Inhaltsverzeichnis

1 Norm und Skalarprodukt

1.1 Norm

Definitheit: $||x|| = 0 \Rightarrow x = 0$
absolute Homogenität: $||\alpha x|| = |\alpha| * ||x||$
Dreiecksungleichung: $||x + y|| \leq ||x|| + ||y||$

1.2 Skalarprodukt

$$\left.\begin{array}{l} < x + y, z >=< x, z > + < y, z > \\ < x, y + z >=< x, y > + < x, z > \\ < \lambda x, y >= \lambda < x, y > \\ < x, \lambda y >= \lambda < x, y > \end{array}\right\} \text{Linearität}$$

$< x, y >=< y, x >$

$$\left.\begin{array}{l} < x, x >\geq 0 \\ < x, x >= 0 \Rightarrow x = 0 \end{array}\right\} positivDefinitheit$$

1.2.1 Vom Skalarprodukt induzierte Norm

$||x|| = \sqrt{< x, x >}$

1.2.2 Cauchy-Schwarzche Ungleichung

$| < x, y > | \leq ||x|| * ||y||$

2 Symmetrische, positiv definite Matrix

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} \begin{pmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{pmatrix}$$

Symmetrische Matrix

insbesonders: Diagonalmatrizen, Einheitsmatrizen
positiv definit: $x^t Ax > 0$ (beliebige Matrix)
alle EW > 0 (symmetrische Matrix)
alle Haupt[TODO: ?] > 0 (symetrische Matrix)
$\begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} \Rightarrow 3$ Hauptminoren[?] $=$ det(a), det $\begin{pmatrix} a & b \\ b & d \end{pmatrix}$, det $\begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}$

2.1 Cholesky-Zerlegung

$A = GG^t$ G unter der Matrix, invertierbar (symmetrische Matrix)

2.2 [?] diagonaldominant und alle Diagonalelemente größer gleich 0

(symmetrische Matrix)

2.3 Eigenwerte

det(λ En - A) = 0

2.4 Eigenvektor

f(v) = λv

3 Matrixnormen

3.1 Natürliche Matrixnorm

$||A||_\infty := \max\limits_{x \neq 0} \frac{||Ax||_\infty}{||x||_\infty} = \max\limits_{||x||=1} ||Ax||_\infty$

$||A|| = 0 \Rightarrow A = 0$
$||\lambda A|| = |\lambda| * ||A||$
$||A + B|| \leq ||A|| + ||B||$
$||A * B|| \leq ||A|| * ||B||$

3.2 Verträglichkeit

$||Ax|| \leq ||A|| * ||x||$

3.3 Zeilensummennorm

= natürliche Matrixnorm

$||A||_\infty = \max\limits_{||x||_\infty = 1} ||Ax||_\infty = \max\limits_{i=1,\ldots,m} \sum_{j=1}^{n} |a_{ij}|$

A = $\begin{pmatrix} 1 & -2 & -3 \\ 2 & 3 & -1 \end{pmatrix}$

$$\begin{aligned} ||A||_\infty &= max\{|1| + |-2| + |-3|, |2| + |3| + |-1|\} \\ &= max\{6, 6\} = 6 \end{aligned}$$

3.4 Spaltensummennorm

$||A||_1 := \max\limits_{x \neq 0} \frac{||Ax||_1}{||x||_1} = \max\limits_{||x||_1 = 1} ||Ax||_1 = \max\limits_{j=1,\ldots,n} \sum_{i=1}^{m} |a_{ij}|$

A = $\begin{pmatrix} 1 & -2 & -3 \\ 2 & 3 & -1 \end{pmatrix}$

$$\begin{aligned} ||A||_1 &= \max\{|1| + |2|, |-2| + |3|, |-3| + |-1|\} \\ &= \max\{3, 5, 4\} = 5 \end{aligned}$$

$||A^t||_1 = ||A||_\infty$

3.5 Spektralnorm

$$\begin{aligned} ||A||_2 &:= \max_{||x||_2=1} ||Ax||_2 \\ &= \max_{x\neq 0} \frac{||Ax||_2}{||x||_2} \\ &= \max_{||x||_2=1} < Ax, Ax > \\ &= \max_{||x||_2=1} < A^t Ax, x > \\ &= max\sqrt{|\lambda|}, \lambda * EW von A^t A \end{aligned}$$

A $= \begin{pmatrix} 3 & 2 \\ -2 & 0 \end{pmatrix}$, $A^tA = \begin{pmatrix} 13 & 6 \\ 6 & 4 \end{pmatrix}$ $\det(\mu E_n - A^tA) = 0 \Leftrightarrow \mu_{1,2} = 16, 1$
$||A||_2 = \sqrt{max(\mu_1, \mu_2)} = \sqrt{\mu_1} = \sqrt{16} = 4$

4 Spektralradius, Konditionszahl einer Matrix

4.1 Spektralradius ρ

$\varphi(A) = max : 1 \leq i \leq n|\lambda_i(A)| = spr(A)$ der betragsmäßig größte Eigenwert von A

$||A|| \geq |\lambda|$ (für jede Matrixnorm, die mit einer Vektornorm verträglich ist)

4.2 Konditionszahl einer Matrix A

$cond(A) = ||A|| * ||A^{-1}||$

4.3 Sonderfall symmetrisch, positiv definite Matrix

$cond(A) = \frac{\lambda_{max}}{\lambda_{min}}$

5 Ähnlichkeitstransformation, Invarianz der Eigenwerte

y = Ax

$$\overline{x} = Cx, \overline{y} = Cy \qquad (\text{det C} \neq 0), C \in GL$$

$y = Ax \Rightarrow C^{-1}\overline{y} = AC^{-1}\overline{x} \Rightarrow \overline{y} = CAC^{-1}\overline{x} \Rightarrow \overline{y}\overline{A}\overline{x}$
$\overline{A} = CAC^{-1} \Rightarrow \overline{A} \sim A$
λ EW, v EV zu A
$\Rightarrow Av = C^{-1}\overline{A}Cv = \lambda v$
$\Rightarrow \overline{A}$ und A haben dieselben Eigenwerte, algebraisch und geometrische Vielfalten stimmen überein (Invarianz der Eigenwerte)

5.1 Reduktionsmethoden

A duch Ähnlichkeitstransformationen
$A = A^{(0)} = T_1^{-1}A^{-1}T_1 = Q... = T_i^{-1}A^{(i)}T_i = ...$
auf Form bringen, für welche EW und EV leicht zu berechnen sind (z.B. Jordan-Normalform)

6 Gleitkommazahlen, ...

6.1 Gleitkommazahl (normalisiert)

$b \in \mathbb{N}, b \geq 2, x \in \mathbb{R}$
$x = \pm m * b^{\pm e}$
Mantisse: $m = m_1 b^{-1} + m_2 b^{-2} + ... \in \mathbb{R}$
Exponent: $e = e_{s-1} b^{s-1} + ... + e_0 b^0 \in \mathbb{N}$
für $x \neq 0$ eindeutig

6.2 Gleitkommagitter

$A = A(b, r, s)$ größte Darstellbare Zahl: $(1 - b^{-r}) * b^{b^s - 1}$
mit b als Basis, r als Mantissenlänge, s als Exponentenlänge
$(b = 10) : 0,314 * 10^1 = 3,14$
$0,123 * 10^6 = 123.000$
Beispiel: konvertiere von Basis 8 zu Basis 10:
$x = (0,5731 * 10^5)_8 \in A(8, 5, 1)$
$x = (5 * 8^{-1} + 7 * 8^{-2} + 3 * 8^{-3} + 1 * 8^{-4}) * 8^5$
$x = 5 * 8^4 + 7 * 8^3 + 3 * 8^2 + 1 * 8^1 = 24.264 * 10^0$

6.3 Maschienengenauigkeit eps

$eps = \frac{1}{2} b^{-r+1}, IEEE : eps = \frac{1}{2} * 2^{-52} \approx 10^{-16}$

6.4 Rundungsfehler

$absolut : |x - rd(x)| \leq \frac{1}{2} b^{-r} b^e$
$relativ : |\frac{x - rd(x)}{x}| \leq \frac{1}{2} b^{-r+1} = eps$

7 Darstellung des Interpolationsfehlers

7.1 Fehler I

f$\in C^{n+1}$[a, b], $\forall$ x $\in$ [a, b] $\exists \xi_x \in (\overline{x_0, ..., x_n, x})$, wobei das Intervall das kleinst mögliche Intervall, das alle x_i enthällt, s.d.

$f(x) - p(x) = \frac{f^{(n+1)}(\xi x)}{(n+1)!} \prod\limits_{j=0}^{n} (x - x_j)$

7.2 Fehler II

$f \in C^{n+1}[a, b], \forall x \in [a, b] \setminus x_0, ..., x_n gilt :$

$f(x) - p(x) = f[x_0, ..., x_n, x] \prod\limits_{j=0}^{n} (x - x_j)$

mit $f[x_i, ..., x_{i+k}] = y[x_i, ..., x_{i+k}]$

und $f[x_0, ..., x_n, x] = \int\limits_0^1 \int\limits_0^{t_1} ... \int\limits_0^{t_n} f^{n+1}(x_0 + t_1(x_1 - x_0) + ... + t_n(x_n - x_{n-1} + t(x - x_n))dt dt_n ... dt_1$

für $x_0 = x_1 = ... = x_n :$

$f[x_0, ..., x_n] = \frac{1}{n!} f^{(n)}(x_0)$

$\frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod\limits_{j=0}^{n} (x - x_j) = f(x) - p(x) = f[x_0, ..., x_n, x] \prod\limits_{j=0}^{n} (x - x_j)$

$\Rightarrow f[x_0, ..., x_n, x] = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}$

8 Konditionierung einer numerischen Aufgabe, Konditionszahlen $k_{i,j}$

8.1 numerische Aufgabe

$x_j \in \mathbb{R}$ mit $f(x_1, ..., x_m) \Rightarrow y_i = f_i(x_j)$
fehlerhafte Eingangsgrößen $x_i + \Delta y_i$
$|\Delta y_i|$ ist der absolute Fehler, $|\frac{\Delta y_i}{y_i}|$ ist der relative Fehler

8.2 Konditionszahl (relativ)

$k_{ij}(x) = \frac{\partial f_i}{\partial x_i}(x)\frac{\Delta x_j}{x_j}$
$\frac{\Delta y_i}{y_i} = \sum_{j=1}^{m} k_{ij}(x)\frac{\Delta x_j}{x_j}$
$|k_{ij}(x)| >> 1 \Rightarrow$ schlecht konditioniert
$|k_{ij}(x)| << 1 \Rightarrow$ gut konditioniert, ohne Fehlerverstärkung
$|k_{ij}(x)| > 1 \Rightarrow$ Fehlerverstärkung
$|k_{ij}(x)| < 1 \Rightarrow$ Fehlerdämpfung

9 Stabilität eines Algorithmus

9.1 stabiler Algorithmus

akkumulierte Fehler der Rechnung (Rundungsfehler, Auswertungsfehler, etc.) übersteigen den unvermeidbaren Problemfehler der Konditionierung der Aufgabe nicht. Aka Trotz Ungenauigkeiten bei den Eingabe Variablen erhalten wir fast sehr genaue Ergebnisse.

10 Auslöschung

Verlust von Genauigkeit bei der Subtraktion von Zahlen mit gleichem Vorzeichen

TODO: bei bedarf ein Beispiel

11 Horner-Schema*

Das sogenannte "Horner-Schema"

$b_n = a_n$, k = n - 1, ..., 0 $b_k = a_k + \xi b_{k+1}$ liefert den Funktionswert $p(\xi) = b_0$ des Polynoms

$p(x) = a_0 + x(... + x(a_{n-1} + a_n x)...)$

11.1 Code

```
def horner(Ac, Ax, n, x):
    y = 0.0
    for i in reversed range(n):
        y = y * (x - Ax[i]) + Ac[i]
    return y
```

Ac: Vektor mit Koeffizienten, ist ein np Array

Ax: Stützstellen, ist ein np Array

n: Anzahl der Stützstellen, ist ein int

x: Auswertungspunkt, ist ein double

Immer Horner-Schema zur Auswertung von Polynomen verwenden.

11.2 Auswertung

TODO: subsection

12 Interpolation und Approximation

12.1 Grundproblem

Darstellung und Auswertung von Funktionen

12.2 Aufgabenstellung

f(x) nur auf Diskreter Menge von Argumenten $x_0, ..., x_n$ bekannt und soll rekonstruiert werden

analytisch gegebene Funktion soll auf Reelwerte dargestellt werden, damit jederzeit Werte zu beliebigen x berechnet werden können.

Einfach konstruierte Funktionen in Klassen P:

Polynome: $p(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$

rationale Funktion: $r(x) = \frac{a_0+a_1x+...+a_nx^n}{b_0+b_1x+...+b_mx^m}$

trigonometrische Funktion: $t(x) = \frac{1}{2}a_0 + \sum\limits_{k=1}^{n}(a_kcos(kx) + b_ksin(kx))$

Exponentialsummen: $e(x) = \sum\limits_{k=1}^{n} a_kexp(b_kx)$

12.3 Interpolation

Zuordnung von $g \in P$ zu f durch Fixieren von Funktionswerten

$g(x_i) = y_i = f(x_i), i = 0, ..., n$

12.4 Approximation

$g \in P$ beste Darstellung, z.B.

$\max\limits_{a \leq x \leq b} |f(x) - g(x)| minimal$

$(\int\limits_a^b |f(x) - g(x)|^2 dx)^{\frac{1}{2}} minimal$

13 Lagransche Interpolationsaufgabe

13.1 Aufgabe

Finde zu n + 1 verschiedene Stützstellen/Knoten $x_0, ..., x_n \in \mathbb{R}$ und Werten $y_0, ..., y_n \in \mathbb{R}$ ein Polynom p $\in P_n mit p(x_i) = y_i$

13.2 Eindeutigkeit + Existenz

Die Lagransche Interpolationsaufgabe ist eindeutig lösbar

TODO: bei bedarf Beweis rein kopieren den Ich nicht verstanden hab

13.3 Lagransche Basispolynome

$L_i^{(n)}(x) = \prod\limits_{j=0, j\neq i}^{n} \frac{x-x_j}{x_i-x_j} \in P_n, i = 0, ..., n$

13.4 Eigenschaften

$$\text{ortogonal: es gilt } L_i^{(n)}(x_k) = d_{ik} = \begin{cases} 1 & i = k \\ 0 & sonst \end{cases}$$

bilden Basis von P_n

haben Grad n

13.5 Lagransche Darstellung

$p(x) = \sum\limits_{i=0}^{n} y_i L_i^{(n)}(x) \in P_n$ mit $p(x_j) = y_j$

Nachteil: Bei Hinzunahme von (x_{n+1}, y_{n+1}) ändert sich das Basispolynom komplett

TODO: Beispiel

14 Newtonsche Basispolynome...

14.1 Newton-Polynome

$N_0(x) = 1, N_i(x) = \prod\limits_{j=0}^{i-1}(x - x_j)$ mit $p(x) = \sum\limits_{i=0}^{n} a_i N_i(x)$

14.1.1 Auswertung

$y_0 = p(x_0) = a_0$
$y_1 = p(x_1) = a_0 + a_1 * (x_1 - x_0)$
$\vdots$
$y_n = p(x_n) = a_0 + a_1(x_1 - x_0) + ... + a_n(x_n - x_0) * ... * (x_n - x_{n-1})$

14.1.2 Vorteil

Bei Hinzunahme von (x_{n+1}, y_{n+1}) muss nur eine neue Rechnung durchgeführt werden, und nicht das gesamte Polynom neu berechnet werden

TODO: Beispiel

14.2 Newtonsche Darstellung(stabile Variante)

$p(x) = \sum\limits_{i=0}^{n} y[x_0, ..., x_i] N_i(x)$

14.3 Dividierte Differenzen*

$y[x_i, ..., x_{k+1}] = \frac{y[x_{i+1}, ..., x_{k+1}] - y[x_i, ..., x_{i+k-1}]}{x_{i+k} - x_i}$ mit k = 1, ..., j und i = k - j

für beliebige [?] $\sigma : 0, ..., n \rightarrow 0, ..., n$ gilt $y[\tilde{x_0}, ..., \tilde{x_n}] = y[x_0, ..., x_n]$

15 Nevillsche Darstellung

$$p_{jj}(x) = y_j \qquad j = 0, ..., n \qquad k = 1, ..., j \qquad i = k - j$$

$$p_{i,i+k}(x) = p_{i,i+k-1}(x) + (x - x_i)\frac{p_{i+1,i+k}(x) - p_{i,i+k-1}(x)}{x_{i+k} - x_i}$$

15.1 Schema

$$\begin{array}{cccccccc}
x & k=0 & & k=2 & \dots & k=n-1 & k=n \\
x_0 & y_0 & \longrightarrow & p_{0,1} & \dots & p_{0,n-1} & p_{0,n} \\
x_1 & y_1 & \longrightarrow & p_{1,2} & \dots & p_{1,n} & \\
\vdots & \vdots & \vdots & \iddots & & & \\
x_{n-1} & y_{n-1} & \longrightarrow & p_{n-1,n} & & & \\
x_n & y_n & & & & &
\end{array}$$

TODO: add the diagonal arrows

Hinzunahme von (x_{n+1}, y_{n+1}) ist problemlos

Auswertung von $p_{0,n}(x)$ in $\xi \neq x_i$ ohne vorherige Bestimmung der Koeffizienden der Newton-Darstellung ist einfach und Numerisch stabil möglich

15.2 Code

```
def divDiffs(xi, yi, x):
    n = len(xi)
    p = n * [0]
    for k in range(n):
        for i in range(n - k):
            if k == 0:
                p[i] = yi[i]
            else:
                p[i] = ((x - xi[i + k]) * p[i] + (xi[i] - x) * p[i + 1]) / (xi[i]
- xi[i + k])
    return p[0]
```

16 Hermite-Interpolation

16.1 Aufgabe

$$\begin{array}{llll} Gegeben: & x_i & i=0,...,m & paarweiseverschieden \\ & y_i^{(k)} & i=0,...,m & k=0,...,\mu_i(\mu_i \geq 0) \end{array}$$

Gesucht: $p \in P_n$, n = m + $\sum\limits_{i=0}^{m} \mu_i$: $p^{(k)}(x_i) = y_i^{(k)}$
x_i sind (μ_i + 1)-fache Stützstellen
$x_0 = -1$, $x_1 = 1$, $m = 1$, $y_0^{(0)} = 0$, $y_1^{(0)}$, $y_1^{([l?])} = 2$
$\Rightarrow \mu_0 = 0$, $\mu_1 = 1$
$\Rightarrow$ n = 1 + 0 + 1 = 2
$\Rightarrow$ p(x) = x^2

16.2 Existenz + Eindeutig

analog zur Lagrange-Interpolation

16.3 Fehler

$f \in C^{n+1}[a,b] : \forall x \in [a,b] \exists \xi_x \in (\overline{x_0,...,x_m,x})$, s.d.

$$f(x) - p(x) = f[x_0,...,x_0,...,x_m,...,x_m,x] \prod_{i=0}^{m} (x-x_i)^{\mu_i+1}$$

$= \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod\limits_{i=0}^{m} (x-x_i)^{\mu_i+1}$

17 Extrapolation zum Limes + Fehler

17.1 Richardson-Extrapolation

nicht direkt berechenbare Größe

$$a(0) = \lim_{k \to 0} a(k), \qquad k \in \mathbb{R}_+$$

berechne a(k_i) für gewisse k_i, i = 0, ..., n und [?] $p_n(0)$ des Interpolations Polynoms zu $(h_i, a(h_i))$ als Schätzung für a(0)

$a(0) := \lim\limits_{x \to 0^+} \frac{cos(x)-1}{sin(x)} \quad (= 0)$

$a(x) := \frac{(cos(x)-1)}{sin(x)}$

Interpolation a(x) an Stützstellen k_i nahe bei 0:

$k_0 = \frac{1}{8} \quad a(k_0) = -6,258151 * 10^{-2}$

$k_1 = \frac{1}{16} \quad a(k_1) = -3,126018 * 10^{-2}$

$k_2 = \frac{1}{32} \quad a(k_2) = -1,562627 * 10^{-2}$

17.2 Lagrange

$p_2(x) = a(k_0)\frac{(x-\frac{1}{16})(x-\frac{1}{32})}{(\frac{1}{8}-\frac{1}{16})(\frac{1}{8}-\frac{1}{32})} + a(k_1)\frac{(x-\frac{1}{8})(x-\frac{1}{32})}{(\frac{1}{16}-\frac{1}{8})(\frac{1}{16}-\frac{1}{32})} + a(k_2)\frac{(x-\frac{1}{8})(x-\frac{1}{16})}{(\frac{1}{32}-\frac{1}{8})(\frac{1}{32}-\frac{1}{16})}$

$\Rightarrow a(0) \sim p_2(0) = -1,02 * 10^{-5}$

17.3 Neville

$p_{i,i+k}(0) = p_{i,i+k-1}(0) + \frac{p_{i,i+k-1}(0) - p_{i+1,i+k}(0)}{\frac{x_{i+k}}{x_i}-1}$, k = 1, 2

i	x_i	$p_{i,i}(0) = a(k_i)$	$p_{i,i+1}(0)$	$p_{i,i+2}(0)$
0	$x_0 = \frac{1}{8}$	$-6,258151 * 10^{-2}$	$6,115 * 10^{-5}$	$-1,02 * 10^{-5}$
1	$x_1 = \frac{1}{16}$	$-3,126018 * 10^{-2}$	$7,64 * 10^{-6}$	
2	$x_2 = \frac{1}{32}$	$-1,562627 * 10^{-2}$		

17.4 Extrapolationsfehler

a(n) habe die Entwickling:

$a(h) = a_0 + \sum\limits_{j=1}^{n} a_j h^{jq} + a_{n+1}(h)h^{(n+1)q}$ mit q > 0, Koeffizienten a_j

und $a_{n+1}(h) = a_{n+1} + a(1[?????])$

$(h_k)_{k \in \mathbb{N}}$ erfülle:

$0 \leq \frac{h_{k+1}}{h_k} \leq p < 1$ ($\Rightarrow h_k$ positiv monoton fallend)

Dann gilt für $p_1^{(k)} \in P_n$ (in h^q) durch $(h_k^q, a(h_k))$, ..., $(h_{k+n}^q, a(h_{k+1}))$

$$a(0) - p_n^{(k)}(0) = O(h_k^{(n+1)q}) \qquad (k \to \infty)$$

18 Spline-Interpolation

18.1 Interpolationsnachteil

Starke Oszillation von Polynomen höheren Grades

18.2 Abhilfe

Spline-Interpolation, d.h. stückweise polynomielle Interpolation mit (n - 1)-mal stetig diff.baren Knoten

18.3 Lineare Spline

alle Abschnitt-Splines sind lineare Funktionen

18.4 Kubischer Spline

$s_n : [a, b] \to \mathbb{R}$ kubischer Spline bezüglich $a = x_0 < x_1 < ... < x_n = b$, wenn gillt

1. $s_n \in C^2[a, b]$
2. $S_n|_{I_i} \in P_3$, i = 1, ..., n

natürlicher Spline:

3. $s_n''(a) = s_n''(b) = 0$

18.5 Existenz

Der interpolierende kubische Spline existiert und ist eindeutig bestimmt durch zusammen Vorgabe von $s_n''(a), s_n''(b)$

für natürlichen Spline s_n durch $x_0, ..., x_n$, $y_0, ..., y_n$ gilt:

$$\int\limits_a^b |s'(x)|^2 dx \leq \int\limits_a^b |g''(x)|^2 dx$$

bezüglich g $\in C^2[a, b]$ mit g(x_i) = y_i, i = 1, ..., n

18.6 Approximationsfehler

$f \in C^4[a, b]$, $s_1''(a) = f''(a) \wedge s_n''(b) - f''(b)$:

$$\max_{x \in [a,b]} |f(x) - s : n(x)| \leq \frac{1}{2} h^4 \max_{x \in [a,b]} |f^{(4)}(x)$$

19 Gauß-Approximation

$< f, g >:= \int\limits_a^b f(t)\overline{g(t)}dt \qquad ||f|| = \sqrt{< f, f >}$

H Prähilbertraum, $\delta \subset H$ endlich Dimensional

$\exists f \in H$ eindeutig bestimmte "beste Approximation" $g \in S$

$||f - g|| = \min\limits_{\varphi \in S} ||f - \varphi||$

bes. einfache Lösung, wenn $\{\varphi_1, ..., \varphi_n\}$ eine ONB ist, d.h.

$(\varphi_i, \varphi_j) = \delta_{i,j} \Rightarrow \alpha_i =< f, \varphi_i >$ i = 1, ..., n

$\Rightarrow g = \sum\limits_{i=1}^{n} < f, \varphi_i > \varphi_i$ ist beste Approximation

20 Gram-Schmidt-Algorithmus

$w_1 := \frac{v_1}{||v_1||}$ $\tilde{w_k} := v_k - \sum_{i=1}^{k-1} \gamma < v_k, w_i > w_i$, $w_k := \frac{\tilde{w_k}}{||\tilde{w_k}||}$

20.1 Code

```
n = size(v, 1)
    k = size(v, 2)
    u = np.zeros(n, k)
    u[:, 1] = v[:, 1]/sqrt(v[:, 1] * v[:, 1])
    for i in range(2, k):
        u[:, i] = v[:, i]
        for j in range(1, i - 1):
            u[:, i] = u[:, i] - (u[:, i] * u[:, j]) / (u[:, j] * u[:, j]) * u[:, j]
        u[:, i] = u[:, i] / sqrt(u[:, i] * u[:, i])
```

21 Interpolatorische Quadraturformeln

$I(f) = \int\limits_a^b f(x)dx \approx I^{(n)}(f) = \sum\limits_{i=1}^n \alpha_i f(x_i)$

Stützstellen a $\leq a_0 < x_1 < ... < x_n \leq b$ und Gewichte $\alpha_i \in \mathbb{R}$

21.1 Interpolatorische Quadratur Formel

$I^{(n)}(f) = \int\limits_a^b p_n(x)dx = \sum\limits_{i=0}^n f(x_i) \int\limits_a^b L_i^{(n)}(x)dx$[1]

Lagrange:

$I(f) - I^{(n)}(f) = \int\limits_a^b f[x_0, ..., x_n, x] \prod\limits_{i=0}^n (x - x_i)dx$

$\int\limits_a^b f(x)dx \approx \int\limits_a^b p_0(x)dx = (b-a) * \sum\limits_{i=0}^n w_i f(x_i)$

$w_i = \frac{1}{(b-a)} \int\limits_a^b L_i(x)dx$

21.2 Ordnung

$I^{(n)} vonderOrdnungm \Leftrightarrow \forall p \in P_{m-1}$

$\int\limits_a^b p(x)dx = I^{(n)}(p)$ exakt

$\Rightarrow$ Interpolatorische Quadraturformel zu (n + 1)-Stützstellen sind mindestens von der Ordnung n + 1

$\Rightarrow$ höchstens Ordnung 2n + 2, mindestens n + 1

21.3 Newton-Cotes-Formel*

äquidistante Stützstellen

21.4 Abgeschlossene Formeln

$H = \frac{b-a}{n}, x_i = a + iH, a = x_0, b = x_n$

Trapezregel: $I^{(1)}(f) = \frac{b-a}{2}[f(a) + f(b)]$

Simpsonregel: $I^{(2)}(f) = \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)]$

$^3/_8 - Regel : I^{(3)}(f) = \frac{b-a}{8}[f(a) + 3f(a + H) + 3f(b - h) + f(b)]$

---

[1] α_i

21.5 Offene Formeln

$(H = \frac{b-a}{n+2}, x_i = a + (i+1)H, a < x_0, x_n < b)$

$I^{(0)}(f) = (b-a)f(\frac{a+b}{2})$ Mittelpunktregel

$I^{(1)}(f) = \frac{(b-a)}{2}(f(a+H) + f(b-H))$

$I^{(2)}(f) = \frac{(b-a)}{3}(2f(a+H) - f(\frac{a+b}{2}) + 2f(b-H))$

21.6 Code

21.7 Problem

negative Gewichte $\alpha_i \Rightarrow$ Auslöschungsgefah

Oszilationen des Lagrange Interpolanten (Runge-Phänomen)

$\Rightarrow I^{(n)}(f) \not\xrightarrow{n\to\infty} I(f)$

21.8 Abhilfe: Summierte Quadraturformeln *

$I - n^{(n)}(f) = \sum_{i=1}^{N-1} I^{(n)}_{[x_i, x_i+1]}(f)$ $h = \frac{b-a}{N}, x_i = a + iH$

21.8.1 Fehlerdarstellung

$I_{[x_i,x_{i+1}]}(f) - I^{(n)}_{[x_i,x_{i+1}]}(f) = w_n h^{n+2} f^{(m+1)}(\xi_i),$ $\xi_i \in [a, b]$

$m \geq n : I(f) - I^{(n)}_n(f) = w_n h^{(m+1)}(b-a) f^{(m+1)}(\xi)$

22 Gaußsche Quadraturformeln *

22.1 Gewichtetes Skalarprodukt

$$< f, g >_\omega = \int_a^b f(x)g(x)\omega(x)dx, \qquad \omega(x) \geq 0, x \in (a, b)$$

22.2 Gauß-Quadratur

$\exists!$ interpolierte Quadraturformel [?] (n + 1) paarweise verschiedene Stützstellen auf [-1, b] mit Ordnung 2n + 2. Stützstellen = Nullstellen

$\alpha_i = \int_{-1}^{1} \prod_{j=0, j\neq i} (\frac{x-\lambda_j}{\lambda_i-\lambda_j})^2 dx > 0$[2], i = 0, ..., n

$f \in C^{2n+2}([-1, 1]) Restglied:$

$$R^{(n)} = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_{-1}^{1} \prod_{j=0}^{n} (x - \lambda_j)^2 dx, \qquad \xi \in (-1, 1)$$

22.3 Wahl der Stützstellen

Nullstellen $\lambda_0, ..., \lambda_n \in (-1, 1)$ des (n + 1)-sten Legendre-Polynomes $L_{n+1} \in P_{n+1}$

22.4 Kongergenz der Gauß-Quadraturen

Sei $I^{(n)}(f)$ die (n + 1) punktige [?] Gauß-Formel zu $I(f) = \int_{-1}^{1} f(x)dx \forall f \in C[-1, 1] : I^{(n)}(f) \xrightarrow{n\to\infty} I(f)$

22.5 Code

[2] Positivität der Gewichte

23 Störungssattz...

23.1 Störungssatz

$A \in \mathbb{K}^{n \times n}$ regulär mit $||\delta A|| \leq \frac{1}{||A^{-1}||}$, dann gilt für die

23.2 gestörte Matrix

$\tilde{A} = A + \delta A$ ist regulär

Für den relativen Fehler der Lösung gilt mit Konditionszahl von A:

$cond(A) = ||A|| * ||A^{-1}||$

die Ungleichung:

$\frac{||\delta_x||}{||x||} \leq \frac{cond(A)}{1 - cond(A)\frac{||\delta A||}{||A||}} \left[\frac{||\delta b||}{||b||} + \frac{||\delta A||}{||A||} \right]$

24 Lösung von Dreieckssystemen+Aufwand*

24.1 Rückwärtseinsetzen

$$x_j = \begin{cases} \frac{b_n}{a_{nn}} & j = n \\ \frac{1}{a_{jj}}(b_j - \sum\limits_{k=j+1}^{n} a_{jk} x_k) & j = n-1, ..., 1 \end{cases}$$

24.2 Vorwärtseinsetzen

$$x_j = \begin{cases} \frac{b_n}{a_{nn}} & j = n \\ \frac{1}{a_{jj}}(b_j - \sum\limits_{k=j+1}^{n} a_{jk} x_k) & j = 1, ..., n-1 \end{cases}$$

24.3 Aufwend

$\sum\limits_{j=1}^{n} j = \frac{(n+1)n}{2} = \frac{n^2}{2} + O(n)$

25 Gaußsches Eliminationsverfahren...

25.1 Gauß-Elimination

Umformen von Ax = b auf Rx = c, R ist eine rechte obere Dreieck Matrix

25.2 Spaltenpivotisierung

$|a_{r_{k,k}}^{(k-1)}| = \max\limits_{j=k,\ldots,n} |a_{jk}^{(k-1)}|$

25.3 LR - Zerlegung

Ax = b

$$\begin{pmatrix} 2 & 1 & 7 \\ 8 & 8 & 33 \\ -4 & 10 & 4 \end{pmatrix} \text{x} = \begin{pmatrix} 15 \\ 73 \\ 12 \end{pmatrix} \quad \begin{matrix} \\ (-4)* \\ 2* \end{matrix} \begin{pmatrix} 8 & 8 & 33 \\ 2 & 1 & 7 \\ -4 & 10 & 4 \end{pmatrix} P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Eliminierung

$$\begin{pmatrix} 8 & 8 & 33 \\ 2 & 1 & 7 \\ -4 & 10 & 4 \end{pmatrix} L_1 = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 8 & 8 & 33 \\ 0 & 4 & 5 \\ 0 & 28 & 41 \end{pmatrix} P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\begin{matrix} \\ \\ (-7)* \end{matrix} \begin{pmatrix} 8 & 8 & 33 \\ 0 & 28 & 41 \\ 0 & 4 & 5 \end{pmatrix} L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{7} & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 8 & 8 & 33 \\ 0 & 28 & 41 \\ 0 & 0 & 6 \end{pmatrix} = R$$

$PA = LR \Rightarrow L_2L_1P_2P_1A = F \Rightarrow P_2P_1A = L_1^{-1}L_2^{-1}R, P_2P_1 = P$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$L_2L_1 = (\ddots), (L_2L_1)^{-1} = L = ([linkeunteredreiecksmatrix])$
$Ax = b \Rightarrow LRx = Pb, Pb = \tilde{b}$
$\Rightarrow Ly = \tilde{b}, Rx = y$
A regulär + diagonaldominant ⇒ A = LR kann ohne Pivot bezeichnet werden

25.4 Code

26 Symmetrisch positiv definite Systeme

26.1 Cholesky-Zerlegung

Jede symmetrische positiv definite Matrix A hat eine sogenannte Cholesky - Zerlegung:

$A = LDL^t = \tilde{L}\tilde{L}^t,\ \tilde{L} := LD^{\frac{1}{2}}$

$$\begin{pmatrix} \tilde{l}_{1,1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ \tilde{l}_{n,1} & \dots & \tilde{l}_{n,n} \end{pmatrix} \begin{pmatrix} \tilde{l}_{1,1} & \dots & \tilde{l}_{n,1} \\ \vdots & \ddots & \vdots \\ 0 & \dots & \tilde{l}_{n,n} \end{pmatrix} = \begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{pmatrix}$$

$i \geq j : a_{i,j} = \sum_{k=1}^{j} \tilde{l}_{i,k}\tilde{l}_{j,k} = \sum_{k=1}^{j-1} \tilde{l}_{i,k}\tilde{j,k} + \tilde{l}_{i,j}\tilde{l}_{j,j}$

i = 1, ..., n $\quad \tilde{l}_{i,i} = \sqrt{a_{i,i} - \sum_{k=1}^{i-1} \tilde{l}_{i,k}^2}$

j = i + 1, ..., n $\quad \tilde{l}_{i,j} = \frac{1}{\tilde{l}_{i,i}}(a_{i,j} - \sum_{k=1}^{i-1} \tilde{l}_{i,k}\tilde{l}_{j,k})$

Bandmatrizen: Nullen nicht speichern/berechnen Diagonal-Dominante: keine Pivotisierung notwendig symmetrisch positiv definite: keine Pivotisierung notwending

26.2 Aufwand

$N_{Cholesky}(n) = \frac{n^3}{6} + O(n^2)$ (billiger als A = LR)

26.3 Code

27 Least-Squares-Lösungen

$A \in \mathbb{R}^{m \times n} Ax = b$

keine Lösung, $b \notin im(A)$

unendlich viele Lösungen $\bar{x} + \delta x \Leftrightarrow A\bar{x} = b$, $\delta x \in ker(A) \neq \{0\}$

27.1 Least-Squares-Lösung

Es existiert immer eine "Lösung"$\bar{x} \in \mathbb{R}^n$ mit kleinsten Fehlerquadraten

27.2 Eindeutigkeit

R(A) = n $\Leftrightarrow \bar{x}$ eindeutig, jede weitere Lösung: $\bar{x} + y, y \in ker(A)$

Gerade: b = C + Dt $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$

$$\Rightarrow \quad \begin{aligned} C + x_1 D &= y_1 \\ C + x_2 D &= y_2 \\ C + x_3 D &= y_3 \end{aligned}$$

$$\Rightarrow A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix} x = \begin{bmatrix} C \\ D \end{bmatrix} b = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

1. C = $A^t A$ = [ ], b' = $A^t b$ = [ ]
2. Cholesky Zerlegung: $G^t G$ = C $\Rightarrow$ G = [ ]
3. $G^t y = b' \Rightarrow y = [], Gx = y \Rightarrow x = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$

$\Rightarrow Gerade : b = a_1 + a_2 t$

Alternativ: $Q^t A = R = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, Gx = y \Rightarrow x = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$

$R_1 x = \tilde{b}_1 \Rightarrow D = a_1, C = A_2$

28 QR-Zerlegung, ...

28.1 QR-Zerlegung

$A \in \mathbb{K}^{m\times n}, Rang(A) = n \leq m$

∃eindeutig bestimmte Matrix Q $\in \mathbb{K}^{m\times n}$ $\begin{matrix} Q^tQ = En(\text{für } \mathbb{K} = \mathbb{R}) \\ \overline{Q}^tQ = En(\text{für } \mathbb{K} = \mathbb{C}) \end{matrix}$

und eindeutig bestimmte obere Dreiecks Matrix $R \in \mathbb{K}^{n\times n} r_{i,i} < 0$ reell, s.d.

A = QR Q orthonormale Matrix (m = n unitär) $Q^t = Q^{-1}$

u := v + $\mathfrak{S}||v|| * e_n$

$\mathfrak{S}$

$$\begin{cases} -1 & v_1 < 0 \\ 1 & v_1 \geq 0 \end{cases} \text{ Householder Matrix } H = E_n - 2\frac{uu^t}{u^tu}$$

28.2 Least-Squares mit Vollrang

$A = Q_1R = (Q_1|Q_2) \begin{pmatrix} R \\ 0 \end{pmatrix}$, $Q = (Q_1|Q_2) \in \mathbb{R}^{m\times n}$

$Rx = Q^tb$ R = $\begin{pmatrix} R \\ 0 \end{pmatrix} \in \mathbb{R}^{m\times n}$

Rang(A) = n

$||Ax - b||_2^2 = ||Rx - Q_1^tb||_2^2 + ||Q_2^tb||_2^2$ minimal für $x = R^{-1}Q_1^tb$

28.3 Aufwand

Doppelter Aufwand für QR wie für LR

$N_{QR}(n) = \frac{2}{3}n^3 + O(n^2)$

28.4 Code

29 Householder-Transformation, ...

29.1 Householder Transformation

$v \in \mathbb{K}$, $||v||_2 = 1$

$H = E_n - 2\frac{vv^t}{v^tv}$

Spiegelung eines Vektors an einer Hyperebene durch Null im euklidischen Raum $H_{\vec{x}} = x - 2vv^tx$

$$v\overline{v}^t = \begin{pmatrix} v_1\overline{v}_1 & \dots & v_1\overline{v}_n \\ \vdots & \ddots & \vdots \\ v_n\overline{v}_1 & \dots & v_n\overline{v}_n \end{pmatrix}$$

TODO: skizze

29.2 Eigenschaften von Householder

Symmetrisch: $H = H^t$

Orthogonal: $H^tH = E_n$

Involutorisch: $h^2 = E_n$

EW: -1 einfach, 1 (n - 1)-fach

Mantisse-Vektor-Multiplikationen mit Householder sind schnell berechenbar

29.3 Householder-Verfahren

Gruppe numerischer Verfahren zur Bestimmung von Nullstellen einer skalaren, reellen Funktion

Zur Berechnung der QR-Zerlegung

$S_tA^{(t-1)} = A^{(t)}$, S_t ist Householder-Transformation

29.4 Ergebnis

QR-Zerlegung von A (aber <u>nicht</u>eindeutig!)

30 Intervallschachtelung/Bisektionsverfahren

30.1 Intervallschachtelung/Bisektionsverfahren

n = 1

Erzeugen einen konvergenten Folge von Intervallschachtelungen
$a_0, b_0 \in [a, b]$ mit $f(a_0)f(b_0) < 0$
Konvergenz: $a_k \leq a_{k+1} \leq b_{k+1} \leq b_k$
$|b_{k+1} - a_{k+1}| = \frac{1}{2}|b_k - a_k| = 2^{-k-1}|b_0 - a_0|$

30.2 Eigenschaften

sehr stabil, langsam, Erweiterung für $x \in \mathbb{R}^n \vee x \in \mathbb{C}$ <u>*nicht*</u> möglich

30.3 Code

```
for k = 0, 1, ... do:
        x[k] = 0.5(a[k] + b[k])
        if f(a[k])f(x[k]) ¡ 0:
                a[k + 1] = a[k]
                b[k + 1] = x[k]
        else:
                a[k + 1] = x[k]
                b[k + 1] = b[k]
        if abs(b[k + 1] - a[k + 1]) <TOL abs(a[k + 1]):
                Ende Lösung: 0.5(b[k + 1] + a[k + 1])
```

31 Konvergenz interativer Methoden

31.1 Konvergenzordnung

$\alpha > 1$

$|x_{k+1} - x_*| \leq C|x_k - x_*|^\alpha$ k = 0, 1, ...

31.2 Lineare Konvergenz

$\alpha = 1 \Rightarrow ClineareKontraktionsrate$

31.3 Superlineare Konvergenz

$|x_{k+1} - x_k| \leq C_k|x_k - x_*|$ mit $c_k \to 0$

31.4 Quadratische Konvergenz

$|x_{k+1} - x_*| \leq C|x_k - x_*|$ also $\alpha = 2$

Konvergenz der Ordnung α (quadratische Konvergenz, kubische, etc.) $\Rightarrow$ superlineare Konvergenz $\Rightarrow$ lineare Konvergenz

32 Newton Verfahren im $\mathbb{R}^n$

$x_{k+1} = x_k - j_f(x_k)^{-1} f(x_k)$

$j_f(x_k)^{-1} = \frac{1}{det(j_f)} * j_f^*$, $j_f^* = \begin{pmatrix} a_{2,2} & -a_{1,2} \\ -a_{2,1} & a_{1,1} \end{pmatrix}$

guter Startwert:	superlineare Konvergenz
schlechter Startwert[3]:	lineare Konvergenz

32.1 Code

33 Newton-Kantorisch

33.1 Voraussetzungen

$f : x \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ mit f' L-stetig, d.h.
1. $||j_f(x) - j_f(y)|| \leq L||x - y||$
2. $||j_f^{-1}(x)|| \leq \beta$
3. $x_0 \in D_f(x)$
4. $q := \frac{1}{2}\alpha\beta L$ mit $\alpha = ||j_f^{-1}(x_0)f(x_0)||$ auf der Nieveaumenge $D(x) = y \in D|||f(x)|| \leq ||x||$

$\Rightarrow$ Dann konvergiert (x_k) quadratisch gegen Nullstelle $z \in D$ von f

33.2 Fehlerabschätzung

$||x_k - x_*|| \leq \frac{\alpha}{1-q} q^{(2^k - 1)} \qquad , k \geq 1$

34 Sukzessive Approximation *

$x_{t+1} = x_t + C^{-1}f(x_t)$ C reguläre Matrix $\in \mathbb{R}^{n\times n}$

$x_t \xrightarrow[t\to\infty]{} z \Rightarrow z = z + C^{-1}f(z)$ bzw $f(z) = 0 \Rightarrow$ z Fixpunkt der Abbildung

$g(x) := x + C^{-1}f(x)$

34.1 Konvergenz

$g : G \to G$[?] Kontraktion, $\exists$ Fixpunkt $z \in Gvongx_t \to z$, q Kontraktionskonstante

$||x_t - z|| \leq \frac{q}{1-q}||x_t - x_{t-1} \leq \frac{q_t}{1-q}||x_1 - x_0||$

34.2 Code

35 Newton-Verfahren für affin - lineares f(x) = Ax - b

35.1 Problem

Direkte Methoden $\rightarrow$ großen Speicheraufwand für große n

36 Fixpunktiteration *, Konvergenzaussage

36.1 Problem

für sehr große n ist Gauß-Elimination zu speicherintensiv

$Ax = b \Leftrightarrow a_{j,j} * x_j + \sum_{k=1}^{n} a_{j,k} x_k = b_j$

36.2 Aufspaltung

A = D + L + R

D = $\begin{pmatrix} a_{1,1} & & 0 \\ & \ddots & \\ 0 & & a_{n,n} \end{pmatrix}$ L = $\begin{pmatrix} 0 & \dots & \dots & 0 \\ a_{2,1} & \ddots & \dots & \vdots \\ \vdots & \ddots & \dots & \vdots \\ a_{n,1} & \dots & a_{n,n-1} & 0 \end{pmatrix}$ R = $\begin{pmatrix} 0 & a_{1,2} & \dots & a_{1,n} \\ \vdots & \ddots & \dots & \vdots \\ \vdots & \ddots & \dots & a_{n-1,n} \\ 0 & \dots & \dots & 0 \end{pmatrix}$

$x_t = Bx_{t-1} + c$, B Iterationsmatrix

36.3 Konvergenz

$x_t \xrightarrow[t\to\infty]{} x \Rightarrow x = Bx + c$

$x_t \to x \Leftrightarrow spr(B) = max|\lambda||\lambda EW von B < 1$

Asymptotisches Konvergenzverhalten:

$\sup_{x_0 \in \mathbb{R}^n} \limsup_{t\to\infty} (\frac{||x_t - x||}{x_0 - x})^{\frac{1}{t}} = spr(B)$

36.4 Code

37 Jacobi *, Gauß-Seidel *, SOR *

37.1 Jacobi-Verfahren

$B = J = -D^{-1}(L + R)$ mit A = D + L + R

37.2 Gauß-Seidel

$B = H_1 = -(D + L)^{-1}R$

37.3 SOR

$H_w = (D + wL)^{-1}((1 - w)D - wR)$
(für w = 1: Gauß-Seidel)

37.4 Konvergenz

für positiv definiti symetrische Matrizen:
(I) starke/strikte Diagonaldominanz ⇒ J/G-S konvergiert
(II) diagonaldominanz + ireduzibel[4] ⇒ J/G-S konvergiert
(III) Beachte EW von: $J = -D^{-1}(L + R)$ J konvergiert für $spr(J) < 1$
$H = -(D + L)^{-1}R$ G-S konvergiert für $spr(H) < 1$

37.5 Fehler verringern

Wie viele Iterationsschritte t bis Fehler um 10^k verbessert ist?
$t \geq -\frac{k}{\log_{10} f}$ f = spr(B)

[4] Knotenmodell

38 Allgemeines Abstiegsverfahren

38.1 Voraussetzung

A symmetrisch positiv definiert
$\Rightarrow < Ax, y > = < x, Ay >, < Ax, x > > 0$

38.2 A-Skalarprodukt, A-Norm

$< x, y >_A = < x, Ay >$, $||x||_A = \sqrt{< Ax, y >}$
A hat nur reelle EW
$\lambda_{min} := \lambda_1 \leq ... \leq \lambda_n =: \lambda_{max} \Rightarrow spr(A) = \lambda_{max}, cond_2(A)\frac{\lambda_{max}}{\lambda_{min}}$

38.3 Abstiegsverfahren

Iteratives Verfahren, um lokales Minimum einer Funktion zu finden.
In jedem Schritt entlang einer bestimmten Richtung minimieren

38.4 Gradient

$g_t = Ax_t - b$, Abstiegsrichtung r_t

38.5 Schrittweite

$\alpha_t = -\frac{<g_t, r_t>}{<Ar_t, r_t>}$

38.6 Iteration

$x_{t+1} = x_t + \alpha_l r_t$
$\Rightarrow$ Minimierung von Q minimiert
Defektnorm $||A_y - b||_{A^{-1}}$ und Fehlernorm $||y - x||_A$

38.7 Code

39 Gradientenverfahren

Richtung des steilsten Abstiegs

r_t = -grad $Q(x_t) = -g_t$

39.1 Iteration

$x_0 \in \mathbb{R}^n, g_0 = Ax_0 - b, t \geq 0$

$\alpha_t = \frac{||g_t||^2}{<Ag_t, g_t>}$

$x_{t+1} = x_t - \alpha_t g_t$

$g_{t+1} = g_t - \alpha_t A g_t$

$< Ag_t, g_t >= 0 \Rightarrow g_t = 0 \Rightarrow Ax_t = b$

39.2 Konvergenz

A positiv definit symmetrisch $\Rightarrow$ Gradientenverfahren konvergiert $\forall x_0$ gegen Ax = b

39.3 Lemma von Kantorich

$4\frac{\lambda_{min}\lambda_{max}}{(\lambda_{min}+\lambda_{max})^2} \leq \frac{||<^4||}{<y,Ay><y,A^-1y>}$

39.4 Fehlerabschätzung

$||x_t - x||_A \leq (\frac{1-K^{-1}}{1+K^{-1}})^t ||x_0 - x||_a \qquad , t \in \mathbb{N}$

$\Rightarrow$ Lagransche Konvergenz für $cond_2(A) >> 1$

40 CG-Verfahren

Effiziente numerische Methode zur Lösung großen LGS mit symmetrisch positiv definitem A

Liefert nach spätestens m Schritten die exakte Lösung für $A \in \mathbb{R}^{m \times m}$ für exakte Arithmetik)

Fehler fällt monoton

Wähle Abstiegsrichtung d_t mit $<_i, d_j >= 0 \forall i \neq j$

$x_t = x_0 + \sum_{i=0}^{t-1} \alpha_i d_i \alpha x_0 + B_t$

$\Rightarrow x_{t+1} = x_t + \alpha_t d_t$

$\alpha_t = \frac{||g_t||^2}{<d_t, Ad_t>} \qquad \beta_t = \frac{||g_t+1||^2}{||g_t||^2}$

$x_{t+1} = x_t + \alpha_t d_t$

$g_{t+1} = g_t + \alpha_t A d_t$

$d_{t+1} = -g_{t+1} + \beta_t d_t$

$d_0 = g_0 = b - Ax_0$

40.1 Fehlerabschätzung

$||x_t - x||_A \leq 2(\frac{1-\frac{1}{\sqrt{K}}}{1+\frac{1}{\sqrt{K}}})^t ||x_0 - x||_A \qquad t \in \mathbb{N}$

$K = cond_2(A) = \frac{\lambda_{max}}{\lambda_{min}}$

Reduzierung des Anfangsfehlers: $maxt(\epsilon) \leq \frac{1}{2}\sqrt{K}\ln(\frac{2}{\epsilon}) + 1 Schritte$

$\Rightarrow$ CG konvergiert schneller als Gradientenverfahren

schneller, je näher $cond_2(A)$ bei 1 liegt, da $cond_2(A) = \frac{\lambda_{max}}{\lambda_{min}}$

aber, falls $\lambda_{max} >> \lambda_{min} \Rightarrow$ CG verfahren langsam

Lösung: Vorkonditionierung

41 Vorkonditionierung

Ax = b umformen in $\tilde{A}\tilde{x} = \tilde{b}$ mit besser konditioniertem $\tilde{A}$C symmetrisch positiv definit, K regulär: $C = KK^t$

$Ax = b \Leftrightarrow K^{-1}A(K^t)^{-1}$[5] $K^t x$[6] $= K^{-1}b$[7]

Dann CG anwenden auf $\tilde{A}\tilde{x} = \tilde{b}$

Wähle C so, dass $cond_2(\tilde{A}) << cond_2(A)$

Startwerte: $\tilde{x}_0 \in \mathbb{R}^n$ $\tilde{d}_0 = -\tilde{g}_0 = \tilde{b} - \tilde{A}\tilde{x}_0$

$g_0 = A_x 0 - b$

$C_{\rho_{t+1}} = g_{t+1}$ $C_{\rho_0} = g_0$ $d_0 = -\rho_0$

---

[5] $\tilde{A}$

[6] $\tilde{x}$

[7] $\tilde{b}$

42 Satz von Gerschgorin

Alle EW von $A\, in \mathbb{K}^{n\times n}$ liegen in den Vereinigung der sogenannten Gerschgorin-Kreise

$k_j = \zeta \in \mathbb{C} | |\zeta * a_{j,j}| \leq \sum\limits_{k=1,k\neq j}^{n} |a_j k|$

$U = \bigcup\limits_{i=1}^{m} k_{j,i}$, $V = \bigcup\limits_{j=1}^{n} k_j \backslash U$

Sind $U \cup V = \emptyset \Rightarrow$ m EW in U, n - m in V

43 Stabilitätsatz

$A \in \mathbb{K}^{n \times n}$ mit n linear unabhängig EV, $B \in \mathbb{K}^{n \times n}$
$\Rightarrow \forall \lambda_B$ EW von B $\exists \lambda_A$ EW von A:
$|\lambda_A - \lambda_B| \leq cond_2(W)||A - B||_2$
$W = (w_1, ..., w_n) \in \mathbb{K}^{n \times n}$ EV von A

44 Potenzmethode *, inverse Iteration *

Verfahren, um einen (nicht alle) EW zu finden

44.1 Potenzmethode

$\zeta \in \mathbb{C}^n \qquad ||\zeta_0|| = 0 \quad t \geq 1$
$\tilde{\zeta}_t = A_{\zeta_{t_1}} \quad \zeta_t = \frac{\tilde{\zeta}_t}{||\tilde{\zeta}||_t} \quad \lambda_t := \frac{\tilde{\zeta}_k^t}{\zeta_k^t} \quad \zeta_\zeta^t \neq 0$

44.2 Rayleigh-Quotient

A diagonalisierbar mit $|\lambda_n| > |\lambda_i|$
$\lambda_t = \frac{<\zeta_t, A\zeta_t>_2}{<\zeta_t, \zeta_t \quad _2}$

44.3 Inverse Iteration

A quadratisch $x_0 \in \mathbb{R}^n$, $\theta \in \mathbb{R}$ s.d.:
1. $q_k = \frac{x_{k-1}}{||x_{k-1}||}$
2. Löse $(A - \theta En)x_k = q_k \qquad (A - \theta En)$ regulär
EW ergibt sich mit Rayleigh-Quadratur: $\lambda_k = \frac{x_k^t A x_k}{x_k^t x_k}$

45 Hessenberg-Normalform

$\forall A \in \mathbb{R}^{n \times n} \exists$ Folge T_i von Householder-Matrizen, so dass:
i = 1, ..., n - 2
TAT^t Hessenberg-Matrix ist

45.1 Hessengerg-Matrix

$H \in \mathbb{C}^{n \times n}$ mit $h_{i,j} = 0 \forall i > j + 1$

$$\text{H} = \begin{pmatrix} h_{1,1} & \dots & \dots & \dots & h_{1,n} \\ h_{2,1} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & h_{n,n-1} & h_{n,n} \end{pmatrix}$$

alle Einträge unterhalb der ersten Nebendiagonalen sind 0 (obere H-Matrix) analog definiert man untere H-Matrix

sowohl obere als auch untere H-Matrix
$\Rightarrow$ Tridiagonalmatrix

46 QR-Verfahren

Berechnung aller EW einer quadratischen Matrix $A_1 = A \in \mathbb{K}^{n \times n}$

$A_t = Q_t R_t \to A_{t+1} = R_t * Q_t = Q_{t+1} * R_{t+1} \to \cdots \to A_{t+n}$ Diagonalmatrix

für die im QR-Verfahren erzeugten Matrix A_t gilt:

$\{\lim\limits_{t\to\infty} a_{j,j}^{(t)} | j = 1, ..., n\} = \{\lambda_1, ..., \lambda_n\}$