

Beispiel: Summation: Wir hatten für $y \in \mathbb{F}^n$

$$\tilde{S}(y) = (y_1 + y_2)(1 + \vartheta_{n-1}) + \sum_{k=3}^n y_k(1 + \vartheta_{n-k+1})$$

Definiere nun

$$\begin{aligned}\hat{y}_1 &:= y_1(1 + \vartheta_{n-1}) \\ \hat{y}_2 &:= y_2(1 + \vartheta_{n-1}) \\ \hat{y}_k &:= y_k(1 + \vartheta_{n-k+1}) \text{ für } k \geq 3\end{aligned}$$

$$\Rightarrow S(\hat{y}) = \tilde{S}(y)$$

Es gilt die Abschätzung

$$|\hat{y} - y|_1 \leq (|y_1| + |y_2|)|\vartheta_{n-1}| + \sum_{k=3}^n |y_k| \cdot |\vartheta_{n-k+1}| \leq \gamma_{n-1}|y|_1$$

Es folgt also

$$\varrho = \gamma_{n-1} = \varrho$$

3 Lineare Gleichungssysteme

3.1 Direkte Verfahren: Gauß-Elimination

3.1.1 Das Gaußsche Eliminationsverfahren

2×2 **Systeme:** Betrachte das Gleichungssystem

$$\begin{aligned}A_{11}u_1 + A_{12}u_2 &= b_1 \\ A_{21}u_1 + A_{22}u_2 &= b_2\end{aligned}$$

wobei die A_{ij} und die b_i gegeben (sodass $A_{11} \neq 0$) und die u_i gesucht sind.

$$\begin{aligned}A_{11}u_1 + A_{12}u_2 &= b_1 & | \cdot L_{21} &= \frac{A_{21}}{A_{11}} \\ A_{21}u_1 + A_{22}u_2 &= b_2 & | - L_{21} \cdot 1. \text{ Zeile}\end{aligned}$$

Äquivalentes System:

$$\begin{aligned}A_{11}u_1 + A_{12}u_2 &= b_1 \\ 0 \cdot u_1 + (A_{22} - L_{21}A_{12})u_2 &= b_2 - L_{21}b_1\end{aligned}$$

$$\tilde{A}_{22} := A_{22} - L_{21}A_{12}, \quad \tilde{b}_2 = b_2 - L_{21}b_1$$

2. Gleichung ist $\tilde{A}_{22}u_2 = \tilde{b}_2$

$$\tilde{A}_{22} \neq 0 \Rightarrow u_2 = \tilde{b}_2 / \tilde{A}_{22} \Rightarrow u_1 = (b_1 - A_{12} \cdot \frac{\tilde{b}_2}{\tilde{A}_{22}}) / A_{11}$$

$n \times n$ Systeme

$$\begin{array}{ccccccc} A_{11}u_1 & + & A_{12}u_2 & + & \dots & + & A_{1n}u_n & = & b_1 \\ A_{21}u_1 & + & A_{22}u_2 & + & \dots & + & A_{2n}u_n & = & b_2 & | - L_{21} \cdot 1. \text{ Zeile} \\ \vdots & & \vdots & & & & \vdots & & \vdots & \\ A_{n1}u_1 & + & A_{n2}u_2 & + & \dots & + & A_{nn}u_n & = & b_n & | - L_{n1} \cdot 1. \text{ Zeile} \end{array}$$

wobei $L_{j1} := \frac{A_{j1}}{A_{11}}$, falls $A_{11} \neq 0$

Mit $\tilde{A}_{22} := A_{22} - L_{21} \cdot A_{12}, \dots, \tilde{A}_{2n} := A_{2n} - L_{21} \cdot A_{1n}, \tilde{A}_{23} := \dots$, allgemein

$$\tilde{A}_{ij} := A_{ij} - L_{i1} \cdot A_{1j} \quad \text{und} \quad \tilde{b}_i := b_i - L_{i1} \cdot b_1$$

ergibt sich das äquivalente System:

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ 0 & \tilde{A}_{22} & \dots & \tilde{A}_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & \tilde{A}_{n2} & \dots & \tilde{A}_{nn} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_n \end{bmatrix}$$

$\tilde{A}_{22} \neq 0$ erlaubt den Algorithmus auf die $n-1 \times n-1$ Untermatrix anzuwenden. Nach $n-1$ Schritten erhalten wir, falls $\tilde{A}_{kk}^{(k)} \neq 0$ gilt

$$\begin{bmatrix} * & \dots & * \\ & \ddots & \vdots \\ 0 & & * \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} * \\ \vdots \\ * \end{bmatrix} \quad (\text{Rechts - obere Dreiecksmatrix})$$

Dieses lässt sich einfach auflösen:

n -te Gleichung: $\tilde{A}_{nn}^{(n)} u_n = \tilde{b}_n^{(n)}$

$$\Rightarrow u_n = \tilde{b}_n^{(n)} / \tilde{A}_{nn}^{(n)} \text{ falls } \tilde{A}_{nn}^{(n)} \neq 0$$

$$(n-1)\text{-te Gleichung: } \underbrace{\tilde{A}_{n-1,n-1}^{(n)}}_{\substack{= \tilde{A}_{n-1,n-1}^{(n-1)} \\ \neq 0 \text{ n. Vor}}} u_{n-1} + \tilde{A}_{n-1,n}^{(n)} \underbrace{u_n}_{\text{bek.}} = \tilde{b}_{n-1}^{(n)}$$

$$\Rightarrow u_{n-1} = \dots$$

usw...

3.1.2 Die LR-Zerlegung

Ziel: formalisiere diesen Algorithmus.

Wir wollen die Elimination in der Form $A \mapsto L \cdot A$ schreiben. Suche L . Sei L_1 die Matrix, die die erste Spalte von A (ab 2. Element) zu 0 mache:

$$(L_1 A)_{ij} = \sum_{k=1}^n L_{1;ik} A_{kj} \stackrel{!}{=} A_{ij} - L_{i1} \cdot A_{1j}$$

$k = i : L_{1;ii} = 1$

$k = 1 : L_{1;i1} = -L_{i1}$ und $L_{1;ik} = 0$ sonst.

L_1 hat also die Gestalt

$$L_1 = \begin{bmatrix} 1 & & & & \\ -L_{21} & \ddots & & & \\ \vdots & & \ddots & & \\ -L_{n1} & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

Genauso folgt:

$$L_2 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & -L_{32} & \ddots & & \\ & \vdots & & \ddots & \\ & -L_{n2} & & & 1 \end{bmatrix}$$

Nach Durchführung von $n - 1$ Schritten erhalten wir die rechts-obere Dreiecksmatrix

$$R = L_{n-1} \cdot \dots \cdot L_2 \cdot L_1 \cdot A$$

sowie

$$\tilde{b} = L_{n-1} \cdot \dots \cdot L_2 \cdot L_1 \cdot b$$

Schreibweise: Zu $a, b \in \mathbb{R}^n$ sei

$$a \otimes b := ab^\top \in \mathbb{R}^{n,n} \text{ (} a \text{ tensor } b \text{)}$$

Achtung: $a \otimes b \stackrel{\text{i.A.}}{\neq} b \otimes a$

Es gilt aber:

$$(a \otimes b) \otimes c = \left(\sum_j a_i \cdot b_j \cdot c_j \right)_i = a(b \cdot c)$$

D.h. $\dim(\text{Bild}(a \otimes b)) = 1$.

Wir definieren

$$\begin{aligned} \vec{i}_k &:= k\text{-ter euklidischer Einheitsvektor} \\ \vec{L}_k &:= [0, \dots, \underbrace{0}_k, L_{k+1,k}, \dots, L_{n,k}] \end{aligned}$$

Damit gilt: $L_k = Id_n - \vec{L}_k \otimes \vec{i}_k =$

$$\begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -L_{k+1,k} & \ddots & \\ & & \vdots & & \ddots \\ & -L_{n,k} & & & 1 \end{bmatrix}$$

Nun ist

$$\begin{aligned} (Id + \vec{L}_k \otimes \vec{i}_k)L_k &= (Id + \vec{L}_k \otimes \vec{i}_k)(Id - \vec{L}_k \otimes \vec{i}_k) \\ &= Id + \underbrace{\vec{L}_k \otimes \vec{i}_k - \vec{L}_k \otimes \vec{i}_k}_0 - \underbrace{\vec{L}_k \otimes \vec{i}_k \cdot \vec{L}_k \otimes \vec{i}_k}_{=\vec{L}_k(\vec{i}_k \cdot \vec{L}_k)\otimes \vec{i}_k} \\ &= Id \end{aligned}$$

$$\Rightarrow L_k^{-1} = Id + \vec{L}_k \otimes \vec{i}_k$$

Damit erhalten wir ($A = L_1^{-1} \cdot \dots \cdot L_{n-1}^{-1}R$):

$$\begin{aligned} L_1^{-1}L_2^{-1} &= (Id + \vec{L}_1 \otimes \vec{i}_1)(Id + \vec{L}_2 \otimes \vec{i}_2) \\ &= Id + \vec{L}_1 \otimes \vec{i}_1 + \vec{L}_2 \otimes \vec{i}_2 + \underbrace{\vec{L}_1 \otimes \vec{i}_1 \cdot \vec{L}_2 \otimes \vec{i}_2}_{=\vec{i}_1 \cdot \vec{L}_2 \cdot \vec{L}_1 \otimes \vec{i}_2} \\ &= Id + \vec{L}_1 \otimes \vec{i}_1 + \vec{L}_2 \otimes \vec{i}_2 \end{aligned}$$

Mit Induktion folgt:

$$L := L_1^{-1} \cdot \dots \cdot L_{n-1}^{-1} = Id + \sum_{k=1}^{n-1} \vec{L}_k \otimes \vec{i}_k = \begin{bmatrix} 1 & & & \\ L_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ L_{n1} & \dots & L_{nn-1} & 1 \end{bmatrix}$$

Satz 1. Ist die Gaußsche Elimination für ein $A \in \mathbb{R}^{n,n}$ durchführbar (d.h. gilt $\tilde{A}_{kk}^{(k)} \neq 0$ für $k = 1, \dots, n-1$), so besitzt A eine LR-Zerlegung, d.h.

$$A = LR$$

mit L links-untere Dreiecksmatrix mit Diagonale 1 und R eine rechts-obere Dreiecksmatrix.

Diese Zerlegung ist für invertierbare Matrizen eindeutig.

Beweis. Beh.: A invertierbar $\Leftrightarrow R$ invertierbar
(unter der Voraussetzung der Existenz von L und R)

$$\begin{aligned} A &= LR \\ \Leftrightarrow A^{-1} &= R^{-1}L^{-1} \end{aligned} \quad (1)$$

$$\Leftrightarrow R^{-1} = A^{-1}L \quad (2)$$

L^{-1} existiert immer. Weiter gilt:

$$(1) : A^{-1} \text{ ex.} \Rightarrow R^{-1} \text{ ex.}$$

$$(2) : R^{-1} \text{ ex.} \Rightarrow A^{-1} \text{ ex.}$$

Wegen der Behauptung folgt im Fall, dass A invertierbar ist die Invertierbarkeit von R .
Eindeutigkeit: Es sei $A = LR = \hat{L}\hat{R}$

$$\Rightarrow \underbrace{R\hat{R}^{-1}}_{\text{r.o. } \Delta\text{matrix}} = \underbrace{L^{-1}\hat{L}}_{\text{l.u. } \Delta\text{matrix}}$$

Beide Seiten sind also gleich der Identität, also folgt $R = \hat{R}$ und $L = \hat{L}$ □

3.1.3 Pivotisierung

Tritt der Fall $\tilde{A}_{kk}^{(k)} = 0$ auf, so wollen wir den Algorithmus modifizieren:

Skizze

Finde nun $l \in \{k+1, \dots, n\}$, so dass

$$|\tilde{A}_{lk}^{(k)}| \geq \max\{|\tilde{A}_{jk}^{(k)}| : j \in \{k+1, \dots, n\}\}$$

Ist $|\tilde{A}_{lk}^{(k)}| = 0$, so hat die Gesamtmatrix keinen maximalen Rang. Ist A invertierbar, so kann dieser Fall nicht auftreten.

Wir vertauschen nun die k -te mit der l -ten Zeile und setzen das Verfahren fort.

Die mathematische Beschreibung dieser Vertauschung ist die Anwendung einer Permutationsmatrix P .

z.B. hier:

$$P = \begin{bmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 0 & & & 1 & & & \\ & & & & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 1 & & & \\ & & & 1 & & & & 0 & & \\ & & & & & & & & 1 & \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{bmatrix} \begin{array}{l} \leftarrow k \\ \\ \leftarrow l \end{array} \quad \text{für } k \neq l$$

bzw. $P = Id$ für $k = l$.

Die Elimination liefert also $R = L_{n-1}P_{n-1} \cdots L_1P_1A$. Dabei suchen wir in jedem Schritt das maximale Element. Man kann zeigen, dass man dies in der Form LP schreiben kann, L links-untere Dreiecksmatrix, P Permutationsmatrix

Satz 2. Ist $A \in \mathbb{R}^{n,n}$ invertierbar, dann existiert eine Permutationsmatrix P , so dass PA eine LR -Zerlegung besitzt.

3.1.4 Rechenaufwand

$A \in \mathbb{R}^{n,n}$ vollbesetzt. (Gauß):

Multiplikationen ist $\sum_{k=1}^{n-1} (n-k)^2 = \frac{1}{3}n^3 + \mathcal{O}(n^2) = \mathcal{O}(n^3)$.

Speicher: Wird A nicht mehr benötigt, so kann man die Matrizen L und R auf A abspeichern.

Die explizite Verwendung der Zerlegung LR zur Lösung der gestaffelten Systeme empfiehlt sich, wenn man mehrere GLS der Form $Ax = b$ zu versch. b lösen muss. Einmal $\mathcal{O}(n^3)$ -Aufwand und dann nur noch $\mathcal{O}(n^2)$ für jedes folgende System.

3.1.5 Gauß-Elimination für Bandmatrizen

Schwach besetzte Matrizen und Bandmatrizen

$A \in \mathbb{R}^{n,n}$ heißt *schwachbesetzt* („sparse“), falls gilt:

$$\text{compl}(A) := \#\{[i, j] \in \{1, \dots, n\}^2 : A_{ij} \neq 0\} = \mathcal{O}(n)$$

Wir definieren die *Bandlänge* von A als das maximale $m \in \mathbb{N}$, für das gilt:

$$|i - j| > \left\lfloor \frac{m-1}{2} \right\rfloor \Rightarrow A_{ij} = 0$$

Diskretisierung von $-u''$ in 1d mit „natürlicher Anordnung“ führte auf Tridiag-matrix ($m = 3$). Diskretisierung von $-\Delta u$ in 2d in lexikographischer Anordnung ergab

$$\begin{bmatrix} \ddots & & & & \ddots \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ \ddots & & & & \ddots \end{bmatrix}$$

$\text{compl}(A) = \mathcal{O}(n)$, aber $m = \mathcal{O}(\sqrt{n})$

Die Elimination zerstört die Bandstruktur („fill in“), erhält aber die Bandlänge. Im 1d-Beispiel bleibt es bei einer Tridiagonalmatrix ($\text{compl}(L) = \text{compl}(R) = \mathcal{O}(n)$), aber in 2d folgt $\text{compl}(L) = \text{compl}(R) = \mathcal{O}(n^{\frac{3}{2}})$

Gauß-Elimination für Bandmatrizen

Hier nur Tridiagonalmatrizen:

$$A = \begin{bmatrix} a_1 & a_1^+ & & & \\ a_2^- & a_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & a_{n-1}^+ \\ & & & a_n^- & a_n \end{bmatrix}$$

mit $a_1 \neq 0$.

1. Schritt:

$$\tilde{A}^{(2)} = \begin{bmatrix} a_1 & a_1^+ & 0 & \cdots \\ 0 & \underbrace{a_2 - \frac{a_2^-}{a_1} a_1^+}_{\tilde{a}_2^{(2)}} & a_2^+ & \cdots \\ \vdots & & & \ddots \end{bmatrix}$$

Induktiv: $L_{i,i-1} = a_i^- / \tilde{a}_{i-1}^{(i-1)}$, $\tilde{a}_i^{(i)} = \tilde{a}_i^{(i-1)} - L_{i,i-1} \cdot a_{i-1}^+$

$$L = \begin{bmatrix} 1 & & & \\ L_{21} & \ddots & & \\ & \ddots & \ddots & \\ & & L_{n,n-1} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} * & a_1^+ & & \\ & \ddots & \ddots & \\ & & \ddots & a_{n-1}^+ \\ & & & * \end{bmatrix}$$

Anzahl Operationen: $\sim 4n$, falls $\tilde{a}_{i-1}^{(i-1)} \neq 0$. Allgemein ist der Aufwand für Bandmatrizen $\mathcal{O}(m^2 n)$ ohne Pivotisierung. Pivotisierung zerstört die Bandstruktur.

3.1.6 Block-Gauß-Elimination

$A \in \mathbb{R}^{n,n}$ mit $n = n_1 + n_2$.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad A_{11} \in \mathbb{R}^{n_1, n_1}; \quad A_{22} \in \mathbb{R}^{n_2, n_2}$$

$u = [u_1, u_2] \in \mathbb{R}^{n_1+n_2}$, $Au = [b_1, b_2] \in \mathbb{R}^{n_1+n_2}$

$$A_{11}u_1 + A_{12}u_2 = b_1$$

$$A_{21}u_1 + A_{22}u_2 = b_2$$

A_{11}^{-1} existiere. Multipliziere 1. Zeile mit $A_{21}A_{11}^{-1} =: L_{21}$ und subtrahiere dies von der 2. Zeile. Wir erhalten das äquivalente System:

$$\begin{aligned} A_{11}u_1 + A_{12}u_2 &= b_1 \\ 0 \cdot u_1 + \underbrace{(A_{22} - A_{21} \cdot A_{11}^{-1} \cdot A_{12})}_{=\tilde{A}_{22}^{(2)}} u_2 &= \underbrace{b_2 - A_{21} \cdot A_{11}^{-1} \cdot b_1}_{=\tilde{b}_2^{(2)}} \end{aligned}$$

Die Block- LR -Zerlegung:

$$A = LR = \begin{bmatrix} Id_{n_2} & 0 \\ L_{21} & Id_{n_2} \end{bmatrix} \cdot \begin{bmatrix} A_{11} & A_{12} \\ 0 & \tilde{A}_{22}^{(2)} \end{bmatrix}$$

3.1.7 Existenz der LR -Zerlegung ohne Pivotisierung

Satz 3. Sei $A \in \mathbb{R}^{n,n}$.

(i) A heißt diagonaldominant, falls

$$\sum_{\substack{j=1 \\ j \neq i}}^n |A_{ij}| < |A_{ii}| \quad i = 1, \dots, n$$

(ii) A heißt symmetrisch und positiv definit, falls

$$A_{ij} = A_{ji} \text{ und } v \cdot Av > 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$$

Für Matrizen A mit (i) oder (ii) ist die Elimination ohne Pivotisierung durchführbar.

Beweis. (i) $A_{11} \neq 0$ nach Voraussetzung

z.z.: $\tilde{A} = L_1 A$ ist wieder diagonaldominant.

Elimination: $\tilde{A}_{ij} = A_{ij} - L_{i1} A_{1j}$ für $i, j = 2, \dots, n$

Für $i = 2, \dots, n$ gilt

$$\begin{aligned} \sum_{\substack{j=2 \\ j \neq i}}^n |\tilde{A}_{ij}| &\leq \sum_{\substack{j=2 \\ j \neq i}}^n \{|A_{ij}| + |L_{i1}| \cdot |A_{1j}|\} \\ &= \sum_{\substack{j=1 \\ j \neq i}}^n |A_{ij}| - |A_{i1}| + |L_{i1}| \cdot \left(\sum_{j=2}^n |A_{1j}| - |A_{1i}| \right) \\ &< |A_{ii}| - |A_{i1}| + |L_{i1}| \cdot (|A_{11}| - |A_{1i}|) \\ &= |A_{ii}| - |A_{i1}| + \left| \frac{A_{i1}}{A_{11}} \right| \cdot (|A_{11}| - |A_{1i}|) \\ &= |A_{ii}| - |L_{i1}| \cdot |A_{1i}| \\ &\leq |A_{ii} - L_{i1} A_{1i}| = |\tilde{A}_{ii}| \end{aligned}$$

(ii) Reicht zu zeigen $\tilde{A} := L_1 A$ wieder symmetrisch und positiv definit.

$$A_{11} = \vec{i}_1 \cdot A \vec{i}_1 > 0.$$

\tilde{A} symmetrisch:

$$\tilde{A}_{ij} = A_{ij} - \frac{1}{A_{11}} \underbrace{A_{i1} \cdot A_{1j}}_{=A_{1j} \cdot A_{i1}=A_{j1} \cdot A_{1i}} = A_{ji} - \frac{1}{A_{11}} A_{j1} A_{1i} = \tilde{A}_{ji}$$

\tilde{A} positiv definit:

wir schreiben $A = \begin{bmatrix} A_{11} & a_1^\top \\ a_1 & A' \end{bmatrix}$, $v = \begin{bmatrix} v_1 \\ v' \end{bmatrix} \in \mathbb{R} \times \mathbb{R}^{n-1}$. Sei $v \neq 0$.

$$\begin{aligned}
0 < v \cdot Av &= \begin{bmatrix} v_1 \\ v' \end{bmatrix} \cdot \begin{bmatrix} A_{11} & a_1^\top \\ a_1 & A' \end{bmatrix} \begin{bmatrix} v_1 \\ v' \end{bmatrix} \\
&= \begin{bmatrix} v_1 \\ v' \end{bmatrix} \cdot \begin{bmatrix} A_{11}v_1 + a_1 \cdot v' \\ v_1 a_1 + A'v' \end{bmatrix} \\
&= A_{11}v_1^2 + 2v_1 a_1 \cdot v' + v' \cdot A'v' + \frac{1}{A_{11}}(a_1 \cdot v')^2 - \frac{1}{A_{11}}(a_1 \cdot v')^2 \\
&= A_{11}(v_1 - \frac{1}{A_{11}}a_1 \cdot v')^2 + v' \cdot A'v' - \frac{1}{A_{11}} \underbrace{(a_1 \cdot v')^2}_{\substack{v' \cdot a_1 a_1 \cdot v' \\ = v' \hat{a}_1 \otimes a_1 v'}} \\
&= A_{11}(v_1 - \frac{1}{A_{11}}a_1 \cdot v')^2 + v' \cdot (A' - \frac{1}{A_{11}}a_1 \otimes a_1)v'
\end{aligned}$$

zu $v' \in \mathbb{R}^{n-1}$ beliebig, wähle $v_1 = -\frac{1}{A_{11}}a_1 \cdot v'$ und erhalten

$$0 < v' \cdot \underbrace{(A' - \frac{1}{A_{11}}a_1 \otimes a_1)v'}_{ij\text{-Komponente ist } A_{ij} - \frac{1}{A_{11}}A_{1i} \cdot A_{1j} = \tilde{A}_{ij}}$$

\Rightarrow Für alle $v' \in \mathbb{R}^{n-1} \setminus \{0\}$ gilt $0 < v' \cdot [\tilde{A}_{ij}]_{\substack{i=2,\dots,n \\ j=2,\dots,n}} v' \Rightarrow \text{Beh.}$

□

3.1.8 Numerische Stabilität

Satz 4. Zu $A \in \mathbb{R}^{n,n}$ sei $\tilde{L}\tilde{R}$ die numerisch berechnete LR-Zerlegung. Dann gilt:

$$\frac{\|\tilde{L}\tilde{R} - A\|_\infty}{\|A\|_\infty} \leq 2n^3 f(A)\varepsilon + o(\varepsilon)$$

mit $f(A) = \frac{\max\{|\tilde{a}_{ij}^{(k)}| : k, i, j\}}{\max\{|a_{ij}| : i, j\}}$.

D.h. Stabilität liegt vor, falls $f(A) \in \mathcal{O}(1)$ ist.

z.B. für diagonaldominante Matrizen $f(A) \leq 2$, aber Beispiele mit $f(A) = 2^n$ sind explizit bekannt.

3.1.9 Bemerkungen

- Man kann mit der Gauß-Elimination auch die Inverse einer Matrix berechnen (Gauß-Jordan-Algorithmus)
- Mit der Gauß-Elimination kann man $\det(A)$ berechnen:

$$\det(A) = \det(LR) = \det(L) \cdot \det(R) = \det(R) = \prod_{k=1}^n R_{kk}$$

3.2 Cholesky-Zerlegung

Satz 5. Sei A spd (symmetrisch, positiv definit) aus $\mathbb{R}^{n,n}$.

Dann ex. eine untere Dreiecksmatrix L mit positiven Diagonaleinträgen, so dass

$$A = L \cdot L^T \quad (\text{Cholesky - Zerlegung})$$

Beweis. Mit Induktion über n :

$$n = 1 : 0 < A_{11} = \sqrt{A_{11}} \cdot \sqrt{A_{11}}$$

$$n - 1 \hookrightarrow n : \text{Sei } A = \begin{bmatrix} A' & a_1 \\ a_1^\top & A_{nn} \end{bmatrix} \text{ mit } A' \in \mathbb{R}^{n-1, n-1}, a_1 \in \mathbb{R}^{n-1}$$

Mit $v = [v', 0]$ sieht man: A' spd.

A' hat nach I.V. eine Zerlegung $A' = L'(L')^\top$

Ansatz:

$$\begin{aligned} A = \begin{bmatrix} A' & a_1 \\ a_1^\top & A_{nn} \end{bmatrix} &\stackrel{!}{=} \underbrace{\begin{bmatrix} L' & 0 \\ r^\top & \alpha \end{bmatrix}}_L \cdot \underbrace{\begin{bmatrix} (L')^\top & r \\ 0 & \alpha \end{bmatrix}}_{L'^\top} \\ &= \begin{bmatrix} L'(L')^\top & L'r \\ (L'r)^\top & |r|^2 + \alpha^2 \end{bmatrix} \end{aligned}$$

Ziel: Gib r, α an.

$$a_1 = L'r \Rightarrow r = (L')^{-1} \cdot a_1$$

Aber:

$$\begin{aligned} |r|^2 + \alpha^2 &\stackrel{!}{=} A_{nn} > 0 \\ \alpha^2 &= A_{nn} - |r|^2 \stackrel{?}{>} 0 \end{aligned}$$

$$\text{Falls ja: } \alpha := \sqrt{A_{nn} - |r|^2}$$

$$\text{Wähle } \tilde{r} := ((L')^\top)^{-1} r \in \mathbb{R}^{n-1} \text{ und nutze } 0 < \begin{bmatrix} \tilde{r} \\ -1 \end{bmatrix} \cdot A \begin{bmatrix} \tilde{r} \\ -1 \end{bmatrix}$$

□

Algorithmus:

$$\text{Ansatz: } A_{ik} = \sum_{j=1}^k L_{ij} \cdot L_{kj}, \quad i \geq k$$

Spaltenweise auflösen:

$$\begin{aligned} k = 1, \quad i = 1, \dots, n \quad & A_{i1} = L_{i1} \cdot L_{11} \\ & i = 1 \quad A_{11} = L_{11}^2 \Rightarrow L_{11} = A_{11}^{1/2} \\ & i > 1 : \quad L_{i1} = A_{i1}/L_{11} = A_{i1}/\sqrt{A_{11}} \\ k = 2, \quad i = 2, \dots, n \quad & A_{i2} = L_{i1}L_{21} + L_{i2}L_{22} \\ & i = 2 \quad A_{22} = L_{21}^2 + L_{22}^2 \Rightarrow L_{22} = \sqrt{A_{22} - L_{21}^2} \\ & i > 2 : \quad L_{i2} = \dots \end{aligned}$$

Satz 6. Sei $A \in \mathbb{R}^{n,n}$ spd. Algorithmus liefere $A = \tilde{L}(\tilde{L})^\top$.

Dann gilt

$$\frac{\|\tilde{L}(\tilde{L})^\top - A\|_2}{\|A\|_2} \leq 8n(n+1)\varepsilon + o(\varepsilon)$$

Der Algorithmus ist also rückwärtsstabil.

3.3 Iterative Verfahren

3.3.1 Basisiteration

Ziel: Schreibe $Au = b$ als Fixpunktiteration zur Lösung $u = Tu + d$ mit geeignetem $T \in \mathbb{R}^{n,n}$, $d \in \mathbb{R}^n$

Sei $B \in \mathbb{R}^{n,n}$ invertierbar. Dann gilt:

$$Au = b \Leftrightarrow B Au = B b \Leftrightarrow u = u - B Au - B b \Leftrightarrow u = \underbrace{(Id - BA)}_{=:T} u + \underbrace{Bb}_{=:d}$$

B nennt man *Vorkonditionierung* (dient der Beschleunigung des folgenden Algorithmus)

Basisiteration:

$$u_{i+1} = (Id - BA)u_i + Bb \quad (\text{Fixpunktiteration})$$

Falls $u_i \rightarrow u$ ($i \rightarrow \infty$), so löst u die Gleichung $Au = b$

Einschub zu 3.1.: Zerlegungen: Idee um an geeignetes B zu kommen: $A = M - N$ („Hauptteil“ M (invertierbar), „Nebenteil“ N)

$$Au = b \Leftrightarrow Mu - Nu = b \Leftrightarrow Mu = Nu + b \Leftrightarrow u = \underbrace{M^{-1}N}_{=:T} u + \underbrace{M^{-1}b}_{=:d}$$

bzw.. $B = M^{-1}$

In 3.3.2.: $M = D$, $N = D(L + R)$

Bemerkung

Optimal wäre $B = A^{-1}$, aber B sollte nur so komplex wie A sein. Widerspricht sich.

3.3.2 Konvergenz linearer Iterationen

Satz 7. Zu $u_0 \in \mathbb{R}^n$ definieren wir $u_{i+1} := Tu_i + d$. Ist u Lösung zu $u = Tu + d$, dann gilt:

- 1.) Gilt in einer Operatornorm $\|T\| < 1$, so konv die Folge $\{u_i\}_{i \geq 0}$ gegen u und es gilt:

$$\begin{aligned} |u_i - u| &\leq \|T\|^i |u_0 - u| \quad (\text{A priori - Abschätzung}) \\ |u_i - u| &\leq \frac{\|T\|}{1 - \|T\|} |u_i - u_{i-1}| \quad (\text{A posteriori}) \end{aligned}$$

- 2.) Es gilt $u_i \rightarrow u$ ($i \rightarrow \infty$) für alle $u_0 \in \mathbb{C}^n \Leftrightarrow \varrho(T) < 1$

Beweis. 1.) *Banachscher Fixpunktsatz:*

$$\begin{aligned} (u_i - u &= Tu_{i-1} + d - Tu - d = T(u_{i-1} - u) \\ |u_i - u| &\leq \|T\| \cdot |u_{i-1} - u| \leq \dots \leq \|T\|^i |u_0 - u|) \\ q &:= \|T\| < 1 \end{aligned}$$

$$\begin{aligned} |u - u_i| &\leq |u - u_{i+1}| + |u_{i+1} - u_i| \\ &\leq |u - u_{i+1}| + q|u_i - u_{i-1}| \\ &\leq |u - u_{i+2}| + q^2|u_i - u_{i-1}| + q|u_i - u_{i-1}| \leq \dots \leq \\ &\leq q \cdot \sum_{j=0}^{\infty} q^j |u_i - u_{i-1}| \\ &= \frac{q}{1-q} \cdot |u_i - u_{i-1}| \end{aligned}$$

2.) „ \Rightarrow “: Definiere $e_i := u - u_i$, dann gilt:

$$e_{i+1} = Te_i$$

Mit Induktion folgt:

$$e_i = T^i e_0$$

Beachte: Es gilt

$$u_i \rightarrow 0 \ (i \rightarrow \infty) \Leftrightarrow e_i \rightarrow 0 \ (i \rightarrow \infty)$$

Es sei $\lambda \in \mathbb{C}$ ein Eigenwert von T und z ein zugehöriger normierter Eigenvektor (also $|z| = 1$)

$$Tz = \lambda z$$

Wähle $u_0 := u - z$

$$\Rightarrow e_i = T^i e_0 = T^i z = \lambda^i z$$

Nach Vor. gilt $|\lambda|^i = |\lambda|^i |z| = |e_i| \rightarrow 0 \ (i \rightarrow \infty)$, also gilt

$$|\lambda| < 1$$

Da λ beliebiger Eigenwert war folgt $\varrho(T) < 1$.

„ \Leftarrow “: Da $\varrho(T) < 1$, gibt es ein $\varepsilon > 0$ mit $\varrho(T) < 1 - \varepsilon$. Mit ÜA gilt: $\exists \|\cdot\|_\varepsilon$, induzierte Matrixnorm, sodass gilt

$$\|T\|_\varepsilon \leq \underbrace{\varrho(T) + \varepsilon}_{<1}$$

□

Eigenwerte von Tridiagonalmatrizen

Es seien $a, b, c \in \mathbb{R}$ mit $ac > 0$ und $A := \text{tridiag}_N[a, b, c]$ eine reelle Tridiagonalmatrix. Dann sind die Eigenvektoren von A gegeben durch

$$s^k = \left[\left(\frac{a}{c} \right)^{\frac{j-1}{2}} \sin \left(\frac{k\pi j}{N+1} \right) \right]_{j=1, \dots, N}, \quad k = 1, \dots, N$$

Die zugehörigen Eigenwerte sind

$$\lambda_k = b + 2 \operatorname{sgn}(a) \sqrt{ac} \cdot \cos \left(\frac{k\pi}{N+1} \right), \quad k = 1, \dots, N$$

3.3.3 Die „klassischen Iterationsverfahren“

Richardson-Verfahren $B = \omega \cdot Id$ für ein geeignetes $\omega \in \mathbb{R}$. D.h.

$$u_{i+1} = u_i - \omega(Au_i - b) \quad (i > 0)$$

Die Iterationsmatrix ist

$$T_R = Id - \omega A$$

Jakobi-Verfahren (Gesamtschrittverfahren) Zerlegung von $A = D(Id - L - R)$ mit $D := \text{diag}(A)$, $-DL$ der links-untere, $-DR$ der rechts-obere Anteil von A (mit Diagonale 0)

$$\underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_A = \underbrace{\begin{bmatrix} * & & \\ & * & \\ & & * \end{bmatrix}}_D + \underbrace{\begin{bmatrix} * & & \\ * & * & \\ * & * & \end{bmatrix}}_{-DL} + \underbrace{\begin{bmatrix} & * & * \\ & & * \\ & & \end{bmatrix}}_{-DR}$$

Iteration:

$$u_{i+1} = u_i - \underbrace{D^{-1}}_{=B}(Au_i - b)$$

Die Iterationsmatrix lautet also:

$$T_J = Id - D^{-1}A$$

In Komponenten:

$$\begin{aligned} u_{i+1,l} &= u_{i,l} - \frac{1}{A_{ll}} \left(\sum_{m=1}^n A_{lm} u_{i,m} - b_l \right) \\ &= -\frac{1}{A_{ll}} \left(\sum_{\substack{m=1 \\ m \neq l}}^n A_{lm} u_{i,m} - b_l \right) \end{aligned}$$

Oft verwendet man noch einen „Dämpfungsfaktor“ $\omega \in \mathbb{R}$

$$u_{i+1} = u_i - \omega D^{-1}(Au_i - b) \quad \text{„Gedämpftes Jakobi-Verfahren“}$$

Hier ist die Iterationsmatrix also

$$T_{J,\omega} = (Id - \omega D^{-1}A)$$

Bemerkung $u = [u_{i,l}]_l, V \subset \mathbb{R}^n$
 $V \leftarrow AU, V \leftarrow V - b, V \leftarrow D^{-1}V$
 $U \leftarrow U - \omega V$

Gauß-Seidel-Verfahren (Einzelschrittverfahren) und das SOR-Verfahren

Einzelschrittverfahren: Idee: nutze schon die neu berechneten Komponenten, um Au zu berechnen.

$$u_{i+1,l} = -\frac{1}{A_{ll}} \left(\sum_{m=1}^{l-1} A_{l,m} u_{i+1,m} + \sum_{m=l+1}^n A_{l,m} u_{i,m} - b_l \right) \quad (*)$$

In Matrix-Schreibweise:

$$\begin{aligned} Du_{i+1} - DLu_{i+1} - DRu_i &= b \\ D(Id - L)u_{i+1} &= b + DRu_i \\ \Rightarrow u_{i+1} &= (Id - L)^{-1}D^{-1}(b + DRu_i) \end{aligned}$$

Die Iterationsmatrix ist also:

$$T_{GS} = (Id - L)^{-1} \cdot R$$

(Formel! Die Implementierung ist die Formel (*))

Hier ist $M = D(Id - L)$ oder $B = (Id - L)^{-1}D^{-1}$

SOR-Verfahren (successive overrelaxation) Mit Dämpfungsparameter $\omega \in \mathbb{R}$

$$u_{i+1} = u_i - \omega D^{-1}(Du_i - DLu_{i+1} - DRu_i - b)$$

bzw.

$$u_{i+1} = u_i - \omega (Id - \omega L)^{-1} D^{-1} (Au_i - b)$$

Die Iterationsmatrix ist

$$T_{\omega}^{\text{SOR}^+} = Id - \omega (Id - \omega L)^{-1} D^{-1} A$$

Implementierung:

Mit Hilfe eines Unterprogramms $(U, l) \rightarrow (AU - b)_l$ spart man sich den Vektor V gegenüber 3.3.2. Das Verfahren ist aber abhängig vom gewählten Durchlauf

SSOR-Verfahren (Symmetrisches SOR) Erst SOR mit Durchlauf $1, \dots, n$ dann $n, \dots, 1$.
Außerdem erhält man damit eine symmetrische Iteration.

Die Iterationsmatrix T_ω^{SSOR} setzt sich zusammen aus

$$\begin{aligned} T_\omega^{\text{SOR}^+} &:= Id - \omega(Id - \omega L)^{-1} D^{-1} A \quad \text{und} \\ T_\omega^{\text{SOR}^-} &:= Id - \omega(Id - \omega R)^{-1} D^{-1} A \end{aligned}$$

zu deren Produkt:

$$T_\omega^{\text{SSOR}} = T_\omega^{\text{SOR}^-} \cdot T_\omega^{\text{SOR}^+}$$

3.3.4 Konvergenz des Jakobi- und Gauß-Seidel-Verfahrens

Satz 8. Sei $A \in \mathbb{R}^{n,n}$ mit $A_{ii} \neq 0$ für alle i .

(i) (Starkes Zeilensummenkriterium:) Gilt

$$\|L + R\|_\infty < 1 \quad (\text{Zeilensummennorm})$$

dann konvergieren Jakobi und GS und es gilt

$$\varrho(T_{GS}) \leq \varrho(T_J) < 1$$

(ii) (Schwachere Zeilensummenkriterium:) Es gelte

$$\sum_{\substack{j=1 \\ j \neq i}}^n \left| \frac{A_{ij}}{A_{ii}} \right| \leq 1, \quad i = 1, \dots, n$$

aber „<“ gelte für wenigstens einen Index. Weiter sei A unzerlegbar (irreduzibel), d.h. gibt es Mengen $M_1, M_2 \subset I = \{1, \dots, n\}$ mit $M_1 \cup M_2 = I$, aber $M_1 \cap M_2 = \emptyset$ und gilt $A_{ij} = 0$ für alle $(i, j) \in M_1 \times M_2$, so folgt $M_1 = \emptyset$ oder $M_2 = \emptyset$.

(Keine Permutation P führt auf $PA = \begin{bmatrix} * & 0 \\ * & * \end{bmatrix}$)

Dann konvergieren Jakobi- und GS-Verfahren.

Beweis. (ii) z.z. Jakobiverfahren: $T \equiv T_J = Id - D^{-1}A \Rightarrow \varrho(T) < 1$.

Sei $\lambda \in \mathbb{C}$, $v \in \mathbb{C}^n$ mit $|v|_\infty = 1$ und $Tv = \lambda v$.

Annahme: $|\lambda| \geq 1$

Dann gilt für jedes $i \in I = \{1, \dots, n\}$

$$|v_i| \leq |\lambda v_i| = |(Tv)_i| \leq \sum_{\substack{j=1 \\ j \neq i}} \left| \frac{A_{ij}}{A_{ii}} \right| \underbrace{|v_j|}_{\leq 1} \leq \sum_{\substack{j=1 \\ j \neq i}} \left| \frac{A_{ij}}{A_{ii}} \right| \leq 1$$

Sei i_0 ein Index mit „<“.

Dann ist für ein $j \in I$: $A_{i_0 j} \neq 0$, denn sonst wäre A reduzibel mit $M_1 = \{i_0\}$, $M_2 = I \setminus \{i_0\}$. Für $i = i_0$ folgt dann also $|v_{i_0}| < 1$.

Nun sei $M_1 = \{i \in I : |v_i| = 1\}$ und $M_2 = I \setminus M_1$.

Dann ist $M_1 \neq \emptyset$ nach Vor. $|v|_\infty = 1$. Sei $i \in M_1$. Weil nicht $A_{ij} = 0$ für alle $j \in M_2$ gelten kann, kann man $|v_i| < 1$ wie oben zeigen. Wid.

Gauß-Seidel-Verfahren: Wähle λ, v wie oben für $T = T_{GS}$
Für T gilt:

$$(Tv)_i = \sum_{j=1}^{i-1} \frac{A_{ij}}{A_{ii}} (Tv)_j + \sum_{j=i+1}^n \frac{A_{ij}}{A_{ii}} \cdot v_j$$

Mit Induktion folgt: $|(Tv)_j| \leq 1$ für $j \in I$.

Damit $|(Tv)_j| = |\lambda v_j| = |\lambda| \cdot |v_j| \leq |v_j| \Rightarrow |\lambda| \leq 1$

Somit

$$|(Tv)_i| \leq \sum_{j=1}^{i-1} \left| \frac{A_{ij}}{A_{ii}} \underbrace{(Tv)_j}_{\leq 1} \right| + \sum_{j=i+1}^n \left| \frac{A_{ij}}{A_{ii}} \right| \underbrace{|v_j|}_{\leq 1} \leq \sum_{\substack{j=1 \\ j \neq i}}^n \left| \frac{A_{ij}}{A_{ii}} \right|$$

Dann geht der Beweis wie oben. □

3.3.5 Konvergenzsatz des SOR-Verfahrens

Satz 9. Es sei $A \in \mathbb{R}^{n,n}$ mit $A_{ii} \neq 0$, $i = 1, \dots, n$ mit $T_{GS,\omega}$ sei die Iterationsmatrix des SOR-Verfahrens. Dann gilt:

1.)

$$\varrho(T_{GS,\omega}) \geq |\omega - 1|$$

D.h. SOR konvergiert höchstens für $\omega \in (0, 2)$

2.) Ist A spd, so gilt:

$$\varrho(T_{GS,\omega}) < 1 \quad \text{für } \omega \in (0, 2)$$

Beweis. 1.) $T = T_{GS,\omega}$ hat die Form

$$\begin{aligned} T &= Id - \omega(Id - \omega L)^{-1} D^{-1} A \\ &= (Id - \omega L)^{-1} (Id - \omega L - \omega D^{-1} A) \\ &= (Id - \omega L)^{-1} ((1 - \omega)Id + \omega R) \end{aligned}$$

$$\det(T) = \det((Id - \omega L)^{-1}) \cdot \det((1 - \omega)Id + \omega R) = (1 - \omega)^n.$$

Wegen $\det(T) = \prod_{i=1}^n \lambda_i$ folgt: es ex ein $i_0 \in I$ mit $\varrho(T) \geq |\lambda_{i_0}| \geq |\omega - 1|$.

2.) Aufwändig. □

Bemerkung Konvergenzkriterium ist unabhängig von der Nummerierung.

3.3.6 Konvergenz des SSOR

Satz 10. Sei $A \in \mathbb{R}^{n,n}$ spd. Zu $\omega \in \mathbb{R}$ sei

$T_{GS,\omega}^+$: SOR-Operator mit Durchlauf $i = 1, \dots, n$

$T_{GS,\omega}^-$: SOR-Operator mit Durchlauf $i = n, \dots, 1$

Dann ist die Iterationsmatrix des SSOR-Verfahrens durch $\delta_\omega = T_{GS,\omega}^- \cdot T_{GS,\omega}^+$ gegeben.

Es gilt:

$$\varrho(\delta_\omega) \geq |\omega - 1|^2 \text{ und } \varrho(\delta_\omega) < 1 \text{ f\"ur } \omega \in (0, 2)$$

Beweis. Korollar zum letzten Theorem □

3.3.7 Beispiele

$A := \text{tridiag}(-1, 2, -1)$.

Mit $h := \frac{1}{n+1}$ erhalten wir die Eigenwerte $\lambda_k = 2(1 - \cos(k\pi h))$. Wir suchen $\varrho(T)$ für verschiedene Verfahren:

1.) Jakobi-Verfahren:

$$T_J = Id - D^{-1}A = Id - \frac{1}{2}A$$

$$\text{Eigenwerte: } \lambda_{J,k} = 1 - \frac{1}{2}\lambda_k = \cos(k\pi h)$$

Bild

$$\Rightarrow \varrho(T_J) = \cos(\pi h) = 1 - \frac{1}{2}(\pi h)^2 + \mathcal{O}(h^4) = 1 - \frac{1}{2}\frac{\pi^2}{n^2} + \mathcal{O}(n^{-3})$$

2.) Gauß-Seidel-Verfahren:

Für die Komponenten von $T_{GS} \cdot u$ gilt:

$$(T_{GS}u)_l = \frac{1}{2}((T_{GS}u)_{l-1} + u_{l+1})$$

Ist $T_{GS}u = \lambda_{GS}u$, so folgt:

$$\begin{aligned} \lambda_{GS}u_l &= \frac{1}{2}(\lambda_{GS}u_{l-1} + u_{l+1}) \quad | \cdot \lambda_{GS}^{-\frac{l+1}{2}} = \sqrt{\lambda_{GS}}^{-(l+1)} \\ \Leftrightarrow \sqrt{\lambda_{GS}}^{-l+1}u_l &= \frac{1}{2}(\sqrt{\lambda_{GS}}^{-l+1}u_{l-1} + \sqrt{\lambda_{GS}}^{-l-1}u_{l+1}) \\ \Leftrightarrow \sqrt{\lambda_{GS}}v_l &= \frac{1}{2}(v_{l-1} + v_{l+1}) = (T_Jv)_l \end{aligned}$$

Ist λ_J Eigenwert von T_J , so ist λ_J^2 Eigenwert von T_{GS}

$$\varrho(T_{GS}) = \cos(\pi h)^2 = (1 - \pi h + \mathcal{O}(h^4))^2 = 1 - \pi^2 h^2 + \mathcal{O}(n^{-3})$$

3.) SOR-Verfahren:

$$T_\omega \equiv T_{SOR,\omega} \quad (\omega = 1 : T_1 = T_{GS})$$

$$(T_\omega u)_l = (1 - \omega)u_l + \frac{1}{2}\omega(\lambda_\omega u_{l-1} + u_{l+1})$$

$$\Rightarrow (1 - \omega)u_l + \frac{1}{2}\omega\sqrt{\lambda_\omega}(\sqrt{\lambda_\omega}u_{l-1} + \frac{1}{\sqrt{\lambda_\omega}}u_{l+1}) = \lambda_\omega u_l$$

Multiplikation der Gleichung mit $\sqrt{\lambda_\omega}^{-l}$ und Substitution von $v_l = \sqrt{\lambda_\omega}^{-l} \cdot u_l$ ergibt:

$$\begin{aligned} (1 - \omega)v_l + \frac{1}{2}\omega\sqrt{\lambda_\omega}(v_{l-1} + v_{l+1}) &= \lambda_\omega v_l \\ \Rightarrow \frac{1}{\omega\sqrt{\lambda_\omega}}(\lambda_\omega + \omega - 1)v_l &= \frac{1}{2}(v_{l-1} + v_{l+1}) \end{aligned}$$

D.h. $\lambda_\omega \in \text{spec}(T_\omega) \Rightarrow \frac{1}{\omega\sqrt{\lambda_\omega}}(\lambda_\omega + \omega - 1) \in \text{spec}(T_J) = \{\cos(k\pi h) : k = 1, \dots, n\}$

$$\Rightarrow (\sqrt{\lambda_\omega})^2 - \omega \cos(k\pi h)\sqrt{\lambda_\omega} + \omega - 1 = 0$$

Lösung der quadratischen Gleichung ergibt die Eigenwerte des SOR-Verfahrens.

($\omega = 1$: Eigenwerte des GS-Verfahrens: $\lambda_{\omega=1} = \cos(k\pi h)^2$)

Wir berechnen nun ω , so dass $\varrho(T_\omega)$ minimal ist.

Bild

Es folgt:

$$\begin{aligned} \omega_{\text{opt}} &= \frac{2}{1 + \sqrt{1 - \varrho(T_J)^2}} \geq 1 \\ \varrho_{\text{opt}} &= \omega_{\text{opt}} - 1 \end{aligned}$$

Für unser Beispiel und $h \rightarrow 0$:

$$\begin{aligned} 1 - \varrho(T_J)^2 &\approx 1 - (1 - \frac{1}{2}\pi^2 h^2)^2 \approx \pi^2 h^2 \\ \omega_{\text{opt}} &= \frac{2}{1 + \pi h} \approx 2(1 - \pi h) \\ \varrho_{\text{opt}} &\approx 1 - 2\pi h \end{aligned}$$

3.3.8 Konsistent geordnete Matrizen

Definition. $A \in \mathbb{R}^{n,n}$ heißt konsistent geordnet, wenn gilt: bzgl. der Zerlegung $A = D(Id - L - R)$ sind die Eigenwerte von $\alpha L + \frac{1}{\alpha}R$ unabhängig von $\alpha \in \mathbb{C} \setminus \{0\}$

Satz 11. Für konsistent geordnete Matrizen A mit $A_{ii} \neq 0$ und $\text{spec}(T_J) \subset (-1, 1)$ gilt

$$\varrho(T_{GS}) = \varrho(T_J)^2$$

und für das SOR-Verfahren gilt

$$\begin{aligned} \omega_{\text{opt}} &= \frac{2}{1 + \sqrt{1 - \varrho(T_J)^2}} \in (1, 2) \\ \varrho_{\text{opt}} &= \omega_{\text{opt}} - 1 \end{aligned}$$

Beweis. ÜA

□

Beispiele konsistent geordneter Matrizen:

- Tridiagonalmatrizen
- Block-Tridiagonalmatrizen
- Zwei-zyklische oder Red-Black-Matrizen

A heißt *zwei-zyklisch* oder *red-black-Matrix*, falls es eine Permutation gibt, so dass A auf die Form $\begin{bmatrix} D_1 & * \\ * & D_2 \end{bmatrix}$ mit Diagonalmatrizen D_1, D_2 gebracht werden kann.

3.3.9 Rechenaufwand

- 1.) Es sei $\varrho = \varrho(T) < 1$ und $\text{compl}(T) \approx \text{compl}(A)$. Der Aufwand zur Fehlerreduktion um den Faktor $\tau \in (0, 1)$ sei die Anzahl der Rechenoperationen um u_m mit

$$|u_m - u_*| \leq \tau |u_0 - u_*|$$

($Au_* = b$) zu erhalten.

Wir erhalten $\frac{|u_m - u_*|}{|u_0 - u_*|} \leq \varrho^m \stackrel{!}{\leq} \tau$.

$$\Rightarrow m \cdot \log(\varrho) \leq \log(\tau) \Rightarrow m \geq \frac{\log(\tau)}{\log(\varrho)} = \frac{\log(1/\tau)}{\log(1/\varrho)}$$

Aufwand := $m \cdot \text{compl}(T) \approx m \cdot \text{compl}(A)$

$\text{compl}(A) \sim n$

$$\log(1/\varrho) = |\log(\varrho)| \approx |\log(1 - \frac{1}{2}\pi^2 h^2)| \approx \frac{1}{2}\pi^2 h^2 \approx \frac{1}{2} \frac{\pi^2}{n^2}$$

für das Beispiel aus 3.7

$$\Rightarrow \text{Aufwand}_J \sim n \cdot n^2 \cdot \log(1/\tau) \sim n^3 \cdot \log(1/\tau)$$

$$\text{Aufwand}_{GS} \approx \frac{1}{2} \cdot \text{Aufwand}_J \sim n^3 \cdot \log(1/\tau)$$

$$\text{Aufwand}_{SOR} \sim n \cdot \frac{\log(1/\tau)}{h} \sim n^2 \cdot \log(1/\tau) \sim \frac{1}{n} \cdot \text{Aufwand}_{GS}$$

- 2.) SSOR-Verfahren ist nicht schneller als das Gauß-Seidel-Verfahren:

$$\varrho(\delta_\omega) = \varrho(T_{GS, \omega(2-\omega)}) \geq \varrho(T_{GS, 1})$$

- 3.) Diagonaldominante A , $A = \text{tridiag}(-1, a, -1)$ mit $a > 2$. Dann wird $\varrho(T_J) = 2/a < 1$ unabhängig von n . Der Aufwand ist dann $\sim n \cdot \log(1/\tau)$

Beispiel:

$$\begin{aligned}\partial_t u - u'' &= 0 \text{ in } (0, 1) \\ u(t, 0) = u(t, 1) &= 0 \quad \forall t > 0 \\ u(0, x) &= \varphi(x) \quad \forall x \in (0, 1)\end{aligned}$$

Wir diskretisieren:

$$\partial_t u(t, x) \approx \frac{u(t, x) - u(t - \Delta t, x)}{\Delta t}$$

Mit $u_i^k \approx u(t_k, x_i)$, $t_k = k\Delta t$, $x_i = ih$

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} + \frac{1}{h^2}(-u_{i+1}^{k+1} + 2u_i^{k+1} - u_{i-1}^{k+1}) = 0$$

In Matrixschreibweise:

$$\left(Id_n + \frac{\Delta t}{h^2} \text{tridiag}_n(-1, 2, -1) \right) u^{k+1} = u^k \quad (*)$$

wobei $u^k = [u_i^k]_{i=1, \dots, n}$
 u^0 (Startwert: $u_i^0 = \varphi(x_i)$)
 $\rightarrow u^1$ (Löse * für $k = 0$)
 $\rightarrow u^2$ (Löse * für $k = 1$)
 $\rightarrow \dots$

Matrix in (*) (Mult mit $\frac{h^2}{\Delta t}$)

$$\text{tridiag}_n(-1, 2 + \underbrace{\frac{h^2}{\Delta t}}_{=: a > 2}, -1)$$

Zusatz: 2D-Fall Bsp.: $\begin{bmatrix} -1 & \frac{1}{4} & -1 \\ & -1 & \end{bmatrix}$ auf $[0, 1]^2$ mit lexikographischer Anordnung.

Sei n Anzahl der Punkte in einer Raumrichtung, $N = n^2$, $h = \frac{1}{n+1} \approx \frac{1}{n} = \frac{1}{\sqrt{N}}$

$$\begin{aligned}\varrho_J &= 1 - \mathcal{O}(h^2) = 1 - \mathcal{O}\left(\frac{1}{N}\right) \\ \varrho_{\text{GS}} &= 1 - \mathcal{O}(h^2) = 1 - \mathcal{O}\left(\frac{1}{N}\right) \\ \varrho_{\text{opt}} &= 1 - \mathcal{O}(h) = 1 - \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)\end{aligned}$$

Weiter gilt:

$$\begin{aligned}\text{Aufwand}_J &\sim N^2 \\ \text{Aufwand}_{\text{GS}} &\sim N^2 \\ \text{Aufwand}_{\text{opt}} &\sim N \cdot \sqrt{N} = N^{3/2}\end{aligned}$$

Gaußelimination: Bandmatrix der Breite $m = n \approx \sqrt{N}$

$$\text{Aufwand}_{\text{GE}} = m^2 N \approx N^2$$

Abbruchkriterium für Iterationen:

$$|u_i - u_{i+1}| \leq \text{Tol} \quad \text{oder} \quad \underbrace{|Au_i - b|}_{\text{Residuum}} \leq \text{Tol} \cdot |b|$$

wobei $\text{Tol} \in \mathbb{R}_+$ die „Toleranz“ ist.

3.3.10 Idee Des Mehrgitterverfahrens

Problem: Aufwand ist noch $\mathcal{O}(n^\kappa)$ mit $\kappa > 1$.

Wir suchen schnelle Löser: $\kappa = 1$.

Gedämpftes Jakobi-Verfahren mit $\omega = 1/2$.

$$T_J = Id - \frac{1}{2} \left(\frac{1}{2} \right) A = Id - \frac{1}{4} A$$

im Beispiel aus 3.7. Dann ist

$$\text{spec}(T_{J,1/2}) = \left\{ 1 - \frac{1}{4} \lambda : \lambda \in \text{spec}(A) \right\} = \left\{ \frac{1}{2} (1 + \cos(k\pi n)) : k = 1, \dots, n \right\}$$

Für den Fehlervektor e_{i+1} gilt: $e_{i+1} = T_{J,1/2} e_i$. Sei $\{s_l\}_{l=1,\dots,n}$ Eigenbasis von A . Stelle e_i als Linearkombination der s_i dar:

$$e_i = \sum_{l=1}^n \alpha_l^{(i)} s_l$$

Dann folgt für $i + 1$:

$$e_{i+1} = T_{J,1/2} e_i = \sum_{l=1}^n \lambda_l \alpha_l^{(i)} s_l$$

Sei nun n gerade. Wir definieren

$$e_i^{\text{NF}} := \sum_{l=1}^{n/2} \alpha_l^{(i)} s_l \quad (\text{Niederfrequenter Anteil})$$

$$e_i^{\text{HF}} := \sum_{l=\frac{n}{2}+1}^n \alpha_l^{(i)} s_l \quad (\text{Hochfrequenter Anteil})$$

Bilder

Es gilt:

$$\begin{aligned} |T_J e_i^{\text{NF}}| &\leq |e_i^{\text{NF}}| \\ |T_J e_i^{\text{HF}}| &\leq \frac{1}{2} |e_i^{\text{HF}}| \end{aligned}$$

Idee: Verwende 2 Löser, einen für den NF-Anteil und gedämpftes Jakobi-Verfahren für den HF-Anteil

Bildchen

NF-Löser ist ein direktes Verfahren auf den Knoten echt unterhalb des feinsten Levels.

Trick: Verfahren analog für das Grobgitterproblem. Hierzu wird Jakobi heute noch verwendet.

Theorie: N -unabhängige Konvergenzrate.

Entwicklung des Mehrgitter-Verfahrens: 1965-1990.

3.4 Das CG-Verfahren

3.4.1 Das Gradientenverfahren

Definition. Sei $A \in \mathbb{R}^{n,n}$ spd und $b \in \mathbb{R}^n$ beliebig. Dann heißt die Abbildung $\varepsilon : \mathbb{R}^n \rightarrow \mathbb{R}$ def. durch

$$\varepsilon(v) := v \cdot Av - b \cdot v$$

die Energie.

ε ist strikt konvexe, nach unten beschränkte Funktion mit $\lim_{|v| \rightarrow \infty} \varepsilon(v) = \infty$. Weiter gilt

$$\varepsilon'' = A$$

Bildchen

Also hat ε ein eindeutiges Minimum in u_* und dies ist charakterisiert durch $\varepsilon'(u_*)[d] = 0 \quad \forall d \in \mathbb{R}^n$. Es gilt:

$$\varepsilon'(v)d = (Av - b) \cdot d \quad \forall d \in \mathbb{R}^n$$

Also folgt:

$$\varepsilon'(u_*) = 0 \Leftrightarrow Au_* = b$$

Idee: konstruiere Folge $\{u_k\}_k$, so dass $\varepsilon(u_{k+1}) < \varepsilon(u_k)$ ist mit $\lim_{k \rightarrow \infty} \varepsilon(u_k) = \min_{v \in \mathbb{R}^n} \varepsilon(v) = \varepsilon(u_*)$

Der steilste Anstieg in u_k ist

$$-\nabla \varepsilon(u_k) = -(Au_k - b) =: -r_k$$

Ansatz für k -ten Schritt:

$$u_{k+1} = u_k - \alpha_k r_k$$

Mit $\alpha_k \in \mathbb{R}$. Bestimme α_k wie folgt: Def.

$$\Phi(\alpha) := \varepsilon(u_k - \alpha r_k) \quad (\alpha \in \mathbb{R})$$

Φ ist nach unten beschränkt und strikt konvex mit $\lim_{|\alpha| \rightarrow \infty} \Phi(\alpha) = \infty$ Daher ex. α_k mit

$\Phi(\alpha_k) = \min_{\alpha \in \mathbb{R}} \Phi(\alpha)$ und es gilt: $\Phi'(\alpha_k) = 0$.

$$\begin{aligned} 0 &\stackrel{!}{=} \Phi'(\alpha_k) &= \varepsilon'(u_k - \alpha_k r_k) \cdot (-r_k) \\ &= (A(u_k - \alpha_k r_k) - b) \cdot (-r_k) \\ &= -(r_k - \alpha_k A r_k) \cdot r_k \\ &= -r_k \cdot r_k + \alpha_k A r_k \cdot r_k \end{aligned}$$

$$\Rightarrow \alpha_k = \frac{|r_k|^2}{Ar_k \cdot r_k}$$

Denn: $r_k \cdot Ar_k \stackrel{!}{=} 0 \Rightarrow r_k = 0 \Rightarrow Au_k = b$. Fertig!

Satz 12. Sei A spd, $(v, w)_A := Av \cdot w$, $\|v\|_A := (v, v)_A^{1/2}$
Ist $Au = b$ und $u_0 \in \mathbb{R}^n$, so konvergiert die Folge $\{u_k\}_k$ mit

$$u_{k+1} = u_k - \frac{|r_k|^2}{\|r_k\|_A^2} r_k, \quad k \geq 0, \quad r_k = Au_k - b$$

gegen u und es gilt:

$$\|u_{k+1} - u\|_A \leq \frac{\kappa - 1}{\kappa + 1} \|u_k - u\|_A = \left(1 - \frac{2}{\kappa + 1}\right) \|u_k - u\|_A$$

wobei $\kappa = \text{cond}_2(A) = \frac{\lambda_{\max}}{\lambda_{\min}}$. Bea.: $\|\cdot\|$ heißt Energienorm.

3.4.2 Fehlerminimierung auf Unterräumen

Algorithmus in 4.1 (CG-Verfahren) ist zu langsam.

Idee: $\{V_k\}_{k=1, \dots, n}$ sei eine Folge von Unterräumen des \mathbb{R}^n mit $\dim V_k = k$.

Ausgehend von $u_0 \in \mathbb{R}^n$ machen wir den Ansatz

$$u_{k+1} = u_k + p_{k+1} \quad \text{mit } p_{k+1} \in V_{k+1}$$

Wir definieren p_{k+1} durch

$$\|e_{k+1}\|_A = \|e_k + p_{k+1}\|_A \stackrel{!}{=} \min_{p \in V_{k+1}} \|e_k + p\|_A.$$

Wegen $0 \in V_{k+1}$ gilt $\|e_{k+1}\|_A \leq \|e_k\|_A$ und mit $V_n = \mathbb{R}^n$ ist $u_n = u$ die Lösung.

Wir definieren $\Phi : V^{k+1} \rightarrow \mathbb{R}$ durch

$$\Phi(p) := \|e_k + p\|_A^2 \quad (p \in V^{k+1})$$

Φ ist strikt konvex und es gilt $\Phi(p) \rightarrow \infty$ ($|p| \rightarrow \infty$). Das Minimum in p_{k+1} ist vollständig charakterisiert durch

$$\begin{aligned} 0 &\stackrel{!}{=} (\nabla \Phi(p_{k+1}), q)_A \\ &= 2(e_k + p_{k+1}, q)_A \\ &= 2(e_{k+1}, q)_A \quad \forall q \in V_{k+1} \end{aligned}$$

$\Rightarrow (e_{k+1}, q)_A = 0$ für alle $q \in V_{k+1}$. Wir nennen diese Eigenschaft von e_{k+1} *A-Orthogonalität* von e_{k+1} und V_{k+1} (Schreibweise $e_{k+1} \perp_A V_{k+1}$)

3.4.3 Krylovräume

Für die Idee aus 4.2 wählen wir zu $d_0 \in \mathbb{R}^n \setminus \{0\}$ die Räume

$$V_k \equiv V_k(A, d_0) = \text{span}\{d_0, Ad_0, \dots, A^{k-1}d_0\} \quad (k \geq 1)$$

Wir nehmen erstmal an, dass $\dim V_k = k$ ist. Wir errichten nun auf V_k eine orthogonale Basis mit dem Gram-Schmidt-Verfahren ausgehend von d_0 . Ansatz:

$$d_{k+1} = Ad_k - \sum_{l=0}^k \sigma_{kl} d_l$$

Bestimme die σ_{kl} durch die Forderung $(d_{k+1}, d_j)_A = 0 \quad j = 0, \dots, k$. (Tatsächlich benötigt man nur σ_{kk} und $\sigma_{k,k-1}$). Es gilt

$$V_k = \text{span}\{d_0, \dots, d_{k-1}\}$$

und Ad_k ist genau dann linear unabhängig von $\{d_0, \dots, d_k\}$ solange $A^{k+1}d_0 \notin \text{span}\{d_0, \dots, A^k d_0\}$ ist.

Beweis. Dazu: $d_1 \in Ad_0 + \text{span}\{d_0\} = Ad_0 + V_1 \subset V_2$

I. V.: $d_k \in A^k d_0 + \text{span}\{d_0, \dots, d_{k-1}\} \subset A^k d_0 + V_k \Rightarrow Ad_k \in A^{k+1} d_0 + AV_k \subset V_{k+1} \quad \square$

3.4.4 Das CG-Verfahren nach Hestenes/ Stiefel (1954)

Idee aus 4.3 aber mit einer Modifikation, die die Zahl der Koeffizienten reduziert.

$u_0 \in \mathbb{R}^n, r_0 = Au_0 - b =: d_0$,

$$d_{k+1} = r_{k+1} + \sum_{l=0}^k \sigma_{kl} d_l \quad (k \geq 0)$$

Lemma 3.

$$\text{span}\{d_l : l = 0, \dots, k\} \subset V_{k+1}(A, r_0) \equiv V_{k+1}$$

Beweis. $k = 1$: $\text{span}\{d_0\} = \text{span}\{r_0\} = V_1$

Ann.: $\text{span}\{d_l : l = 0, \dots, k\} \subset V_{k+1} \xrightarrow{!} d_{k+1} \in V_{k+2}$

$$\begin{aligned} d_{k+1} \in r_{k+1} + \text{span}\{d_0, \dots, d_k\} &\stackrel{\text{I.V.}}{=} Au_{k+1} - b + V_{k+1} \\ &\subset A(u_k + V_{k+1}) - b + V_{k+1} \\ &= \underbrace{r_k}_{\in V_{k+1}} + AV_{k+1} + V_{k+1} \\ &= AV_{k+1} + V_{k+1} \subset V_{k+2} \end{aligned}$$

\square

Konstruktion des Verfahrens

Es gelte $e_k \perp_A V_k$, $(d_i, d_j)_A = 0$ für $i, j \leq k, i \neq j$.

Geforderte Minimalität des Fehlers:

$$\begin{aligned} 0 &\stackrel{!}{=} (e_{k+1}, d_j)_A = A e_{k+1} \cdot d_j \\ &= A(u_{k+1} - u) \cdot d_j \\ &= r_{k+1} \cdot d_j \quad (j = 0, \dots, k) \end{aligned}$$

Weiter

$$0 = r_{k+1} \cdot A d_i = (r_{k+1}, d_i)_A \quad (i = 0, \dots, k-1)$$

Berechnung der σ_{kl} für $j = 0, \dots, k-1$:

$$0 \stackrel{!}{=} (d_{k+1}, d_j)_A \stackrel{\text{orth.}}{=} \underbrace{(r_{k+1}, d_j)_A}_{=0} + \sigma_{kj} \|d_j\|_A^2$$

$\Rightarrow \sigma_{kj} = 0$ für $j = 0, \dots, k-1$.

Es bleibt $j = k$:

$$0 \stackrel{!}{=} (d_{k+1}, d_k)_A = (r_{k+1}, d_k)_A + \sigma_{kk} \|d_k\|_A^2$$

$$\Rightarrow \beta_k := \sigma_{kk} = -\frac{(r_{k+1}, d_k)_A}{\|d_k\|_A^2} \Rightarrow d_{k+1} = r_{k+1} + \beta_k d_k$$

Aus $e_k \perp_A V_k$ folgt

$$\begin{aligned} (e_k, d_k)_A &= (e_k, r_k)_A + \beta_{k-1} \underbrace{(e_k, d_{k-1})_A}_{\in V_k} \\ &= (e_k, r_k)_A = A e_k \cdot r_k = |r_k|^2 \end{aligned}$$

Orthogonalisierung des Fehlers e_{k+1}

$$0 = (e_{k+1}, d_j)_A \text{ für } j < k$$

$\Rightarrow (e_k + p_{k+1}, \underbrace{d_j}_{\in V_k})_A = (p_{k+1}, d_j)_A$ für $j < k$. Also $p_{k+1} \sim d_k$, etwa $p_{k+1} = \alpha_k d_k$ und

damit

$$(*) \quad u_{k+1} = u_k - \alpha_k d_k$$

α_k folgt aus

$$\begin{aligned} (e_{k+1}, d_k)_A &= (e_k - \alpha_k d_k, d_k)_A \\ &= (e_k, d_k)_A - \alpha_k \|d_k\|_A^2 \\ &= |r_k|^2 - \alpha_k \|d_k\|_A^2 \end{aligned}$$

$$\Rightarrow \alpha_k = \frac{|r_k|^2}{\|d_k\|_A^2}$$

Damit lässt sich β_k eleganter schreiben: aus $(*)$ folgt:

$$r_{k+1} = r_k - \alpha_k A d_k$$

Dann ist

$$\begin{aligned}
(r_{k+1}, d_k)_A &= r_{k+1} \cdot Ad_k \\
&= r_{k+1} \cdot \left(-\frac{1}{\alpha_k} (r_{k+1} - r_k) \right) \\
&= -\frac{\|d_k\|_A^2}{|r_k|^2} (|r_{k+1}|^2 - \underbrace{r_{k+1} \cdot r_k}_{=0(z.z.)})
\end{aligned}$$

und es folgt $\beta_k = -\frac{(r_{k+1}, d_k)_A}{\|d_k\|_A^2} = \frac{|r_{k+1}|^2}{|r_k|^2}$

Noch z.z.: $r_{k+1} \cdot r_k = 0$:

$$\begin{aligned}
r_{k+1} \cdot r_k &= (r_k - \alpha_k Ad_k) \cdot r_k \\
&= |r_k|^2 - \alpha_k Ad_k \cdot r_k \\
&= |r_k|^2 - \frac{|r_k|^2}{\|d_k\|_A^2} Ad_k \cdot (d_k - \beta_{k-1} d_{k-1}) \\
&= |r_k|^2 - |r_k|^2 = 0
\end{aligned}$$

Der Algorithmus

Initialisierung $u_0 \in \mathbb{R}^n$, $r_0 = Au_0 - b$, $d_0 = r_0$

Iteration $k \geq 0$

$$\begin{aligned}
\alpha_k &= \frac{|r_k|^2}{d_k \cdot Ad_k} = \frac{|r_k|^2}{\|d_k\|_A^2} \\
u_{k+1} &= u_k - \alpha_k d_k \\
r_{k+1} &= r_k - \alpha_k Ad_k \\
\beta_k &= \frac{|r_{k+1}|^2}{|r_k|^2} \\
d_{k+1} &= r_{k+1} + \beta_k d_k
\end{aligned}$$

Wohldefiniert?

$r_k = 0 \Leftrightarrow Au_k = b \checkmark$

$d_{k+1} = 0$?

Dann wäre $\sum_{j=0}^{k+1} \gamma_j A^j d_0 = 0$ für $\gamma \in \mathbb{R}^{k+2} \setminus \{0\}$

$\gamma_0 \neq 0$: $\underbrace{d_0}_{=r_0} = \sum_{j=1}^{k+1} \frac{\gamma_j}{\gamma_0} A^j d_0$

$$\Rightarrow e_0 = A^{-1}r_0 = A^{-1}d_0 = -\sum_{j=0}^k \frac{\gamma_{j-1}}{\gamma_0} A^j d_0 \in V_{k+1}$$

Folgt genauso, falls $\gamma_0 = 0$ und $\gamma_1 \neq 0$ wäre.

$$e_{k+1} = e_k + p_{k+1} = e_{k-1} + p_k + p_{k+1} = \dots \in e_0 + V_{k+1} \subseteq V_{k+1} \text{ da } e_0 \in V_{k+1}$$

$$\Rightarrow e_{k+1} = 0, \text{ da } e_{k+1} \perp_A V_{k+1}$$

Aufwand	MV	VV	SV	Speicher
	1	2	3	$3N$

- MV $\hat{=}$ Matrix * Vektor
- VV $\hat{=}$ Skalarprodukte
- SV $\hat{=}$ Skalar * Vektor
- Speicher: zusätzlicher Speicher

3.4.5 Konvergenz des CG-Verfahrens

Ausgangspunkt:

$$\|e_k\|_A = \min_{p \in V_k} \|e_{k-1} + p\|_A$$

$V_k = \text{span}\{d_0, \dots, A^{k-1}d_0\}$. Aus $u_k \in u_0 + V_k$ folgt $e_k \in e_0 + V_k$

$$\Rightarrow e_k = e_0 + \sum_{j=0}^{k-1} u_{kj} A^j d_0$$

für geeignete u_{kj} . Es ist $d_0 = r_0 = Ae_0$, also gilt

$$e_k = e_0 + A \cdot \sum_{j=0}^{k-1} u_{kj} \cdot A^j e_0$$

Es gibt also ein Polynom $q_k \in \mathbb{P}_k^* = \{q \in \mathbb{P}_k : q(0) = 1\}$ mit $e_k = q_k(A)e_0$. D.h. wir können auch schreiben

$$\|e_k\|_A = \min_{q \in \mathbb{P}_k^*} \{\|q(A)e_0\|_A\}$$

A spd $\Rightarrow \exists$ ONB $\{z_l\}_l$ mit $Az_l = \lambda_l z_l$, λ_l die Eigenwerte von A . Dann gilt etwa

$$q(A)e_0 = q(A) \sum_{l=1}^n \alpha_l z_l = \sum_{l=1}^n \alpha_l q(\lambda_l) z_l$$

Für den Fehler e_k gilt:

$$\begin{aligned} \|e_k\|_A^2 &= \sum_{l=1}^n \alpha_l^2 q_k(\lambda_l)^2 \\ &\leq \max\{|q_k(\lambda_l)|^2\} \cdot \sum_{l=1}^n \alpha_l^2 \\ &\leq \max_{\lambda \in \text{spec}(A)} \{|q_k(\lambda)|^2\} \cdot \|e_0\|_A^2 \end{aligned}$$

Wir nehmen an, dass $\text{spec}(A) \subset [a, b] \subset \mathbb{R}_+$ ist. Dann ist

$$\max_{\lambda \in \text{spec}(A)} \{|q_k(\lambda)|^2\} \leq \max_{\lambda \in [a, b]} \{|q_k(\lambda)|^2\}$$

Insgesamt ist

$$\|e_k\|_A^2 \leq \min_{q \in \mathbb{P}_k^*} \max_{\lambda \in [a, b]} |q(\lambda)|^2 \cdot \|e_0\|_A^2$$

Den Vorfaktor nennen wir $\varrho_{a,b,k}^2$

Bildchen

Die Lösung ist lange bekannt, es gilt:

$$\varrho_{a,b,k} \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \quad \text{mit } \kappa = b/a > 1$$

$\kappa = 1 \Rightarrow b = a \Rightarrow A \sim Id$.

Optimal: $a = \lambda_{\min}(A)$, $b = \lambda_{\max}(A)$

$\Rightarrow \kappa$ ist die Kondition $\text{cond}_2(A)$

Satz 13. Das CG-Verfahren für eine symmetrisch positive Matrix A konvergiert für alle Startwerte wenigstens linear, d.h.

$$\|u_k - u\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|u_0 - u\|_A = 2 \left(1 - \frac{2}{\sqrt{\kappa} + 1} \right)^k \|u_0 - u\|_A$$

Beweis. A : Das Problem wird gelöst von $q_k(x) = \frac{T_k\left(\frac{b+a-2x}{b-a}\right)}{T_k\left(\frac{b+a}{b-a}\right)}$, d.h.

$$\max_{\lambda \in [a, b]} |q_k(\lambda)|^2 = \min_{q \in \mathbb{P}_k^*} \min_{\lambda \in [a, b]} |q(\lambda)|^2$$

T_k ist das k -te Tschebyscheff-Polynom:

$$T_k(t) = \cos(k \cdot \arccos(t))$$

Das Argument ist die Transformation $[a, b] \rightarrow [-1, 1]$ z.z.: T_k ist ein Polynom:

Sei $\theta := \arccos(t)$. Dann gilt:

$$\begin{aligned}
T_k(t) &= \cos(k\theta) \\
&= \frac{1}{2} (e^{ik\theta} + e^{-ik\theta}) \\
&= \frac{1}{2} \left((e^{i\theta})^k + (e^{-i\theta})^k \right) \\
&= \frac{1}{2} \left((\cos(\theta) + i \cdot \sin(\theta))^k + (\cos(\theta) - i \cdot \sin(\theta))^k \right) \\
&= \frac{1}{2} \cdot \sum_{l=0}^k \binom{k}{l} \cos(\theta)^{k-l} ((i \cdot \sin(\theta))^2 + (-i \cdot \sin(\theta))^2) \\
&= \sum_{\substack{l=0 \\ l \text{ gerade}}}^k \binom{k}{l} \underbrace{\cos(\theta)^{k-l}}_{=t} \cdot \underbrace{(i \cdot \sin(\theta))^2}_{=\sqrt{1-t^2}} \\
&\stackrel{l=2l'}{=} - \sum_{l'=0}^{\lfloor k/2 \rfloor} \binom{k}{2l'} t^{k-2l'} (1-t^2)^{l'} \in \mathbb{P}_k
\end{aligned}$$

Also $q_k \in \mathbb{P}_k$, $q_k(0) = 1$. Für $t \in [-1, 1]$ ist $|T_k(t)| \leq 1$ und mit $\kappa = \frac{b}{a}$ gilt

$$\max_{x \in [a, b]} |q_k(x)| \leq \frac{1}{T_k\left(\frac{\kappa+1}{\kappa-1}\right)}$$

Aus der obigen Rechnung:

$$T_k(t) = \frac{1}{2} \left((t + \sqrt{t^2 - 1})^k + (t - \sqrt{t^2 - 1})^k \right)$$

Weiter gilt:

$$\left(\frac{\kappa+1}{\kappa-1} \right)^2 - 1 = \frac{\kappa^2 + 2\kappa + 1 - (\kappa^2 - 2\kappa + 1)}{(\kappa-1)^2} = \frac{4\kappa}{(\kappa-1)^\kappa}$$

Insgesamt folgt:

$$\begin{aligned}
T_k\left(\frac{\kappa+1}{\kappa-1}\right) &\geq \frac{1}{2} \left(\frac{\kappa+1}{\kappa-1} + \sqrt{\left(\frac{\kappa+1}{\kappa-1}\right)^2 - 1} \right)^k \\
&\geq \frac{1}{2} \left(\frac{\kappa+1}{\kappa-1} + \frac{2\sqrt{\kappa}}{\kappa-1} \right)^k \\
&= \frac{1}{2} \left(\frac{(\sqrt{\kappa}+1)^2}{(\sqrt{\kappa}+1)(\sqrt{\kappa}-1)} \right)^k \\
&= \frac{1}{2} \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)^k
\end{aligned}$$

□

3.4.6 Vorkonditionierung

In der Praxis: $\kappa = \kappa_n \rightarrow \infty (n \rightarrow \infty)$ liefert zu langsame Konvergenz.

C sei spd. Dann schreiben wir

$$CAu = Cb.$$

Wir wenden das CG-Verfahren auf dieses System an. CA ist i.A. nicht symmetrisch. Wir benötigen die Symmetrie aber nur im $(\cdot, \cdot)_A$ -Skalarprodukt. Dies gilt: Seien $x, y \in \mathbb{R}^n$:

$$(CAx, y)_A = A(CA)x \cdot y = CAx \cdot Ay = Ax \cdot CAy = (x, CAy)_A$$

$$\Rightarrow \text{adj}_A(CA) = CA.$$

Damit schreibt sich das CG-Verfahren wie folgt:

Initialisierung:

$$u_0, r_0 = Au_0 - b, d_0 = Cr_0 = h_0$$

Iteration für $k \geq 0$:

$$\begin{aligned}\alpha_k &= \frac{r_k \cdot h_k}{d_k \cdot Ad_k} \\ u_{k+1} &= u_k - \alpha_k d_k \\ r_{k+1} &= r_k - \alpha_k Ad_k \\ h_{k+1} &= Cr_{k+1} \\ \beta_k &= \frac{r_{k+1} \cdot h_{k+1}}{(r_k \cdot h_k)} \\ d_{k+1} &= h_{k+1} + \beta_k d_k\end{aligned}$$

$C = Id$: CG wie vorher.

$h_{k+1} = h_k - \alpha_k (Ad_k)$ (Ad_k ist das Residuum der neuen Gleichung.)

Der Kryllorraum ist $V_k(CA, d_0)$

Abbruch:

$$\sqrt{\frac{|r_k \cdot h_k|}{b \cdot cb}} \leq \text{Tol}$$

In der Fehlerabschätzung steht dann $\kappa = \kappa(CA)$.

Am besten: $C \approx A^{-1}$, aber auch $\text{compl}(C) \approx \text{compl}(A)$ - Widerspricht sich!

Beispiele:

- $C = \text{diag}(A)^{-1}$
Billig, aber nur sinnvoll, wenn die Diagonale stark variiert.

- $C = T$, T ein Schritt eines konvergenten iterativen Verfahrens. Etwa T_{SSOR} (symmetrisch!)
Man erhält $\kappa = \mathcal{O}(\sqrt{N})$ statt $\mathcal{O}(N)$ für das Poissonproblem auf $[0, 1]^2$
oder $C = T_{\text{Multigrid}} \Rightarrow \kappa(CA) = \mathcal{O}(1)$

Bemerkungen

- Die Konvergenz des CG-Verfahrens beschleunigt im Laufe der Iteration
Bildchen
- Die Konvergenz des CG-Verfahrens hängt von der Eigenwertverteilung ab.
Bildchen

3.5 GMRES (Generalized minimal residuals, 1986)

3.5.1 Minimale Residuen

Problem: CG funktioniert nur für symmetrisch positiv definite Matrizen A
In vielen Problemen ist A weder symmetrisch noch positiv definit:

$$\begin{aligned} -u'' + \beta u' &= f & (\text{in } \mathbb{R}) \\ -\Delta u + \underbrace{b \cdot \nabla u}_{\text{Transportterm}} &= f & (\text{in } \mathbb{R}^d) \end{aligned}$$

Ziel: Nutze Prinzipien aus 4

Idee: A invertierbar $\Rightarrow A^\top A$ ist spd.

e_k Fehler $\Rightarrow \|e_k\|_{A^\top A} = \|Ae_k\|_2 = \|r_k\|_2 = \|Au_k - b\|_2$ (i.F.: $|\cdot| = \|\cdot\|_2$)

$$Au = b \Rightarrow A^\top Au = A^\top b$$

„CG-Verfahren für Normalengleichungen“ (ÜA)

Die Konvergenz, die sich aus den Fehlerabschätzungen von 4,5 ergibt, ist meist viel zu langsam: $\kappa(A^\top A) \stackrel{\text{i.A.}}{\gg} \kappa(A)$. (Wir arbeiten hier auf $V_k(A^\top A)$!)

Idee: Nutze $\|\cdot\|_{A^\top A}$ für den Fehler, aber minimiere auf $V_k = V_k(A, d_0)$. Finde $u_k \in u_0 + V_k$ mit

$$|r_k| = \|Au_k - b\| = \min_{v_k \in u_0 + V_k} \|Av_k - b\| \quad (*)$$

$$V_k \in u_0 + V_k \Rightarrow v_k = u_0 + \sum_{l=0}^{k-1} \alpha_l A^l r_0, \text{ falls } d_0 \sim r_0$$

$$\begin{aligned} \Rightarrow Av_k - b &= \underbrace{Au_0 - b}_{=r_0} + A \cdot \sum_{l=0}^{k-1} \alpha_l A^l r_0 \\ &= \left(Id + A \cdot \sum_{l=0}^{k-1} \alpha_l A^l \right) r_0 \\ &= q(A) r_0 \quad \text{mit einem } q \in \mathbb{P}_k^* \end{aligned}$$

Für das Minimum gilt daher:

$$|r_k| = \min_{q \in \mathbb{P}_k^*} |q(A) \cdot r_0| \leq \min_{q \in \mathbb{P}_k^*} \|q(A)\|_2 |r_0| \quad (**)$$

Daraus gewinnen wir Fehlerabschätzungen

Satz 14. (Fehlerabschätzung für GMRES)

Sei $A \in \mathbb{R}^{n,n}$ regulär, u_k Lösung von

$$|Au_k - b| = \min_{v_k \in u_0 + V_k} |Av_k - b|$$

1.) A diagonalisierbar mit $A = XDX^{-1}$, D diagonal, $X, D \in \mathbb{C}^{n,n}$, so gilt:

$$|r_k| \leq \text{cond}_2(X) \cdot \max_{\lambda \in \text{spec}(A)} |q(\lambda)| |r_0| \quad \forall q \in \mathbb{P}_k^*$$

2.) A normal ($AA^\top = A^\top A$). Dann gilt 1.) mit $\text{cond}_2(X) = 1$

3.) $\|Id - A\|_2 \leq \varrho < 1 \Rightarrow |r_k| \leq |r_0| \varrho^k$

Beweis. 1.) $q(A) = q(XDX^{-1}) = Xq(D)X^{-1}$

Also ist

$$\begin{aligned} \|q(A)\|_2 &\leq \|X\|_2 \cdot \|X^{-1}\|_2 \cdot \|\text{diag}(q(\lambda_1), \dots, q(\lambda_n))\|_2 \\ &\leq \text{cond}_2(X) \cdot \max_{\lambda \in \text{spec}(A)} |q(\lambda)| \end{aligned}$$

Behauptung folgt aus (**)

2.) A normal $\Rightarrow X$ orthonormal $\Rightarrow \text{cond}_2(X) = 1$.

3.) Wähle $q(t) := (1 - t)^k$. Dann $q \in \mathbb{P}_k^*$.

$$\|q(A)\|_2 = \|(Id - A)^k\|_2 \leq \|Id - A\|_2^k \leq \varrho^k$$

$$\Rightarrow \min_{q \in \mathbb{P}_k^*} \|q(A)\|_2 \leq \varrho^k \text{ Behauptung folgt mit (**)}$$

□

Bemerkung Ist $\text{spec}(A) \subset [a, b] \subset \mathbb{R}$ für $0 < a < b$, so kann man die Abschätzung aus 4.5 verwenden mit $\kappa = b/a$

3.5.2 Konstruktion des GMRES-Verfahrens

Schritt 1: Konstruiere eine euklidisch orthonormale Basis des Krylovraumes (Gram-Schmidt)

Start: $d_0 = \frac{r_0}{|r_0|}$ (o.B.d.A $|r_0| \neq 0$)

Iteration: für $k \geq 0$:

$$\begin{aligned}\sigma_{kj} &= Ad_k \cdot d_j \quad j = 0, \dots, k \\ v_{k+1} &= Ad_k - \sum_{j=0}^k \sigma_{kj} d_j \\ \sigma_{k,k+1} &= |v_{k+1}| \\ d_{k+1} &= \frac{v_{k+1}}{|v_{k+1}|}\end{aligned}$$

Aufwand im k -ten Schritt: $\frac{MV}{1} \mid \frac{VV}{k+3} \mid \frac{SV}{k+2}$ Speicher: $(k+2)n + \mathcal{O}(k^2)$ insgesamt

Der Aufwand über K Schritte ist $\mathcal{O}(K^2) \sim \sum_{k=1}^K \mathcal{O}(k)$.

Speicher: $\mathcal{O}(K)n + \mathcal{O}(K^2)$

Schritt2: Minimierung des Residuums

$$\begin{aligned}|r_k| &= \min_{v_k \in u_0 + V_k} |Av_k - b| \\ &= \min_{z_k \in V_k} \underbrace{|Au_0 - b|}_{=r_0} + |Az_k| \\ &= \min_{z_k \in V_k} |Az_k - \beta_0 d_0| \quad \text{mit } \beta_0 = -|r_0|\end{aligned}$$

Es sei $P_k : V_k \rightarrow \mathbb{R}^k$ die orthonormale Projektion mit $P_k d_{l-1} = \vec{i}_l$

Aus $Ad_k = \sigma_{k,k+1} d_{k+1} + \sum_{j=0}^k \sigma_{kj} d_j$ folgt $A|_{V_{k+1}} \rightarrow V_{k+2}$, d.h. in der Basis $\{d_0, \dots, d_k\}$ hat A „Hessenberggestalt“:

$$A|_{V_k} = \begin{bmatrix} \sigma_{00} & \sigma_{10} & & & \\ \sigma_{01} & \sigma_{11} & & & * \\ & \ddots & \ddots & & \\ & & \ddots & \sigma_{k+1,k+1} & \\ & & & \sigma_{k,k+1} & \end{bmatrix} \in \mathbb{R}^{k+1}$$

Mit $A_k := P_{k+1} A P_k$ folgt

$$|r_k| = \min_{z_k \in V_k} |P_{k+1} (A \underbrace{P_k^\top P_k}_{=Id_{V_k}} z_k - \beta d_0)| = \min_{\omega_k \in \mathbb{R}^k} |A_k \omega_k - \beta_0 \vec{i}_1|$$

Trick: Mittels orthonormaler Matrizen $L_1, \dots, L_K \in \mathbb{R}^{k+1, k+1}$ kann man erreichen, dass $L_k \cdot \dots \cdot L_1 A_k = \begin{bmatrix} R_k & \\ & 0 \end{bmatrix} \in \mathbb{R}^{k+1, k}$ und R_k ist r.o. Dreiecksmatrix. Dann

$$\begin{aligned} |r_k| &= \min_{\omega_k \in \mathbb{R}^k} \left| \underbrace{L_k \cdot \dots \cdot L_1}_{\text{orthonormal}} (A_k \omega_k - \beta_0 \vec{i}_1) \right| \\ &= \min_{\omega_k \in \mathbb{R}^k} \left\| \begin{bmatrix} R_k \omega_k \\ 0 \end{bmatrix} - \begin{bmatrix} b_k \\ \varrho_k \end{bmatrix} \right\| \quad \text{mit } b_k \in \mathbb{R}^k, \varrho_k \in \mathbb{R} \end{aligned}$$

$$\Rightarrow |r_k| = \min_{\omega_k \in \mathbb{R}^k} (|R_k \omega_k - b_k|^2 + \varrho_k^2)^{1/2}$$

Das Minimum wird für $\omega_k = R_k^{-1} b_k$ angenommen ($\text{Rang}(R_k) = \text{Rang}(A_k) = \text{Rang}(A|v_k) = k$, falls $\dim(V_k) = k$) und dann ist $|r_k| = \varrho_k$.

Zwar ist ω_k billig berechenbar ($\mathcal{O}(k^2)$ Multiplikationen, da Dreiecksmatrix), aber ϱ_k ist bekannt ohne ω_k zu kennen! Wir berechnen ω_k erst, wenn ϱ_k klein genug ist oder $k = k_{max}$ erreicht ist.

Bemerkung:

$$L_j = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & c & s & \\ & & & -s & c & \\ & & & & & 1 \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{bmatrix} \in \mathbb{R}^{k+1, k+1}, \quad j \leq k, \quad c^2 + s^2 = 1$$

Man kann c bestimmen aus der Bedingung, dass der $k+1, k$ -te Eintrag von $L_k(L_{k-1} \cdot \dots \cdot L_1 A_k) = 0$ wird.

Speicher und Anwendung der Matrizen L_j sind $\mathcal{O}(k)$

Algorithmus: Start: $u_0 \in \mathbb{R}^n$, $r_0 = Au_0 - b \neq 0$, $d_0 := r_0/|r_0|$, $b_0 := -|r_0|\vec{i}_1$

Iteration für $k \geq 0$

- Stopp, falls $\varrho_k = |b_{k+1, k+1}| < \text{Tol}$, sonst $k \rightarrow k+1$
- Berechne σ_{kj} für $j = 0, \dots, k$, $d_{k+1}, \sigma_{k, k+1}$
- Berechne $\left[\widetilde{R_{k+1} j} \right]_{j=0, \dots, k} = L_k \cdot \dots \cdot L_1 [\sigma_{kj}]_{j=0, \dots, k}$
 $(L_j \in \mathbb{R}^{k, k} \rightarrow \begin{bmatrix} L_j & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{k+1, k+1})$
- Berechne die Rotation L_{k+1}
- Berechne $[R_{k+1} j]_{j=0, \dots, k} = L_{k+1} [\widetilde{R_{k+1} j}]_{j=0, \dots, k}$

- Berechne $\omega_k = R_{k+1}^{-1} b_{k+1}$
 $u = u_0 + \sum_{j=0}^k \omega_{k+1,j} \cdot d_j$

$$\begin{bmatrix} R_k \\ 0 \dots 0 \end{bmatrix} \rightarrow \left[\begin{array}{c|c} R_k & \begin{matrix} \vdots \\ \sigma_{kj} \\ \vdots \end{matrix} \end{array} \right] \xrightarrow[\text{auf letzte Spalte}]{L_k \dots L_1} \left[\begin{array}{c|c} R_k & \begin{matrix} \vdots \\ \tilde{\sigma}_{kj} \\ * \end{matrix} \end{array} \right] \xrightarrow{L_{k+1}} \begin{bmatrix} R_{k+1} \\ \vdots \\ 0 \end{bmatrix}$$

$$\text{Rechte Seite: } \begin{bmatrix} b_k \\ 0 \end{bmatrix} \xrightarrow{L_{k+1}} \begin{bmatrix} \vdots \\ (*) \end{bmatrix}$$

Satz 15. DAS GMRES-Verfahren (in exakter Arithmetik) ist für invertierbare Matrizen durchführbar und erzeugt eine Folge abnehmender Residuen

$$|r_{k+1}| \leq |r_k|$$

(wobei $(r_k = Au_k - b)$ und $r_n = 0$)

Unter geeigneten Voraussetzungen fällt $|r_k|$ streng monoton (siehe Theorem 5.1).

Der Aufwand für k Schritte ist $\mathcal{O}(k^2 N)$

Speicher: $\mathcal{O}(kN) + \mathcal{O}(k^2)$

Beweis. Fehlt: $\dim(V_k) = k$, bzw. $v_{k+1} \neq 0$ im 1. Schritt (GS). Dann ist $Ad_k \in \text{span}\{d_0, \dots, d_k\}$

$$\Rightarrow e_0 = A^{-1}r_0 \sim A^{-1}d_0 \stackrel{\text{wie 4.4}}{\in} \text{span}\{d_0, \dots, d_k\} = V_{k+1}$$

$$\Rightarrow e_{k+1} \in V_{k+1}, e_{k+1} \perp_{A^T A} V_{k+1} \Rightarrow e_{k+1} = 0$$

$$0 \stackrel{!}{=} v_{k+1} = Ad_k + \sum_{j=0}^k \sigma_{kj} \cdot d_j \quad \square$$

Bemerkung

- 1.) In der Praxis darf k nicht zu groß werden GMRES(k_{\max}) bricht nach k_{\max} Schritten ab und startet mit der bis dahin erhaltenen Lösung neu (Restart). Typisch: GMRES(5) bzw. GMRES(25)

- 2.) Es sei $A = \begin{bmatrix} 0 & & & 1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}$. Man kann $b, u_0 \in \mathbb{R}^n$ wählen mit $u_0 = u_1 = \dots u_{n-1}$, $u_n = u$ (u die exakte Lösung)

$$\text{Also } |r_0| = \dots = |r_{n-1}| \neq 0, r_n = 0$$

$$\text{spec}(A) = \{\lambda \in \mathbb{C} : \lambda^n = 1\}$$

RESTART-GMRES konv. nicht.