}^{m\times n} \to [0, \infty)$ auf dem Vektorraum der $m \times n$-Matrizen definiert:

$$\|A\| := \max_{v \in \mathbb{K}^n \setminus \{0\}} \frac{\|Av\|_\star}{\|v\|_\circ} = \max_{\{v \in \mathbb{K}^n | \|v\|_\circ = 1\}} \|Av\|_\star$$

Eigenschaften

Für $A \in \mathbb{K}^{m\times n}$ gilt $\forall v \in \mathbb{K}^n : \|Av\|_\star \leq \|A\| \cdot \|v\|_\circ$
Submultiplikativität: $\|AB\| \leq \|A\| \cdot \|B\|$

Matrix-p-Normen

Induzierte Matrixnorm bei Wahl der p-Normen über $\mathbb{K}^n$ bzw. $\mathbb{K}^m$:

$$\|A\|_p := \max_{v \in \mathbb{K}^n} \frac{\|Av\|_p}{\|v\|_p} = \max_{\|v\|_p = 1} \|Av\|_p \text{ für } 1 \leq p \leq \infty$$

Spaltensummennorm

Für $A = (a_1, \cdots, a_n)$ mit $a_j \in \mathbb{K}^m$:

$$\|A\|_1 = \max_{1\leq j\leq n} \|a_j\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{m} |a_{i,j}|$$

Zeilensummennorm

$$\|A\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{n} |a_{i,j}|$$

Spektralnorm

Die Matrix-2-Norm wird so genannt, da $\|A\|_2 = \sqrt{\lambda_{max}(A^H A)}$ für $\lambda_{max}(A^H A)$ als Bezeichner des größten Eigenwerts von $A^H A \in \mathbb{K}^{n\times n}$.
$\|A\|_2 = \|A^H\|_2$, $\|A^H A\|_2 = \|A\|_2^2$
$\|QA\|_2 = \|A\|_2$ für unitäre Q.

Induzierte Normen

Für $A \in \mathbb{R}^{n\times n}$ mit $A > 0$ ist $\langle z, z\rangle_A := \langle Az, z\rangle_2$ ein Skalarprodukt auf $\mathbb{R}^n$. Dieses induziert die Energienorm $\|z\|_A = \sqrt{\langle z, z\rangle_A}$.

Kondition einer Matrix

Für $A \in \mathbb{K}^{n\times n} \in GL_n\mathbb{R}$, $\|\cdot\|$ induzierte Matrixnorm:

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|$$

$$1 = \|Id\| = \|AA^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = \kappa(A)$$

Besondere Matrizen

Diagonaldominante Matrizen

$A \in \mathbb{R}^{n\times n}$ ist diagonaldominant, falls:
$\forall i \in \{1, \cdots, n\} : |a_{i,i}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}|$
Insbesondere sind solche A regulär. Gilt nur $\geq$ so heißt die Matrix schwach diagonaldominant.

Positiv definite Matrizen

$A \in \mathbb{R}^{n\times n}$ ist positiv definit d.h. $A > 0$ falls $A = A^T$ und $\forall x \in \mathbb{R}^n \setminus \{0\} : x^T Ax > 0$.

Hessenberg-Matrizen

Fast obere / untere Dreiecksmatrix wobei 1. untere / obere Nebendiagonale besetzt sein kann.

Neumannsche-Reihe

$$(Id - M)^{-1} = \sum_{k=0}^{\infty} M^k$$

Bezüglich $A > 0$ konjugierte Vektoren

Vektoren $p, q \in \mathbb{R}^n$ sind konjugiert bzgl. $A > 0$ d.h. A-orthogonal, falls $Ap \perp q$, also $\langle Ap, q\rangle_2 = \langle p, q\rangle_A = 0$.

Direkte Verfahren zur LGS Lösung

Cramersche Regel

Sei $A = (a_{i,j})_{ij} \in GL_n(\mathbb{R})$, $b \in \mathbb{R}^n$, $A[j] = (a_1, \cdots, a_{j-1}, b, a_{j+1}, \cdots, a_n) \in \mathbb{R}^{n\times n}$, a_k k-ter Spaltenvektor von A. Dann bildet $x_j = \frac{det(A[j])}{det(A)}$ für $j = 1, \cdots, n$ die eindeutige Lösung $x \in \mathbb{R}^n$ s.d. $Ax = b$.
Aufgrund des hohen Aufwands von allg. mehr als $(n+1)!$ arithmetischen Operationen ist die Cramersche Regel nur von theoretischer Bedeutung.

Lösung gestaffelter Systeme

Obere Dreicksmatrizen können mittels Rückwärtssubstitution, untere Dreiecksmatrizen mittels Vorwärtssubstitution in $\mathcal{O}(n^2)$ gelöst werden.

LR-Zerlegung

$A = LR$ wobei L untere Dreiecksmatrix mit 1-Diagonale und R obere Dreicksmatrix.

Berechnung LR-Zerlegung

Die LR-Zerlegung existiert insofern die Diagonaleinträge nicht verschwinden. Insbesondere gilt dies für diagonaldominante Matrizen.

1. Spaltenweises nullen der der unteren Einträge mittels *Gauß*, Matrizen $L_1, \cdots, L_{n-1}$
2. $L = L_1^{-1} \cdots L_{n-1}^{-1}$, $R = L_{n-1} \cdots L_1 A$

Lösung $Ax = b$ mittels LR-Zerlegung

1. $A = LR$ berechnen
2. $Lz = b$ Vorwärtssubstitution
3. $Rx = z$ Rückwärtssubstitution

Spaltenpivotsuche

Die normale LR-Zerlegung ist nur Vorwärts- und nicht Rückwärtsstabil. Dies kann durch Spaltenpivotsuche verbessert werden. Hier wird in jedem Schritt mittels einer Permutationsmatrix immer mit der größten verbleibenden Zeile eliminiert.

Für alle regulären Matrizen existiert eine Spaltenpivotsuchen LR-Zerlegung so, dass $PA = LR$.

Cholesky-Zerlegung

Für symmetrische $A > 0$ existiert untere Dreiecksmatrix L mit positiver Diagonale, so dass $A = LL^T$.
Sym. $A > 0$ können eindeutig als $A = LDL^T$ geschrieben werden s.d. L untere Dreiecksmatrix mit 1-Diagonale, D positive Diagonalmatrix. $D = D^{1/2}D^{1/2}$ und $G = LD^{1/2}$ ergibt die äquivalente Zerlegung $A = GG^T$.

QR-Faktorisierung

Für alle $A \in \mathbb{R}^{m\times n}$ mit $m \geq n$ und $Rang(A) = n$ existiert $A = QR$ mit unitärem $Q \in \mathbb{R}^{m\times m}$ und injektiver oberer Dreiecksmatrix R.

Householder-Reflexionen

$$H(v) := Id_m - 2\frac{vv^T}{v^Tv} = Id_m - 2\frac{vv^T}{\|v\|_2^2} \text{ für } \forall v \in \mathbb{R}^m \backslash \{0\}$$

Solche $H(v)$ sind orthogonal und symmetrisch, d.h. $H(v)^TH(v) = Id_m$ und $H(v)^2 = Id_m$.
Wegen $H(v)v = v - 2v = -v$ und $\forall w \in spann\{v\}^\perp$: $H(v)w = w$ ist $H(v)$ Spiegelung an der Hyperebene $spann\{v\}^\perp$.
Solche Reflexionen können durch wiederholte Anwendung Matrizen in obere Dreiecksgestalt überführen:

Gesucht ist $v \in \mathbb{R}^m$ für $y \in \mathbb{R}^m$ s.d.:

$$H(v)y = y - 2\frac{\langle v, y\rangle}{\|v\|_2^2}v \stackrel{!}{=} \alpha e_1$$

Vermeidung von Auslöschung: $\alpha := -sign(y_1)\|y\|_2$
Es ergibt sich mit $v := y - \alpha e_1$: $H(y - \alpha e_1)y = \alpha e_1$.

Seien Q_k die sukzessiven, auf $m \times m$ erweiterten, Householder-Reflexionen. Dann gilt:

$R := Q_p \cdots Q_1 A$, $Q := Q_1^T \cdots Q_p^T$ s.d. $A = QR$.

Givens-Rotationen

Mit $c^2 + s^2 = 1, c, s \in \mathbb{R}$ und $l < k$:

$$G(l,k) := \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}$$

Wobei c das Diagonalelement der l-ten und k-ten Zeile, s k-tes Element der l-ten Zeile, $-s$ l-tes Element der k-ten Zeile.
Givens-Rotationen sind orthogonal und nicht symmetrisch. $G(l,k)A$ unterscheidet sich von A nur in der l-ten und k-ten Zeile.

$$(G(l,k)x)_i = \begin{cases} cx_l + sx_k & i = l \\ -sx_l + cx_k & i = k \\ x_i & \text{sonst} \end{cases}$$

$\exists \varphi \in (0, 2\pi] : c = \cos\varphi, s = \sin\varphi$ d.h. $G(l,k)$ ist Rotation um φ in Ebene $spann\{e_l, e_k\}$.
Ziel: k-te Komponente von x nullen für $x_l^2 + x_k^2 \neq 0$.

$$|x_l| > |x_k| : \tau := \frac{x_k}{x_l}, c := \frac{1}{\sqrt{1+\tau^2}}, s := c\tau$$
$$|x_l| \leq |x_k| : \tau := \frac{x_l}{x_k}, s := \frac{1}{\sqrt{1+\tau^2}}, c := s\tau$$

Mit einer solchen Givens-Rotation können einzelne Matrixelemente genullt und $A \in \mathbb{R}^{m\times n}$ so sukzessive in eine obere Dreiecksmatrix transformiert werden.
Für Hessenberg-Matrix A ergibt sich $A = QR$ mit: $Q^T := G(n-1, n)\cdots G(1,2)$ und $R := Q^TA$.

QR mit Householder ist ungefähr doppelt so schnell wie mit Givens. Diese sind daher nur bei strukturierten Matrizen wie Tridiagonal- oder Hessenberg-Matrizen sinnvoll einzusetzen.

Lineare Ausgleichsprobleme

Finde $u^* \in \mathbb{R}^n$: $\|Au^* - b\|_2 = \min_{u\in\mathbb{R}^n}\|Au - b\|_2$
Es sind äquivalent:

(a) u^* löst Ausgleichsproblem
(b) u^* löst Normalengleichung $A^TAu^* = A^Tb$
(c) u^* erfüllt $Au^* = P_Ab$ mit Ortogonalprojekt. $P_A : \mathbb{R}^m \to \mathbb{R}^m$ auf Bild von A

Das Residuum $Au^* - b$ steht normal zu Bild von A.
Ein Ausgleichsproblem ist eindeutig lösbar gdw. $Kern(A) = \{0\}$.
Die Lösung mit minimaler euklidischer Norm wird Minimum-Norm-Lösung u^+ genannt.

Singulärwertzerlegung

Sei $A \in \mathbb{R}^{m\times n}$ mit $r = Rang(A) \leq \min\{m, n\}$.
Dann existiert $A = USV^T$ mit orthogonalen $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ sowie Diagonalmatrix $S \in \mathbb{R}^{m\times n}$.

Iterative Verfahren zur LGS Lösung

Ein iteratives Verfahren zur Lösung von $Au = b$ liefert zu Startvektor $u^0 \in \mathbb{R}^n$ eine Folge $\{u^k\}_{k\in\mathbb{N}_0} \subset \mathbb{R}^n$ mittels $\Psi_k : (\mathbb{R}^n)^{k+1} \to \mathbb{R}^n$ durch $u^{k+1} = \Psi_k(u^0, \cdots, u^k)$.
Ein Verfahren konvergiert falls $\forall u^0 \in \mathbb{R}^n, b \in \mathbb{R}^n$: $\lim_{k\to\infty} u^k = A^{-1}b$

Abbruchkriterium

Direkte Löser liefern bei exakter Arithmetik nach endlich vielen Schritten $A^{-1}b$. Bei iterativen Lösern gilt im allg. $\forall k \in \mathbb{N} : u^k \neq A^{-1}b$.
Ein Abbruchkriterium mit Toleranz $\tau > 0$ ist z.B. $\|u^m - u^{m-1}\| \leq \tau$ oder $\frac{\|Au^m - b\|}{\|b\|} \leq \tau$ (rel. Residuum)

Lineare Iterationsverfahren

$u^{k+1} = u^k + N(b - Au^k) = (Id - NA)u^k + Nb$ zu $u^0 \in \mathbb{R}^n$ wobei $N \in GL_n(\mathbb{R})$ das jeweilige Verfahren charakterisiert.
$M := Id - NA$ heißt Iterationsmarix.

Spektralradius

$\rho(M) = \max\{|\lambda| : \lambda \in Spec(M)\}$

Konvergenz

Ein lineares Iterationsverfahren konvergiert gdw. Spektralradius von M $\rho(M) < 1$.
Ist $\|M\| < 1$ für induzierte Matrixnorm $\|\cdot\|$ dann konvergiert das entsprechende lineare Iterationsverfahren ebenso.
Sind A oder A^T diagonaldominant so konvergieren sowohl das Jacobi- als auch das Gauß-Seidel-Verfahren.

Gesamtschritt- / Jacobi-Verfahren

$$u_i^{k+1} = u_i^k + \frac{1}{a_{i,i}}\left(b_i - \sum_{j=1}^{n} a_{i,j}u_j^k\right) \text{ für } i = 1, \cdots, n$$

Für $A = D - L - R$:

$$\begin{aligned} u^{k+1} &= u^k + D^{-1}(b - Au^k) \\ &= (Id - D^{-1}A)u^k + D^{-1}b \end{aligned}$$

Also: $M^{Jac} = Id - D^{-1}A = D^{-1}(L + R)$, $N^{Jac} = D^{-1}$

Einzelschritt- / Gauß-Seidel-Verfahren

$$u_i^{k+1} = u_i^k + \frac{1}{a_{i,i}}\left(b_i - \sum_{j=1}^{i-1} a_{i,j}u_j^{k+1} - \sum_{j=i}^{n} a_{i,j}u_j^k\right)$$

Für $A = D - L - R$:
$M^{GS} = (D - L)^{-1}R$, $N^{GS} = (D - L)^{-1}$.

Krylov-Raum-Verfahren

Der Krylov-Raum m-ter Ordnung bzgl. B und v:

$$\mathcal{U}_m(B, v) := spann\{v, Bv, B^2v, \cdots, B^{m-1}v\} \subset \mathbb{R}^n$$

Das Residuuum der k-ten Iterierten u^k:

$$r^k = (I - AN)r^{k-1} = (I - AN)^k r^0$$

Es gilt: $u^k \in V_k$ mit $V_k = u_0 + \mathcal{U}_k(NA, Nr^0)$ sowie $r^0 = b - Au^0$.
Minimaleigenschaft der k-ten Iterierten bzgl. Norm $\|\cdot\|_\star$ auf $\mathbb{R}^n$:

$$u^k = argmin\{\|v - A^{-1}b\|_\star : v \in V_k\}$$

Ein Krylov-Raum-Verfahren bzgl. einer Norm $\|\cdot\|_\star$ ist nur dann sinnvoll, wenn u^k mit geringem, d.h. im Bereich von zwei Matrix-Vektormultiplikationen liegendem, numerischen Aufwand aus u^{k-1} hervorgeht.

$$u^{k+1} = u^k + N(b - Au^k)$$

u^k wird in jeder Iteration entsprechend der jeweiligen Charakterisierung minimiert.

Vorkonditionierer

Anforderungen: Nv sollte schnell zu berechnen sein, weiterhin sollte $N \approx A^{-1}$ möglichst gelten.

CG-Verfahren

Das Verfahren der konjugierten Gradienten ist durch die Energienorm charakterisiert und definiert als:

$$u^k = argmin\{\|v - A^{-1}b\|_A : v \in V_k\}$$

Für positiv definite $A, N \in \mathbb{R}^{n\times n}$ sowie $b \in \mathbb{R}^n$ liefert das CG-Verfahren nach spätestens n Schritten $A^{-1}b$. Eigentlich ist es bei exakter Arithmetik also ein direkter Verfahren, wird aber durch früheren Abbruch als iteratives Verfahren verwendet.

Umsetzung

Minimum $\hat{x}$ von $f(x) = \frac{1}{2}x^TAx - b^Tx$ ist $A\hat{x} = b$.

$\nabla f(x) = Ax - b \stackrel{!}{=} 0$ ergibt Minimum da nach oben geöffnete Parabel wg. $A > 0$.
Initiale Suchrichtung p_0 ist $r_0 = b - Ax_0$. Folgende p_{k+1} ergeben sich aus $p_{k+1} = r_{k+1} - \beta_k p_k$.
p_k und x_k sind A-konjugiert. $f(x_k + \alpha_k p_k)$ wird via α_k entlang dieser Richtung minimiert mittels:

$$\alpha_k = \frac{\langle r_k, p_k\rangle_2}{\|p_k\|_A^2} = \frac{\langle r_k, r_k\rangle_2}{\|p_k\|_A^2}$$
$$\beta_k = \frac{\langle r_{k+1}, p_k\rangle_A}{\|p_k\|_A^2} = \frac{\langle r_{k+1}, r_{k+1}\rangle_2}{\|r_k\|_2^2}$$

Es gilt also $x_{k+1} = x_k + \alpha_k p_k$ und $r_{k+1} = r_k - \alpha_k Ap_k$.

GMRES-Verfahren

Das Verfahren des verallgemeinerten minimalen Residuum liefert die Lösung $Ax = b$ für $A \in GL_n(\mathbb{R})$ und ist charakterisiert durch:

$$\begin{aligned} u^k &:= argmin\{\|N(b - Av)\|_2 : v \in V_k\} \\ &= argmin\{\|NA(A^{-1}b - v)\|_2 : v \in V_k\} \end{aligned}$$

Das GMRES-Verfahren ist also durch die nicht Skalarprodukt-induzierte Norm $\|NA\cdot\|_2$ induziert.

Interpolation

Interpolation von f mit φ s.d. $\varphi(t_i) = f(t_i)$ für $i = 0, \cdots, n$. Approximation von f mit φ s.d. $\|\varphi - f\|$ möglichst klein in geeigneter Norm.

Klassische Polynom-Interpolation

Zu gegebenen Knoten $t_0 < \cdots < t_n$ und Stützwerten $f_i = f(t_i)$ für $i = 0, \cdots, n$ wird Polynom $\in \Pi_n$ gesucht s.d. $P(t_i) = f_i$ für $i = 0, \cdots, n$.
Zu $n+1$ Stützwerten f_i und paarweise verschiedenen Knoten t_i existiert dabei genau ein solches Interpolationspolynom $P = P(f|t_0, \cdots, t_n) \in \Pi_n$.

Interpolationsfehler

$$\|f - P(f|t_0, \cdots, t_n)\|_\infty \leq \sup_{\tau \in [a,b]} \frac{|f^{(n+1)}(\tau)|}{(n+1)!} \|\omega_{n+1}\|_\infty$$

$\omega_{n+1} \in \Pi_{n+1}$ ist das *Newton-Polynom* bzgl. $t_0, \cdots, t_n$ mit $\omega_{n+1}(t) := \prod_{i=0}^{n}(t - t_i)$.

Vandermonde-Matrix

$$\begin{pmatrix} 1 & t_0 & t_0^2 & \cdots & t_0^n \\ 1 & t_1 & t_1^2 & \cdots & t_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_n & t_n^2 & \cdots & t_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ \vdots \\ \vdots \\ f_n \end{pmatrix}$$

Die Lösung der Vandermonde-Matrix beschreibt $P(f|t_0, \cdots, t_n) \in \Pi_n$, was jedoch zu aufwändig ist.

Lagrange-Basis

Basis $\{L_{n,0}, \cdots, L_{n,n}\}$ von Π_n abhg. $t_0 < \cdots < t_n$ wg.

$$L_{n,k}(t_i) = \delta_{k,i} = \begin{cases} 1 & k = i \\ 0 & sonst \end{cases}$$

$$L_{n,k}(t) := \prod_{j=0, j \neq k}^{n} \frac{t - t_j}{t_k - t_j}$$

Es gilt also: $P(f|t_0, \cdots, t_n) = \sum_{k=0}^{n} f_k \cdot L_{n,k}$

Ein Lagrange Polynom zu Stützstelle t_k nimmt an dieser 1, an allen anderen Stützstellen 0 an.

Lemma von Aitken

$P = P(f|t_0, \cdots, t_n)(t) =$

$$\frac{(t_0 - t)P(f|t_1, \cdots, t_n)(t) - (t_n - t)P(f|t_0, \cdots, t_{n-1})(t)}{t_0 - t_n}$$

Schema von Neville

Sei $t \in \mathbb{R}$ fest, $P_{i,k}(t) = P_{i,k} = P(f|t_{i-k}, \cdots, t_i)(t)$ für $i \geq k$. Dann ist insb. $P_{i,0} = f_i$ und $P_{n,n} = P(f|t_0, \cdots, t_n)(t)$ kann rekursiv mit dem *Schema von Neville* berechnet werden:

$$P_{i,k} = \frac{(t_{i-k} - t)P_{i,k-1} - (t_i - t)P_{i-1,k-1}}{t_{i-k} - t_i} = \frac{t - t_{i-k}}{t_i - t_{i-k}} P_{i,k-1} - \frac{t - t_i}{t_i - t_{i-k}} P_{i-1,k-1}$$

Tschebyscheff-Knoten

Für $i = 0, \cdots, n$:

$$t_i^{[a,b]} = \frac{b+a}{2} + \frac{b-a}{2} \cos\left(\frac{2i+1}{2n+2}\pi\right)$$

Diese Knotenfolge liegt dichter zu den Intervallgrenzen hin und ergibt eine bessere Interpolation als äquidistante Knoten.

Satz von Faber

Zu jeder Folge von Knoten $\{t_0^{(n)}, \cdots, t_n^{(n)}\}_{n \in \mathbb{N}}$ in $[a,b]$ gibt es ein $f \in C([a,b])$ so, dass $\{P(f|t_0^{(n)}, \cdots, t_n^{(n)})\}_{n \in \mathbb{N}}$ für $n \to \infty$ nicht glm. gegen f konvergiert.

Splines

Nachteile der Polynom-Interpolation bei einer größeren Anzahl von Knoten:

1. Starke Oszillation des Polynoms
2. Konvergenz des Polynoms gegen die interpolierte Funktion ist nicht gewährleistet

Sei $\Delta = \{t_0, \cdots, t_{l+1}\}$ ein Gitter paarweise verschiedener Knoten $a = t_0 < \cdots < t_{l+1} = b$.
$s \in C^{k-2}(a,b)$ ist Spline der Ordnung $k \in \mathbb{N}$ bzgl. Δ wenn sie auf jedem Interval $[t_i, t_{i+1}]$ mit einem Polynom $s_i \in \Pi_{k-1}$ übereinstimmt.

Spline-Raum

$S_{k,\Delta}$ ist Raum aller Splines der Ordnung k bzgl. Δ. Der Spline-Raum $S_{k,\Delta}$ ist ein reeller Vektorraum mit $\Pi_{k-1}[a,b] \subset S_{k,\Delta}$.
Zusätzlich gilt auch $(t - t_i)_+^{k-1} \in S_{k,\Delta}$

Abgebrochene Potenzen

Abgebrochene Potenzen vom Grad $k-1$:

$$(t - t_i)_+^{k-1} := \begin{cases} (t - t_i)^{k-1} & : t \geq t_i \\ 0 & : t < t_i \end{cases}$$

Für $t_i \in \Delta$, $i \neq l+1$

Basis des Spline-Raumes

$$\mathcal{B} = \{1, t, \cdots, t^{k-1}, (t - t_1)_+^{k-1}, \cdots, (t - t_l)_+^{k-1}\}$$

ist eine Basis von $S_{k,\Delta}$ mit $\dim(S_{k,\Delta}) = k + l$.

Spline-Interpolation

Interpolation einer Funktion f bzgl. eines Gitters $\Delta = \{t_0, \cdots, t_{l+1}$ durch Spline der Ordnung k.

Im linearen Fall mit $k = 2$ stimmt die Anzahl der Knoten $l + 2$ mit $\dim(S_{2,\Delta}) = l + 2$ überein. Es gibt also genau einen Spline der $(t_i, f(t_i))$ interpoliert.

$$I_2 f = \sum_{i=0}^{l+1} f(t_i) B_i$$

Für $B_i \in S_{2,\Delta}$ mit $B_i(t_k) = \delta_{i,k}$

Kubische Splines

Kubische Splines der Ordnung 4 eigenen sich für die Darstellung von Kurven, da das menschliche Auge diese als glatt empfindet.

Die Interpolationsbedingungen reichen zur eindeutigen Bestimmung eines interpolierenden Spline aus $S_{4,\Delta}$ nicht aus.
Wegen $\dim(S_{4,\Delta}) - (l+2) = l + 4 - (l+2) = 2$ bleiben zwei Freiheitsgrade unbestimmt.
Eine zusätzliche Bedingung ist, dass der interpolierende kubische Spline die minimale Krümmung aller interpolierenden $\mathcal{C}^2$-Funktionen besitzen soll.

Krümmung einer Funktion

Krümmung von $y : [a,b] \to \mathbb{R}, y \in \mathcal{C}^2$:

$$\kappa(t) := \frac{y''(t)}{(1 + y'(t))^{3/2}}$$

$1/\kappa(t)$ ist der Radius des *Krümmungskreises*.
Das Krümmungsverhalten von y über ganz $[a,b]$ ist durch ein Integral messbar:

$$\|y''\|_2 := \left(\int_a^b y''(t)^2 dt\right)^{1/2}$$

Randbedingungen

Für $s \in S_{4,\Delta}$, $\Delta = \{t_0, \cdots, t_{l+1}\}$ sind mögliche Randbedingungen:

(a) $s'(a) = f'(a)$ und $s'(b) = f'(b)$: *vollständige Spline-Interpolation*
(b) $s''(a) = s''(b) = 0$: *natürliche Interpolation*
(c) $s'(a) = s'(b)$ und $s''(a) = s''(b)$ falls f $b-a$ periodisch: *periodische Interpolation*

Ist eine dieser Randbedingungen erfüllt, so ist s eindeutig bestimmt. Ferner gilt für alle anderen interpolierenden $y \in \mathcal{C}^2(a,b)$: $\|s''\|_2 < \|y''\|_2$

Momente von Splines

Sei $h_{j+1} := t_{j+1} - t_j$ Länge von $[t_j, t_{j+1}]$.
$M_j = s''(t_j)$ für $j = 0, \cdots, l+1$ sind die Momente des Splines $s \in S_{4,\Delta}$. Aus den Momenten kann der Spline vollständig rekonstruiert werden.
Da $s_j := s|_{[t_j, t_{j+1}]}$ kubisch ist gilt für $s_j'' = s''|_{[t_j, t_{j+1}]}$:

$$s_j''(t) = M_j \frac{t_{j+1} - t}{h_{j+1}} + M_{j+1} \frac{t - t_j}{h_{j+1}}$$

$$s_j'(t) = -M_j \frac{(t_{j+1} - t)^2}{2h_{j+1}} + M_{j+1} \frac{(t - t_j)^2}{2h_{j+1}} + A_j$$

$$s_j(t) = M_j \frac{(t_{j+1} - t)^3}{6h_{j+1}} + M_{j+1} \frac{(t - t_j)^3}{6h_{j+1}} + A_j(t - t_j) + B_j$$

Die Integrationskonstanten A_j, B_j lassen sich aus den Interpolationsbedingungen berechnen.

Ein paar Matlab Grundlagen

```
A = [ 1 0 ; 0 1 ] % = eye(2)
b = [ 3 4 ]'      % = [ 3; 4 ]
b(1) = 4          % => b = [ 4 4 ]'
A(2,1) = 2        % => A = [ 1 0 ; 2 1 ]
c = 1:3           % = [ 1 2 3 ]
c = 1:2:6         % = [ 1 3 5 ]
A(2,:)            % = [ 3 1 ]
A(:,1)            % = [ 1 ; 3 ]
A.^2              % = [ 1 0 ; 9 1 ]
```

Beispiel: LR-Zerlegung

```
function [A, L, R] = lr(A)
  [w,h] = size(A);

  if w ~= h
    error('A is not a square matrix')
  end

  for k = 1:w-1
    if abs(A(k,k)) < eps
      error('No LU-decomposition');
    end

    A(k+1:w,k) = A(k+1:w,k) / A(k,k);
    A(k+1:w,k+1:w) = A(k+1:w,k+1:w)
                   - A(k+1:w,k)
                   * A(k,k+1:w);
  end

  L = tril(A,-1) + eye(w);
  R = triu(A);
end
```