ÜBER EINIGE CG-ÄHNLICHE VERFAHREN ZUR LÖSUNG LINEARER GLEICHUNGSSYSTEME

Dissertation zur Erlangung des Doktorgrades
der Fakultät für Mathematik
der Julius-Maximilians-Universität Würzburg

vorgelegt von

Roland Freund

aus

Gochsheim

Würzburg 1983

Eingereicht am:	10. Juni 1983
1. Berichterstatter	Prof. Stoer
2. Berichterstatter	Prof. Velte
Tag d. mündl. Prüfung	3. Aug. 1983

Promotion 7. Sep. 1983

INHALTSVERZEICHNIS

Einleitung

Die Methode der konjugierten Gradienten (CG-Algorithmus) von Hestenes und Stiefel [16] ist ein Verfahren zur Lösung linearer Gleichungssysteme

$$Ax = b \tag{0.1}$$

mit positiv definiter n×n-Matrix A, welches zugleich Züge iterativer und direkter Methoden aufweist: Ausgehend von einem Startwert x_o wird eine Folge von Vektoren $x_k \in \mathbb{R}^n$ geliefert, die, zumindest bei rundungsfehlerfreier Rechnung, nach spätestens n Iterationen mit der exakten Lösung $\bar{x}$ von (0.1) abbricht. Für die praktische Anwendung ist bekanntlich der iterative Charakter wichtiger (vgl. Reid [28], Axelsson [1], Chandra [6]): In vielen Fällen, insbesondere bei - im Sinne der Spektralkondition $\kappa(A)$ - gut konditioniertem A, erhält man nämlich bereits nach sehr viel weniger als n Schritten eine hinreichend genaue Näherung x_k für $\bar{x}$, wie sich als Folge der Minimaleigenschaft

$$(x_k - \bar{x})^T A(x_k - \bar{x}) = \min_{x \in x_o + S_k} (x - \bar{x})^T A(x - \bar{x})$$

ergibt. Hier bezeichnet $S_k := \mathrm{span}\{r_o, Ar_o, \ldots, A^{k-1}r_o\}$ den k-ten von $r_o := b - Ax_o$ erzeugten Krylov-Unterraum. Schließlich sei noch erwähnt, daß der CG-Algorithmus wie andere iterative Verfahren von A lediglich bei der Bildung eines Matrix-Vektor-Produkts $A \cdot x$ pro Iteration Gebrauch mancht und sich daher vor allem für sehr große, aber dünn besetzte Systeme eignet.

Zur Lösung von (0.1) mit symmetrischer, aber indefiniter oder unsymmetrischer Koeffizientenmatrix bietet sich an, das CG-Verfahren auf die Normalgleichungen $A^TAx = A^Tb$, wie von Hestenes und Stiefel [16] vorgeschlagen wurde, oder auf $AA^Ty = b$, $x := A^Ty$ (Algorithmus von Craig [8,12]) anzuwenden; wegen $\kappa(A^TA) = \kappa(AA^T) = \kappa(A)^2$ ist dann allerdings im allgemeinen nur langsame Konvergenz zu erwarten. In den letzten Jahren wurden dagegen ähnlich wie die CG-Methode gebaute Verfahren vorgeschlagen, die das System Ax=b unmittelbar angehen: So für symmetrische, indefinite A von Paige und Saunders [26] der

SYMMLQ-Algorithmus, in welchem versucht wird, Näherungen x_k für $\bar{x}$ zu berechnen, die der Galerkin-Bedingung

$$(0.2) \qquad b - Ax_k \perp S_k, \quad x_k \in x_o + S_k$$

genügen. Diese x_k, sofern sie überhaupt definiert sind, erfüllen allerdings keine Minimierungseigenschaft. SYMMLQ ist jedoch eng verwandt mit dem "Orthogonal Direction"-Verfahren von Fridman [14], das Vektoren x_k mit

$$(0.3) \qquad \|x_k - \bar{x}\| = \min_{x \in x_o + AS_k} \|x - \bar{x}\|$$

liefert. Leider ist der OD-Algorithmus, trotz aller Eleganz, numerisch instabil, so daß in der Rechenpraxis die Minimaleigenschaft (0.3) bald verlorengeht und das Verfahren sogar divergiert. Von Fletcher [13] bzw. Chandra [6] wurde der MCR-Algorithmus für symmetrische, indefinite A angegeben; die Iterierten x_k sind hier durch

$$(0.4) \qquad \|b - Ax_k\| = \min_{x \in x_o + S_k} \|b - Ax\|$$

definiert.

In Teil I dieser Arbeit wird nun das OD-Verfahren theoretisch näher analysiert; insbesondere schlagen wir eine einfache numerisch stabile Version ("STOD-Algorithmus") dieser Methode vor. Bei der theoretischen Untersuchung werden Abschätzungen für die Fehler $\|x_k - \bar{x}\|$ angegeben, die zeigen, welchen Einfluß die Eigenwertverteilung von A auf die Konvergenzgeschwindigkeit bzw. den Abbruchindex des Verfahrens hat; analoge Resultate waren bislang nur für CG- und MCR-Algorithmus bekannt. Unsere Fehlerschranken motivieren außerdem, die OD-Methode in Verbindung mit Prekonditionierungstechniken anzuwenden, und wir stellen entsprechende Algorithmen vor.

Teil II der vorliegenden Dissertation beschäftigt sich mit CG-ähnlichen Methoden zur Lösung von Gleichungssystemen Ax=b mit unsymmetrischem A. Im allgemeinen führt nun der Versuch, durch (0.2) oder (0.4) definierte Iterierte x_k zu berechnen, auf Algorithmen, die in jedem Schritt Informationen aus sämtlichen

vorhergehenden Iterationen benötigen und dadurch sehr aufwendig sind (vgl. Axelsson [2], Young und Jea [38]). Ähnlich effektive Verfahren wie für symmetrische A lassen sich jedoch in einem wichtigen Spezialfall angeben: Für Matrizen der Form $A=I-N$, $N=-N^T$ (der allgemeinere Fall, daß A positiv reell, d.h. $(A+A^T)/2$ positiv definit, ist, läßt sich durch eine Art Prekonditionierung auf diesen Spezialfall reduzieren) fanden Concus,Golub [7] und Widlund [36] einen einfachen Algorithmus (CGW) zur Berechnung der Galerkin-Näherungen (0.2). Wir stellen der CGW-Methode zwei weitere effiziente Verfahren für Matrizen der Gestalt $A=I-N$ an die Seite: Der STOD-Algorithmus liefert Iterierte x_k mit

$$\|x_k - \bar{x}\| = \min_{x \in x_o + A^T S_k} \|x - \bar{x}\| ,$$

der MCR-Algorithmus berechnet die durch (0.4) definierten Näherungen. Bei der theoretischen Analyse der beiden Verfahren zeigt sich, daß die Konvergenzgeschwindigkeit jeweils durch die Größe der Matrix N bestimmt wird. Es ergeben sich ferner enge Zusammenhänge zwischen dem STOD-Algorithmus, dem Verfahren von Craig und der CGW-Methode, und wir weisen nach, daß gewisse Teilfolgen der vom CGW-Verfahren erzeugten Näherungen Minimaleigenschaften besitzen. Schließlich stößt man im Zusammenhang mit Fehlerabschätzungen für den MCR-Algorithmus auf ein auch an sich interessantes Problem der Approximationstheorie:

$$\min_{p \in \Pi_k} \max_{-\Lambda \leq t \leq \Lambda} |p(1+it)| \quad (=: m_k),$$

wobei $\Pi_k := \{p(z) \equiv 1+\sigma_1 z+\sigma_2 z^2+\cdots+\sigma_k z^k \mid \sigma_1, \sigma_2, \ldots, \sigma_k \in \mathbb{R}\}$ und $\Lambda>0$. Für diese m_k geben wir obere und untere Schranken an, die in bestimmtem Sinn asymptotisch exakt sind.

Im III. Teil werden die beschriebenen neuen Algorithmen an großen linearen Gleichungssystemen (Dimension ≤ 1000), die bei der Diskretisierung partieller Differentialgleichungen auftreten, getestet, und mit anderen Verfahren verglichen. Die Resultate bestätigen die theoretischen Eigenschaften und belegen die numerische Stabilität und praktische Brauchbarkeit der vorgeschlagenen Algorithmen.

An dieser Stelle möchte ich Herrn Professor Dr. J. Stoer herzlichst danken. Seine Anregungen und Ideen haben wesentlich zum Zustandekommen dieser Arbeit beigetragen.

I. DER OD-ALGORITHMUS FÜR GLEICHUNGSSYSTEME MIT SYMMETRISCHER MATRIX

1. Bezeichnungen. Vorbereitungen

Wir wollen zunächst einige Bezeichnungen vereinbaren; es sei vorausgeschickt, daß stets reelle Vektoren und Matrizen betrachtet werden.
Für Vektoren $v_1, v_2, \ldots, v_k$ des $\mathbb{R}^n$ sei $[v_1, v_2, \ldots, v_k]$ der von diesen Vektoren aufgespannte Unterraum, $(v_1, v_2, \ldots, v_k)$ die mit diesen Vektoren gebildete $n \times k$-Matrix.
$U^{\perp}$ sei wie üblich der zu einem linearen Teilraum $U \subseteq \mathbb{R}^n$ gehörige Orthogonalraum.
Mit $\|x\| = (x^T x)^{1/2}$ sei stets die euklidische Norm, mit $\|x\|_B = (x^T B x)^{1/2}$ (B eine positiv (semi-)definite Matrix) die B-(Halb-)Norm des Vektors x, mit $\|A\|$ die von der euklidischen Norm induzierte Matrixnorm, die sogenannte Spektralnorm, einer Matrix A bezeichnet.
Für eine symmetrische, nichtsinguläre Matrix A mit den Eigenwerten $0 < |\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_n|$ sei $\kappa(A) = |\lambda_n| / |\lambda_1|$ die Konditionszahl bezüglich der Spektralnorm.

Mit

$$R(A) := \{Ax \mid x \in \mathbb{R}^p\}$$

bezeichnen wir das Bild, mit

$$N(A) := \{x \in \mathbb{R}^p \mid Ax = 0\}$$

den Kern einer $n \times p$-Matrix A. Man beachte, daß wegen $R(A)^{\perp} = N(A^T)$ die Zerlegung

$$\mathbb{R}^n = R(A) \oplus N(A^T)$$

zur Verfügung steht.
Falls A nichtquadratisch oder singulär ist, so sei A^+ die Pseudoinverse von A.
Für zwei Vektoren $u, v \in \mathbb{R}^n \setminus \{0\}$ bezeichnen wir mit $\sphericalangle(u,v)$, definiert durch

$$\cos \sphericalangle(u,v) = \frac{|u^T v|}{\|u\| \, \|v\|}, \quad 0 \leq \sphericalangle(u,v) \leq \frac{\pi}{2},$$

den spitzen Winkel zwischen u und v. Ist außerdem $U \subseteq \mathbb{R}^n$ ein linearer Teilraum mit $U \neq \{0\}$, so setzen wir

$$\sphericalangle(v,U) := \min_{u\in U\setminus\{0\}} \sphericalangle(v,u);$$

ferner wird von

$$(1.1) \qquad \tan^2\sphericalangle(v,U) = \min_{u\in U\setminus\{0\}} \frac{\|u\|^2\,\|v\|^2 - (u^Tv)^2}{(u^Tv)^2},$$

sowie von

$$(1.2) \qquad \sin\sphericalangle(v,U) = \frac{\|(I-P)v\|}{\|v\|}, \quad \cos\sphericalangle(v,U) = \frac{\|Pv\|}{\|v\|}$$

Gebrauch gemacht, dabei ist $P: \mathbb{R}^n \to U$ die Matrix der orthogonalen Projektion auf U.
Den ganzen Anteil einer reellen Zahl α wollen wir

$$[\alpha] := \max\{k\in\mathbb{Z} \mid k \leq \alpha\}$$

nennen.
Schließlich führen wir die Abkürzung

$$\Pi_k := \{p(t) \equiv 1+\sigma_1 t+\sigma_2 t^2+ \cdots +\sigma_k t^k \mid \sigma_1,\ldots,\sigma_k \in \mathbb{R}\}$$

ein und bezeichnen mit

$$T_k(t) \equiv \frac{1}{2}\left((t+\sqrt{t^2-1})^k + (t-\sqrt{t^2-1})^k\right), \quad k=0,1,2,\ldots$$

die Tschebyscheffpolynome. Eine wichtige Optimalitätseigenschaft (siehe etwa [37]) dieser Polynome halten wir fest in folgendem

(1.3) Lemma: Sei $0 < \alpha < \beta$. Es gilt:

$$\min_{p\in\Pi_k} \max_{\alpha\leq t\leq\beta} |p(t)| = \left(T_k\left(\frac{\beta+\alpha}{\beta-\alpha}\right)\right)^{-1}.$$

Eine wesentliche Rolle wird im weiteren ein Verfahren von Lanczos [18] spielen, welches auf einfache Weise Orthogonalbasen der Krylovunterräume

$$U_k := [f, Af, A^2f, \ldots, A^{k-1}f], \quad k=1,2,\ldots,$$

- dabei ist $f\in\mathbb{R}^n$ und A eine symmetrische, nicht notwendig nichtsinguläre $n\times n$-Matrix - liefert.

(1.4) Lanczos-Algorithmus:

Start: Setze $p_o = f$, $p_{-1} = 0$.

Für $k=0,1,2,\ldots$:

Falls $p_k = 0$: stop,

andernfalls setze

$$\gamma_k = p_k^T A p_k / p_k^T p_k \quad , \quad \delta_k = \begin{cases} 0 & \text{für } k=0 \\ p_k^T p_k / p_{k-1}^T p_{k-1} & \text{für } k>0 \end{cases} ,$$

$$p_{k+1} = A p_k - \gamma_k p_k - \delta_k p_{k-1} .$$

Einige wohlbekannte Eigenschaften dieses Algorithmus stellen wir in folgendem Satz zusammen, den wir der Vollständigkeit wegen auch beweisen wollen.

(1.5) Satz: a) Es gibt eine kleinste ganze Zahl m, $0 \leq m \leq n$ mit $p_m = 0$.

b) $p_j^T p_k = 0$ für $0 \leq j < k \leq m$.

c) $[p_o, p_1, \ldots, p_{k-1}] = U_k$ für $1 \leq k \leq m$.

d) m ist die kleinste ganze Zahl mit der Eigenschaft:

$f, Af, A^2 f, \ldots, A^m f$ sind linear abhängig.

e) m ist die Anzahl der Eigenräume von A, in denen f von Null verschiedene Komponenten hat:

(1.6) $$f = \sum_{j=1}^{m} \eta_j z_j \quad , \quad \eta_1, \ldots, \eta_m \neq 0 ,$$

dabei sind $z_1, \ldots, z_m$ orthonormale Eigenvektoren zu paarweise verschiedenen Eigenwerten $\lambda_1, \ldots, \lambda_m$ von A.

f) $[p_o, p_1, \ldots, p_{m-1}] = [z_1, z_2, \ldots, z_m] = U_m$.

Beweis: zu a)b) Sei $l \geq 0$ mit $p_k \neq 0, 0 \leq k < l$; wir zeigen durch Induktion

nach k, daß $p_j^T p_k = 0$ für $0 \leq j < k \leq l$ gilt. Für k=1 ergibt sich $p_o^T p_1 = p_o^T A p_o - \gamma_o p_o^T p_o = 0$ nach Definition von γ_o. Sei die Behauptung bereits für alle Indizes $\leq k$ bewiesen, und sei $k<l$. Dann folgt mit

$$p_j^T A p_k = (p_{j+1} + \gamma_j p_j + \delta_j p_{j-1})^T p_k = \begin{cases} p_k^T p_k & \text{für } j=k-1 \\ 0 & \text{für } 0 \leq j < k-1 \end{cases}$$

$$p_j^T p_{k+1} = p_j^T A p_k - \gamma_k p_j^T p_k - \delta_k p_j^T p_{k-1} = 0 \qquad \text{für } 0 \leq j \leq k,$$

wobei für j=k die Definition von γ_k, für j=k-1 die von δ_k verwendet wird.
Insbesondere ist damit die lineare Unabhängigkeit von $p_o, p_1, \ldots, p_{l-1}$ gezeigt, es gibt also einen ersten Index $m \leq n$ mit $p_m = 0$, und wir dürfen l=m wählen.
c) wird mittels Induktion nach k bewiesen:
Für k=1 ist die Aussage wegen $p_o = f$ richtig. Sei die Behauptung bereits für ein $1 \leq k < m$ gezeigt; es folgt

$$A p_{k-1} \in [Af, A^2 f, \ldots, A^k f]$$

und weiter

$$p_k = A p_{k-1} - \gamma_{k-1} p_{k-1} - \delta_{k-1} p_{k-2} \in [f, Af, \ldots, A^k f] = U_{k+1} ,$$

also gilt

$$[p_o, p_1, \ldots, p_k] \subseteq U_{k+1} .$$

Nach a)b) sind jedoch $p_o, \ldots, p_k$ linear unabhängig, und man erhält

$$[p_o, p_1, \ldots, p_k] = U_{k+1}$$

aus Dimensionsgründen.
zu d) Aus Teil c) folgt zum einen die lineare Unabhängigkeit von $f, Af, \ldots, A^{m-1} f$, zum anderen mit $A p_k = p_{k+1} + \gamma_k p_k + \delta_k p_{k-1}$, $0 \leq k < m$, und $p_m = 0$:

$$A^m f = A(A^{m-1} f) \in [Ap_0, Ap_1, \ldots, Ap_{m-1}] \subseteq [p_0, p_1, \ldots, p_{m-1}]$$

$$= [f, Af, \ldots, A^{m-1} f] \ .$$

zu e) Zunächst sei daran erinnert, daß wegen $A=A^T$ der $\mathbb{R}^n$ eine Basis aus orthonormalen Eigenvektoren von A besitzt; insbesondere läßt sich f als Linearkombination solcher Vektoren darstellen. Durch Zusammenfassen zum selben Eigenraum gehöriger Vektoren und geeignete Numerierung läßt sich o.B.d.A. die Form (1.6) erreichen. Bezeichnen wir die Anzahl der Komponenten aus (1.6) mit $\tilde{m}$, so ist also $m=\tilde{m}$ zu zeigen.
Nach c) ist die Zahl m identisch mit dem Grad eines Polynom p kleinsten Grades mit

$$p(A)f = \sum_{j=1}^{\tilde{m}} \eta_j p(\lambda_j) z_j = 0 \ , \quad p \not\equiv 0 \ .$$

Für ein solches muß wegen der linearen Unabhängigkeit der z_j und $\eta_j \neq 0$ aber

$$p(\lambda_j) = 0 \ , \quad j=1, \ldots, \tilde{m},$$

gelten, also $m \geq \tilde{m}$, da $\lambda_1, \ldots, \lambda_{\tilde{m}}$ paarweise verschieden sind. Andererseits liefert

$$p(t) \equiv \prod_{j=1}^{\tilde{m}} \left(1 - \frac{t}{\lambda_j}\right)$$

das Gewünschte, somit folgt $m=\tilde{m}$.
zu f) Wegen (1.6) gilt $A^j f \in [z_1, \ldots, z_m]$ für $j=0,1,\ldots$, also

$$U_m \subseteq [z_1, z_2, \ldots, z_m] .$$

Da beide Unterräume die Dimension m besitzen, folgt Gleichheit. □

2. Definition des Algorithmus. Eigenschaften

Im folgenden sei stets $A=A^T$ eine symmetrische, nichtsinguläre $n\times n$-Matrix, $b\in\mathbb{R}^n$ und $\bar{x}$ die Lösung des linearen Gleichungssystems

$$Ax = b \,. \tag{2.1}$$

Wir formulieren zunächst den OD-Algorithmus zur Lösung von (2.1), dabei sei der triviale Fall $x_o=\bar{x}$ ausgeschlossen, es gelte also stets

$$e_o := \bar{x} - x_o \neq 0,\quad r_o := b - Ax_o = A(\bar{x} - x_o) = Ae_o \neq 0.$$

(2.2) OD-Algorithmus (Fridman [14], Fletcher [13]):

Start: Wähle $x_o\in\mathbb{R}^n$ und setze

$$p_{-1} = r_o = b - Ax_o,\quad p_o = Ar_o.$$

Für $k=0,1,2,\ldots$

1) Falls $p_k=0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst setze

2) $\alpha_k = r_k^T p_{k-1}/p_k^T p_k$,

$$x_{k+1} = x_k + \alpha_k p_k\,,\quad r_{k+1} = r_k - \alpha_k Ap_k\,,$$

3)
$$\gamma_k = p_k^T Ap_k/p_k^T p_k\,,\quad \delta_k = \begin{cases} 0 & \text{für } k=0 \\ p_k^T p_k/p_{k-1}^T p_{k-1} & \text{für } k>0 \end{cases},$$

$$p_{k+1} = Ap_k - \gamma_k p_k - \delta_k p_{k-1}\,.$$

In 3) erkennt man den typischen Lanczos-Schritt aus (1.4) wieder, nach Satz(1.5)a)b) gibt es also einen Index m, $1\leqslant m\leqslant n$, mit:

(2.3) a) $p_k \neq 0$ für $k=0,1,\ldots,m-1$, $p_m = 0$;

b) $p_j^T p_k = 0$ für $0 \leqslant j < k \leqslant m$.

Die Größen des OD-Algorithmus besitzen weiter folgende Eigenschaften:

(2.4) Satz: Für $0 \leqslant k < m$ gilt:

a) $Ap_k \in [p_0, p_1, \ldots, p_{k+1}]$;

b) $[p_0, p_1, \ldots, p_k] = [p_0, Ap_0, A^2p_0, \ldots, A^kp_0]$

$= [Ar_0, A^2r_0, \ldots, A^{k+1}r_0] = [A^2e_0, A^3e_0, \ldots, A^{k+2}e_0]$.

Für $0 \leqslant k \leqslant m$ gilt:

c) $r_k = b - Ax_k$;

d) $r_k \in [p_{-1}, p_0, p_1, \ldots, p_k]$;

e) $$r_k^T A^{-1} p_j = \begin{cases} r_k^T p_{k-1} & \text{für } j=k \\ 0 & \text{für } 0 \leqslant j < k \end{cases} ;$$

f) $r_j^T A^{-1} p_k = r_0^T A^{-1} p_k$ für $0 \leqslant j \leqslant k$.

g) $r_m = 0$: $x_m = \bar{x}$ löst $Ax = b$.

Beweis: a) folgt sofort aus $Ap_k = p_{k+1} + \gamma_k p_k + \delta_k p_{k-1}$, b) ist lediglich eine Umschreibung von Satz(1.5)c) unter Verwendung von $f = p_0 = Ar_0 = A^2e_0$. c)-e) werden durch Induktion nach k gezeigt. Dabei ist jeweils die Induktionsannahme, daß die Behauptung für alle Indizes $\leqslant k$ bereits bewiesen sei, und es sei $k+1 \leqslant m$.

zu c) $r_0 = b - Ax_0$ nach Definition.

$r_{k+1} = r_k - \alpha_k Ap_k = b - A(x_k + \alpha_k p_k) = b - Ax_{k+1}$.

zu d) $r_0 = p_{-1} \in [p_{-1}, p_0]$.

$r_{k+1} = r_k - \alpha_k Ap_k \in [p_{-1}, p_0, \ldots, p_{k+1}]$ nach Induktionsannahme und a).

zu e) $k=0$: $r_0^T A^{-1} p_0 = r_0^T p_{-1}$ wegen $p_0 = Ar_0 = Ap_{-1}$.

Für $0 \leqslant j < k$ folgt aus der Induktionsannahme und der Orthogonalität der Suchrichtungen

$$r_{k+1}^T A^{-1} p_j = (r_k - \alpha_k A p_k)^T A^{-1} p_j = r_k^T A^{-1} p_j - \alpha_k p_k^T p_j = 0$$

und für $j=k$

$$r_{k+1}^T A^{-1} p_k = r_k^T A^{-1} p_k - \alpha_k p_k^T p_k = r_k^T p_{k-1} - \alpha_k p_k^T p_k = 0$$

nach Definition von α_k. Mit dem eben Gezeigten ergibt sich weiter

$$r_{k+1}^T A^{-1} p_{k+1} = r_{k+1}^T A^{-1} (A p_k - \gamma_k p_k - \delta_k p_{k-1}) = r_{k+1}^T p_k .$$

zu f) Aus $p_i^T p_k = 0$ für $0 \leq i \leq j-1 < k$ folgt

$$r_j^T A^{-1} p_k = (r_o - \sum_{i=0}^{j-1} \alpha_i A p_i)^T A^{-1} p_k = r_o^T A^{-1} p_k \quad \text{für } 0 \leq j \leq k \leq m.$$

zu g) Wegen a)d) und $p_m = 0$ gilt

$$r_m = r_{m-1} - \alpha_{m-1} A p_{m-1} \in [p_{-1}, p_o, \ldots, p_{m-1}],$$

sowie

$$A r_m \in [A p_{-1}, A p_o, \ldots, A p_{m-1}] \subseteq [p_o, p_1, \ldots, p_{m-1}];$$

mit e) ergibt sich daher

$$r_m^T r_m = r_m^T A^{-1} (A r_m) = 0 , \quad \text{also } r_m = 0. \square$$

Wie gerade gesehen, liegt bei Abbruch des OD-Algorithmus $x_m = \bar{x}$ vor; es ist jedoch nicht möglich, daß man bereits wesentlich früher auf die Lösung von (2.1) stößt:

(2.5) Satz: a) Sei für ein k mit $1 \leq k < m$ $\quad r_{k-1} \neq 0$, $r_k = 0$.

Dann folgt $k = m - 1$.

b) Es gilt: i) $\alpha_o > 0$.

ii) Falls $\alpha_k = 0$ für ein k mit $1 \leq k \leq m - 2$,

so folgt $\alpha_{k+1} \neq 0$.

Beweis: zu a) Aus $0 = r_k = r_{k-1} - \alpha_{k-1} A p_{k-1}$, $r_{k-1} \neq 0$ folgt mit Satz(2.4)d)

$$A p_{k-1} = r_{k-1} / \alpha_{k-1} \in [p_{-1}, p_o, \ldots, p_{k-1}] .$$

Es ergibt sich weiter

$$p_k = Ap_{k-1} - \gamma_{k-1}p_{k-1} - \delta_{k-1}p_{k-2} \in [p_{-1},p_o,\ldots,p_{k-1}]$$

und

$$p_{k+1} = Ap_k - \gamma_k p_k - \delta_k p_{k-1} \in [p_o,p_1,\ldots,p_k] \ .$$

Aus (2.3) und wegen $k<m$ folgt $p_{k+1}=0$, $p_k\neq 0$, also $k=m-1$.

zu b)i) $\alpha_o=r_o^T p_{-1}/p_o^T p_o=r_o^T r_o/p_o^T p_o>0$ wegen unserer Forderung $r_o\neq 0$.

ii) Annahme: $\alpha_k=\alpha_{k+1}=0$ für ein k mit $1\leq k\leq m-2$. O.B.d.A. sei $\alpha_{k-1}\neq 0$.
Es folgt $r_{k+1}=r_k-\alpha_k Ap_k=r_k$ und mit Satz(2.4)d)a)

$$r_{k+1} \in [p_{-1},p_o,\ldots,p_k] \ ,$$

$$Ar_{k+1} \in [Ap_{-1},Ap_o,\ldots,Ap_k] \subseteq [p_o,p_1,\ldots,p_{k+1}] \ ,$$

$$\text{d.h. } Ar_{k+1} = \sum_{j=0}^{k+1} \sigma_j p_j \ .$$

Satz(2.4)e) liefert

$$r_{k+1}^T r_{k+1}=r_{k+1}^T A^{-1}\Big(\sum_{j=0}^{k+1} \sigma_j p_j\Big)=\sigma_{k+1} r_{k+1}^T p_k=\sigma_{k+1}\alpha_{k+1} p_{k+1}^T p_{k+1}=0.$$

Das ergibt $0=r_{k+1}=r_k=r_{k-1}-\alpha_{k-1}Ap_{k-1}$, $r_{k-1}\neq 0$ (wegen $\alpha_{k-1}Ap_{k-1}\neq 0$),
und aus Teil a) dieses Satzes folgt $k=m-1$ im Widerspruch zu $k\leq m-2$. □

Wegen $\bar{x}=x_m=x_o+\sum_{j=0}^{m-1} \alpha_j p_j$ erhält man

$$\bar{x}-\Big(x_o+\sum_{j=0}^{k-1} \xi_j p_j\Big)=\sum_{j=0}^{k-1} (\alpha_j-\xi_j)p_j+\sum_{j=k}^{m-1} \alpha_j p_j$$

für $1\leq k\leq m$, $\xi_o,\xi_1,\ldots,\xi_{k-1}\in\mathbb{R}$; aufgrund der Orthogonalität der p_j
führt dies zu

$$\Big\|\bar{x}-\Big(x_o+\sum_{j=0}^{k-1} \xi_j p_j\Big)\Big\|^2=\sum_{j=0}^{k-1} (\alpha_j-\xi_j)^2 p_j^T p_j+\sum_{j=k}^{m-1} \alpha_j^2 p_j^T p_j$$

und

$$\|\bar{x}-x_k\|^2 = \sum_{j=k}^{m-1} \alpha_j^2 p_j^T p_j = \min_{\xi_0,\ldots,\xi_{k-1} \in \mathbb{R}} \|\bar{x}-(x_0+\sum_{j=0}^{k-1} \xi_j p_j)\|^2,$$

wobei dieses Minimum genau für $\xi_j=\alpha_j$, $0\leq j\leq k-1$, angenommen wird. Damit ist bereits der erste Teil des folgenden Satzes gezeigt, zu dessen Formulierung wir noch einige Abkürzungen einführen:

$$e_k := \bar{x} - x_k ,$$

$$\bar{S}_k := [p_0,p_1,\ldots,p_{k-1}] ,$$

$\bar{P}_k := \mathbb{R}^n \to \bar{S}_k$ sei die orthogonale Projektion auf $\bar{S}_k$.

(2.6) Satz: Für den OD-Algorithmus gilt:

a) $$x_k = \arg\min_{x \in x_0+\bar{S}_k} \|x - \bar{x}\| ,$$

$$\|e_k\| = \min_{x \in x_0+\bar{S}_k} \|x - \bar{x}\| , \quad k=1,2,\ldots,m;$$

b) $$e_k = (I - \bar{P}_k)e_0 ,$$

$$\frac{\|e_k\|}{\|e_0\|} = \sin\sphericalangle(e_0,\bar{S}_k) , \quad k=1,2,\ldots,m;$$

c) $$\|e_k\| > \|e_{k+2}\| , \quad k=0,1,\ldots,m-2 .$$

Beweis: zu b) Aus a) folgt mittels der Transformation $x=x_0+w$

$$\|e_k\| = \min_{w \in \bar{S}_k} \|e_0 - w\| ,$$

also $w_k=\bar{P}_k e_0$ und $e_k=e_0-w_k=(I-\bar{P}_k)e_0$. Hiermit und nach (1.2) erhält man weiter $\|e_k\|/\|e_0\| =\sin\sphericalangle(e_0,\bar{S}_k)$.

c) resultiert aus

$$\|e_k\|^2 = \|e_{k+2}\|^2 + \alpha_k^2 p_k^T p_k + \alpha_{k+1}^2 p_{k+1}^T p_{k+1}$$

und Satz(2.5)b). □

Die Minimierungseigenschaft (2.6)a) gestattet die Herleitung von Fehlerschranken, mit denen wir uns im nächsten Abschnitt befassen werden.

3. Fehlerschranken

Nach (2.4)b) gilt $\bar{S}_k = [A^2 e_o, A^3 e_o, \ldots, A^{k+1} e_o]$, und mit (2.6)a) folgt

$$\|e_k\| = \min_{x \in x_o + [A^2 e_o, \ldots, A^{k+1} e_o]} \|x - \bar{x}\|$$

$$= \min_{\sigma_2, \sigma_3, \ldots, \sigma_{k+1} \in \mathbb{R}} \|(I + \sigma_2 A^2 + \sigma_3 A^3 + \cdots + \sigma_{k+1} A^{k+1}) e_o\| ,$$

also

$$(3.1) \qquad \|e_k\| = \min_{p \in \bar{\Pi}_{k+1}} \|p(A) e_o\| , \quad k = 0, 1, \ldots, m,$$

wobei

$$\bar{\Pi}_k := \{p \in \Pi_k \mid p'(0) = 0\} = \{p(t) \equiv 1 + \sigma_2 t^2 + \sigma_3 t^3 + \cdots + \sigma_k t^k \mid \sigma_j \in \mathbb{R}\}$$

gesetzt wurde.
Nach Satz(1.5)e) ist m gerade die Anzahl der Eigenräume von A, in denen $f = p_o$ von Null verschiedene Komponenten besitzt. Da A nichtsingulär ist erhält man analog zu (1.6) auch für $e_o = A^{-2} p_o$ eine Darstellung der Form

$$(3.2) \qquad e_o = \sum_{j=1}^{m} \rho_j z_j , \quad \rho_1, \ldots, \rho_m \neq 0 ,$$

mit orthonormalen Eigenvektoren $z_1, \ldots, z_m$, die zu paarweise verschiedenen Eigenwerten $\lambda_1, \ldots, \lambda_m$ von A gehören.
Damit können wir nun eine Bedingung angeben, wann (vgl. Satz(2.5)a)) $r_{m-1} = 0$ auftritt:

(3.3) Satz: Es gilt:

a) $x_{m-1} = \bar{x}$ genau dann, wenn $\sum_{j=1}^{m} \frac{1}{\lambda_j} = 0$;

$$\text{b)}\quad \frac{\|e_k\|}{\|e_o\|} \leq \min_{p\in\overline{\Pi}_{k+1}} \max_{1\leq j\leq m} |p(\lambda_j)|, \quad k=0,1,\ldots,m.$$

Beweis: zu a) Nach Wahl der z_j als orthonormale Eigenvektoren ergibt sich für Polynome p

$$p(A)e_o = \sum_{j=1}^{m} \rho_j p(\lambda_j) z_j ,$$

$$\|p(A)e_o\|^2 = \sum_{j=1}^{m} \rho_j^2 p(\lambda_j)^2 ,$$

und wegen (3.1)

$$\|e_k\|^2 = \min_{p\in\overline{\Pi}_{k+1}} \sum_{j=1}^{m} \rho_j^2 p(\lambda_j)^2 . \tag{3.4}$$

Daher gilt:

$x_k=\bar{x}$ genau dann, wenn es in $\overline{\Pi}_{k+1}$ ein Polynom p mit $p(\lambda_j)=0$, $j=1,\ldots,m$, gibt. Gesucht ist daher ein Polynom kleinsten Grades mit $p(0)=1$, $p'(0)=0$, $p(\lambda_j)=0$, $j=1,\ldots,m$.

Offensichtlich ist

$$p(t) \equiv (1 + t \sum_{j=1}^{m} \frac{1}{\lambda_j}) \prod_{j=1}^{m} (1 - \frac{t}{\lambda_j})$$

ein solches, und man erhält:

$$x_{m-1} = \bar{x} \Leftrightarrow \text{grad } p = m \Leftrightarrow \sum_{j=1}^{m} \frac{1}{\lambda_j} = 0 .$$

zu b) Aus (3.4) folgt

$$\|e_k\|^2 \leq \min_{p\in\overline{\Pi}_{k+1}} (\sum_{j=1}^{m} \rho_j^2) \max_{1\leq j\leq m} p(\lambda_j)^2$$

und mit $\|e_o\|^2 = \sum_{j=1}^{m} \rho_j^2$ die Behauptung. □

(3.5) Korollar: Hat A genau l verschiedene Eigenwerte, dann liefert der OD-Algorithmus nach spätestens l Iterationen die exakte Lösung $\bar{x}$ von $Ax = b$.

Um weitere Abschätzungen zu erhalten, benötigen wir zunächst einen Hilfssatz.

(3.6) Lemma: Sei $0 < \alpha < \beta$, $k \in \mathbb{N}$ und

$$\hat{\Pi}_k := \{p \in \Pi_k \mid p \text{ gerade}\}.$$

Es gilt:

$$\min_{p \in \Pi_k} M(p) = \min_{p \in \overline{\Pi}_k} M(p) = \min_{p \in \hat{\Pi}_k} M(p) = \frac{1}{T_{[k/2]}\left(\frac{\beta^2+\alpha^2}{\beta^2-\alpha^2}\right)} .$$

Dabei ist $M(p) := \max_{t \in \mathbb{R}: \alpha \leq |t| \leq \beta} |p(t)|$,

sowie T_j das j-te Tschebyscheffpolynom.

Beweis: Sei $p \in \Pi_k$. Für $\hat{p}(t) \equiv (p(t)+p(-t))/2 \in \hat{\Pi}_k$ gilt

$|\hat{p}(t)| \leq (|p(t)|+|p(-t)|)/2 \leq M(p)$ für $t \in \mathbb{R}: \alpha \leq |t| \leq \beta$, also $M(\hat{p}) \leq M(p)$.
Damit ist gezeigt: Zu jedem $p \in \Pi_k$ gibt es ein $\hat{p} \in \hat{\Pi}_k$ mit $M(\hat{p}) \leq M(p)$.

$$\min_{p \in \hat{\Pi}_k} M(p) \leq \min_{p \in \Pi_k} M(p)$$

und wegen $\hat{\Pi}_k \subseteq \overline{\Pi}_k \subseteq \Pi_k$

$$\min_{p \in \Pi_k} M(p) \leq \min_{p \in \overline{\Pi}_k} M(p) \leq \min_{p \in \hat{\Pi}_k} M(p) ,$$

es gilt somit Gleichheit.

Jedes $p \in \hat{\Pi}_k$ hat die Form

$$p(t) = 1+\sigma_2 t^2+\sigma_4 (t^2)^2+\cdots+\sigma_{2[k/2]} (t^2)^{[k/2]};$$

die Substitution $x=t^2$ führt deshalb zu

$$\min_{p \in \hat{\Pi}_k} M(p) = \min_{p \in \Pi_{[k/2]}} \max_{\alpha^2 \leq x \leq \beta^2} |p(x)| = \frac{1}{T_{[k/2]}\left(\frac{\beta^2+\alpha^2}{\beta^2-\alpha^2}\right)},$$

wobei das letzte Gleichheitszeichen aus Lemma(1.3) resultiert. □

Seien nun die Eigenwerte $\lambda_1,\ldots,\lambda_n$ von A so numeriert, daß

$$0 < |\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_n|, \quad \kappa := \kappa(A) = |\lambda_n|/|\lambda_1|$$

gilt. Setzen wir $\alpha := |\lambda_1|$, $\beta := |\lambda_n|$ (der triviale Fall $\alpha=\beta$ sei ausgeschlossen), so ergibt Satz(3.3)b) in Verbindung mit Lemma(3.6)

$$\frac{\|e_k\|}{\|e_o\|} \leq \min_{p \in \overline{\Pi}_{k+1}} \max_{1 \leq j \leq n} |p(\lambda_j)| \leq \min_{p \in \overline{\Pi}_{k+1}} \max_{t \in \mathbb{R}: \alpha \leq |t| \leq \beta} |p(t)|$$

$$= \frac{1}{T_{[(k+1)/2]}\left(\frac{\beta^2/\alpha^2+1}{\beta^2/\alpha^2-1}\right)}.$$

Verwenden wir noch

(3.7) $$T_j\left(\frac{x+1}{x-1}\right) \geq \frac{1}{2}\left(\frac{x+1}{x-1} + \sqrt{\left(\frac{x+1}{x-1}\right)^2 - 1}\right)^j$$

$$= \frac{1}{2}\left(\frac{\sqrt{x}+1}{\sqrt{x}-1}\right)^j, \quad x > 1,$$

so ist gezeigt:

(3.8) <u>Satz:</u> Für $0 \leq k \leq m$ gilt:

$$\frac{\|e_k\|}{\|e_o\|} \leq \frac{1}{T_{[(k+1)/2]}\left(\frac{\kappa^2+1}{\kappa^2-1}\right)} \leq 2\left(\frac{\kappa-1}{\kappa+1}\right)^{[(k+1)/2]}.$$

(3.9) Korollar: Sei $0 < \varepsilon < 1$. Dann gilt:

$$\frac{\|e_k\|}{\|e_o\|} \leq \varepsilon \qquad \text{für} \quad k \geq \kappa \ln \frac{2}{\varepsilon} .$$

Beweis: Aus

$$\left[\frac{k+1}{2}\right] \geq \frac{k}{2} \geq \frac{\kappa}{2} \ln \frac{2}{\varepsilon} \geq \left(\ln \frac{\kappa+1}{\kappa-1}\right)^{-1} \ln \frac{2}{\varepsilon}$$

folgt

$$2 \left(\frac{\kappa - 1}{\kappa + 1}\right)^{[(k+1)/2]} \leq \varepsilon$$

und mit dem letzten Satz die Behauptung. □

Im letzten Abschnitt wurde bereits $\|e_{k+2}\| < \|e_k\|$ gezeigt. Nun können wir ein schärferes Resultat beweisen:

(3.10) Korollar:

$$\text{a)} \quad \frac{\|e_1\|}{\|e_o\|} \leq \frac{\kappa^2 - 1}{\kappa^2 + 1} ;$$

$$\text{b)} \quad \frac{\|e_{k+2}\|}{\|e_k\|} \leq \frac{\kappa^2 - 1}{\kappa^2 + 1} \quad \text{für } 0 \leq k \leq m-2.$$

Beweis: a) folgt sofort aus Satz(3.8) wegen $T_1(t) \equiv t$.

zu b) Wir fassen $x_k = \hat{x}_o$ als neuen Startwert für den OD-Algorithmus auf. Dann gilt für den nächsten Näherungswert $\hat{x}_1$

$$\|\hat{x}_1 - \bar{x}\| = \min_{x \in \hat{x}_o + [\hat{p}_o]} \|x - \bar{x}\|$$

mit $\hat{p}_o = A\hat{r}_o = Ar_k$. Satz(2.4)d)a) liefert

$$\hat{p}_o = Ar_k \in [p_o, Ap_o Ap_1, \ldots, Ap_k] \subseteq [p_o, p_1, \ldots, p_{k+1}],$$

und mit $\hat{x}_o = x_k = x_o + \alpha_o p_o + \cdots + \alpha_{k-1} p_{k-1}$ folgt

$$\hat{x}_o + [\hat{p}_o] \subseteq x_o + [p_o, p_1, \ldots, p_{k+1}] .$$

Man erhält daher

$$\|\hat{x}_1 - \bar{x}\| \geq \min_{x \in x_o + [p_o,\ldots,p_{k+1}]} \|x - \bar{x}\| = \|x_{k+2} - \bar{x}\| ,$$

und mit Teil a) dieses Korollars

$$\frac{\|x_{k+2} - \bar{x}\|}{\|x_k - \bar{x}\|} \leq \frac{\|\hat{x}_1 - \bar{x}\|}{\|\hat{x}_o - \bar{x}\|} \leq \frac{\kappa^2 - 1}{\kappa^2 + 1} \quad . \square$$

Zur Herleitung unserer bisherigen Schranken wurde als einzige Information über das Spektrum von A die Kenntnis der Konditionszahl ausgenutzt; es ist daher nicht verwunderlich, daß diese Abschätzungen in den meisten Fällen viel zu pessimistisch sind. Wesentlich bessere Resultate lassen sich erzielen, wenn spezielle Eigenwertverteilungen vorliegen, so zum Beispiel wenn A einige wenige, betragsmäßig große, negative Eigenwerte besitzt, und die restlichen Eigenwerte in einem kleinem Intervall der positiven reellen Achse liegen.

(3.11) Satz: Sei $0 < \alpha < \beta$. A habe Eigenwerte λ_j mit

$$\lambda_1,\ldots,\lambda_l < 0; \quad \lambda_{l+1},\ldots,\lambda_n \in [\alpha,\beta] \quad .$$

Dann gilt für $k=0,1,2,\ldots$

$$\frac{\|e_{l+k}\|}{\|e_o\|} \leq 2 \Big(\prod_{j=1}^{l} (1 - \frac{\beta}{\lambda_j}) \Big)(1 - \beta \sum_{j=1}^{l} \frac{1}{\lambda_j} + \sqrt{\frac{\beta}{\alpha}}\, k)\left(\frac{\sqrt{\frac{\beta}{\alpha}} - 1}{\sqrt{\frac{\beta}{\alpha}} + 1}\right)^k$$

Beweis: Nach Satz(3.3)b) liefert jedes Polynom $p \in \bar{\Pi}_{l+k+1}$ eine Abschätzung folgender Art:

$$(3.12) \qquad \frac{\|e_{l+k}\|}{\|e_o\|} \leq \max_{1 \leq j \leq n} |p(\lambda_j)|, \quad k=0,1,\ldots \quad .$$

Wir betrachten speziell $p(t) \equiv r(t)q(t)$ mit

$$r(t) \equiv (1 + \mu t) \prod_{j=1}^{l} (1 - \frac{t}{\lambda_j}) \quad , \quad q(t) \equiv \frac{T_k(\frac{-2t + \beta + \alpha}{\beta - \alpha})}{T_k(z)} \quad ,$$

wobei $z := \frac{\beta+\alpha}{\beta-\alpha}$ gesetzt wurde.

Es ist grad $p \leq l+k+1$, $p(0)=1$ und

$$p'(0) = \mu - \sum_{j=1}^{l} \frac{1}{\lambda_j} - \frac{2}{\beta-\alpha} \frac{T_k'(z)}{T_k(z)} ,$$

also $p \in \overline{\Pi}_{l+k+1}$, wenn wir

$$\mu = \sum_{j=1}^{l} \frac{1}{\lambda_j} + \frac{2}{\beta-\alpha} \frac{T_k'(z)}{T_k(z)}$$

wählen. Nach Konstruktion von p gilt $p(\lambda_j)=0$ für $j=1,\ldots,l$, und mit (3.12) ergibt sich

$$(3.13) \qquad \frac{\|e_{l+k}\|}{\|e_o\|} \leq \max_{l+1 \leq j \leq n} |p(\lambda_j)| \leq \max_{\alpha \leq t \leq \beta} |p(t)|$$

$$\leq \max_{\alpha \leq t \leq \beta} |r(t)| \max_{\alpha \leq t \leq \beta} |q(t)| .$$

Nach Wahl von q und wegen (3.7) gilt

$$(3.14) \qquad \max_{\alpha \leq t \leq \beta} |q(t)| = \frac{1}{T_k(\frac{\beta+\alpha}{\beta-\alpha})} \leq 2 \left(\frac{\sqrt{\frac{\beta}{\alpha}} - 1}{\sqrt{\frac{\beta}{\alpha}} + 1} \right)^k .$$

Unter Verwendung von

$$\frac{T_k'(z)}{T_k(z)} = \frac{k}{\sqrt{z^2-1}} \sqrt{1 - \frac{1}{(T_k(z))^2}} \leq \frac{k}{\sqrt{z^2-1}}$$

erhalten wir

$$(3.15) \qquad \max_{\alpha \leq t \leq \beta} |r(t)| \leq (1 + |\mu|\beta) \prod_{j=1}^{l} (1 - \frac{\beta}{\lambda_j})$$

$$\leq (1 - \beta \sum_{j=1}^{l} \frac{1}{\lambda_j} + \sqrt{\frac{\beta}{\alpha}}\, k) \prod_{j=1}^{l} (1 - \frac{\beta}{\lambda_j}) .$$

(3.13),(3.14),(3.15) ergeben nun die Behauptung. □

Der Spezialfall l=0 liefert das folgende

(3.16) Korollar: Sei A positiv definit. Dann gilt:

$$\frac{\|e_k\|}{\|e_o\|} \leq 2(1 + \kappa \cdot k)\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k , \quad k=0,1,\ldots .$$

Axelsson [1] zeigte eine Abschätzung für das CG-Verfahren unter der Annahme, daß die Eigenwerte von A in zwei disjunkten, gleichlangen Intervallen der positiven reellen Achse liegen. Ein ähnliches Resultat erhält man für den OD-Algorithmus:

(3.17) Satz: Sei $0 < \varepsilon < 1$. Sei $0 < \alpha < \beta < \gamma < \delta$ und $\beta^2-\alpha^2=\delta^2-\gamma^2$.

Für die Eigenwerte von A gelte:

$$\lambda_j \in M := [-\delta,-\gamma]\cup[-\beta,-\alpha]\cup[\alpha,\beta]\cup[\gamma,\delta] , \quad j=1,\ldots,n.$$

Dann gilt:

$$\text{i)} \quad \frac{\|e_k\|}{\|e_o\|} \leq \frac{1}{T_{[(k+1)/4]}\left(\frac{(\frac{\beta\gamma}{\alpha\delta})^2 + 1}{(\frac{\beta\gamma}{\alpha\delta})^2 - 1}\right)} \leq 2\left(\frac{\frac{\beta\gamma}{\alpha\delta} - 1}{\frac{\beta\gamma}{\alpha\delta} + 1}\right)^{[(k+1)/4]} ;$$

$$\text{ii)} \quad \frac{\|e_k\|}{\|e_o\|} \leq \varepsilon \quad \text{für} \quad k \geq 2\,\frac{\beta\gamma}{\alpha\delta}\ln\frac{2}{\varepsilon} + 2 .$$

Beweis: Wie im Beweis von Lemma(3.6) zeigt man

$$\min_{p \in \overline{\Pi}_k} \max_{t \in M} |p(t)| = \min_{p \in \hat{\Pi}_k} \max_{t \in M} |p(t)| = \min_{p \in \Pi_{[k/2]}} \max_{x \in [\alpha^2,\beta^2]\cup[\gamma^2,\delta^2]} |p(x)|$$

Mit $\tilde{k} := [\frac{1}{2}[\frac{k}{2}]] = [\frac{k}{4}]$ gilt $2\tilde{k} \leq [\frac{k}{2}]$, und es folgt

(3.18)
$$\min_{p \in \Pi_{[k/2]}} \max_{x \in [\alpha^2,\beta^2]\cup[\gamma^2,\delta^2]} |p(x)| \leq \min_{p \in \Pi_{2\tilde{k}}} \max_{x \in [\alpha^2,\beta^2]\cup[\gamma^2,\delta^2]} |p(x)| .$$

Die rechte Seite von (3.18) ist aber nach einem Resultat von Lebedev [19] gleich

$$(T_{\tilde{k}}\left(\frac{(\frac{\beta\gamma}{\alpha\delta})^2 + 1}{(\frac{\beta\gamma}{\alpha\delta})^2 - 1}\right))^{-1} \quad .$$

Zusammen mit Satz(3.3)b) und der Abschätzung(3.7) ergibt sich nun Behauptung i) und daraus ii). □

In den nächsten Abschnitten werden Zusammenhänge zwischen dem OD-Algorithmus und dem SYMMLQ- bzw. MCR-Algorithmus betrachtet. Dabei werden weitgehendst die Bezeichnungen von Paige und Saunders [26] bzw. Chandra [6] verwendet. Um die Größen des OD- und MCR-Algorithmus unterscheiden zu können, werden diese, falls nötig, mit oben stehendem OD bzw. MCR gekennzeichnet. Außerdem sei stets vorausgesetzt, daß alle drei Verfahren mit demselben Anfangswert $x_o \neq \bar{x}$ gestartet werden, also für das Startresiduum

$$r_o = r_o^{MCR} = r_o^{OD} \neq 0$$

gilt.

4. Zusammenhang mit dem SYMMLQ-Algorithmus

Wir stellen zunächst die Überlegungen, die Paige und Saunders [26] zu SYMMLQ führten, kurz dar.
Man versucht für $\bar{x}$ Näherungen x_k zu berechnen, die der Galerkin-Bedingung

$$(4.1) \qquad r_k = b - Ax_k \in S_k^{\perp} , \; x_k \in x_o + S_k$$

genügen. Dabei liegen die Krylovunterräume

$$S_k := [r_o, Ar_o, \ldots, A^{k-1} r_o] , \quad k=1,2,\ldots,m,$$

(m wie in (2.3)a); vgl. auch Satz(1.5)) zugrunde, für welche Orthonormalbasen mittels folgender Implementierung des Lanczos-Algorithmus berechnet werden:

$$\beta_1 = \|r_o\| , \; v_1 = r_o/\beta_1 , \; v_o = 0 ;$$

$$v'_{k+1} = Av_k - \alpha_k v_k - \beta_k v_{k-1}, \; \alpha_k = v_k^T A v_k, \; \beta_{k+1} = \|v'_{k+1}\|, \; k=1,2,\ldots,m;$$

$$v_{k+1} = v'_{k+1}/\beta_{k+1}, \; k=1,2,\ldots,m-1 .$$

Mit $V_k := (v_1, \ldots, v_k)$ gilt:

(4.2)

a) $\beta_{m+1} = 0$;

b) $AV_m = V_m T_m$;

und für $k=1,2,\ldots,m$:

c) $[v_1, v_2, \ldots, v_k] = S_k$;

d) $V_k^T V_k = I_k$;

e)

$$V_k^T A V_k = T_k = \begin{bmatrix} \alpha_1 & \beta_2 & & \sigma \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & & \beta_k \\ \sigma & & \beta_k & \alpha_k \end{bmatrix} .$$

Über die Transformation $x_k = x_o + V_k y_k$ ist damit (4.1) äquivalent zu

$$T_k y_k = \beta_1 e_1 \,, \tag{4.3}$$

wobei $e_1 = (1,0,0,\ldots,0)^T \in \mathbb{R}^k$.

Ist A positiv definit, so auch T_k, $k=1,\ldots,m$, und es existiert jeweils die Cholesky-Zerlegung $T_k = \mathcal{L}_k \mathcal{L}_k^T$. Die zu den y_k gehörigen

$$x_k^c = x_k = x_o + (V_k \mathcal{L}_k^{-T})\, \mathcal{L}_k^{-1} \beta_1 e_1$$

sind gerade die Näherungen, welche das CG-Verfahren für $\bar{x}$ liefert, und die Spalten von V_k bzw. diejenigen von $V_k \mathcal{L}_k^{-T}$ stellen, bis auf skalare Faktoren, die Residuen bzw. die Suchrichtungen des CG-Verfahrens dar.
Ist A indefinit, so können einzelne T_k singulär werden; man kann nicht mehr garantieren, daß die Cholesky-Zerlegung existiert, und numerisch können Instabilitäten auftreten.
In ihrem SYMMLQ-Algorithmus umgehen Paige und Saunders diese Schwierigkeiten, indem sie anstelle von (4.3) ein Gleichungssystem mit einer modifizierten, aber stets nichtsingulären Koeffizientenmatrix lösen, und so Näherungswerte x_k^L erhalten, aus denen sich (falls T_k nichtsingulär) x_k^c berechnen läßt. Dabei wird ausgenutzt, daß sich T_k mittels Givensreflexionen $Q_{j,j+1}$ (mit den wesentlichen Elementen $q_{jj} = -q_{j+1,j+1} = c_j$, $q_{j,j+1} = q_{j+1,j} = s_j$) auf stabile Weise faktorisieren läßt:

$$T_k Q_{1,2} \cdots Q_{k-1,k} = T_k Q_k^T = \bar{L}_k = \begin{bmatrix} \gamma_1 & & & 0 \\ \delta_2 & \gamma_2 & & \\ \varepsilon_3 & \delta_3 & \gamma_3 & \\ & \ddots & \ddots & \ddots \\ 0 & & & \gamma_{k-1} \\ & & \varepsilon_k & \delta_k \quad \bar{\gamma}_k \end{bmatrix} . \tag{4.4}$$

Mit

$$(4.5) \qquad \gamma_k = (\bar{\gamma}_k^2 + \beta_{k+1}^2)^{1/2} , \quad c_k = \bar{\gamma}_k/\gamma_k , \quad s_k = \beta_{k+1}/\gamma_k$$

ergibt sich dann im nächsten Schritt aus (4.4) leicht die Zerlegung von T_{k+1}, und die Matrix L_k, die man erhält, wenn in $\bar{L}_k$ $\bar{\gamma}_k$ durch γ_k ersetzt wird, tritt gerade als Hauptuntermatrix von $\bar{L}_{k+1}$ auf. Setzen wir

$$\bar{W}_k = (w_1,\ldots,w_{k-1},\bar{w}_k) = V_k Q_k^T , \quad W_k = (w_1,\ldots,w_{k-1},w_k) ,$$

$$\bar{z}_k = (\xi_1,\ldots,\xi_{k-1},\bar{\xi}_k)^T = Q_k y_k , \quad z_k = (\xi_1,\ldots,\xi_{k-1},\xi_k)^T ,$$

so ist (4.3) äquivalent zu

$$(4.6) \qquad \bar{L}_k \bar{z}_k = \beta_1 e_1 .$$

Statt dieses wird jedoch stets das Gleichungssystem

$$L_k z_k = \beta_1 e_1$$

gelöst (wegen $\gamma_i > 0$, $i=1,\ldots,m$, ist jedes L_k nichtsingulär), und es ergibt sich

$$x_k^L = x_o + W_k z_k .$$

Falls T_k nichtsingulär ($\Leftrightarrow \bar{L}_k$ nichtsingulär $\Leftrightarrow$ $\bar{\gamma}_k \neq 0$), so liefert (4.6) $\bar{\xi}_k$ und damit

$$x_k^c = x_{k-1}^L + \bar{\xi}_k \bar{w}_k .$$

Im letzten Schritt folgt wegen $\beta_{m+1}=0$

$$W_m = \bar{W}_m , \quad L_m = \bar{L}_m ,$$

und man überzeugt sich leicht, daß $x_m^c = x_m^L = \bar{x}$ gilt.

Fletcher [13] zeigte nun, daß SYMMLQ- und OD-Algorithmus in engem Zusammenhang stehen, denn es gilt

$$(4.7) \qquad x_k^L = x_k^{OD} , \quad k=0,1,2,\ldots,m .$$

Wir werden dieses Resultat hier auf etwas andere Weise als Fletcher gewinnen:

(4.8) Satz: Für $k=0,1,\ldots,m-1$ gilt:

$$p_k^{OD} = \|p_k^{OD}\| w_{k+1} \,, \quad \|p_k^{OD}\| / \|p_{k-1}^{OD}\| = s_k \gamma_{k+1} \,,$$

(wobei $s_o := 1$ gesetzt wurde).

Beweis: Wegen $W_m = \overline{W}_m = V_m Q_m^T$, Q_m orthogonal, folgt aus (4.2)d)

$$W_m^T W_m = I_m \,, \text{ d.h. } w_1,\ldots,w_m \text{ sind orthonormal,}$$

und aus (4.2)b)

$$(4.9) \qquad AW_m = V_m T_m Q_m^T = W_m \tilde{T}_m \quad \text{mit} \quad \tilde{T}_m := Q_m T_m Q_m^T \,.$$

Mit $L_m = \overline{L}_m$ und (4.4) ergibt sich weiter

$$\tilde{T}_m = Q_m L_m = Q_{m-1,m} \cdots Q_{1,2} L_m \,,$$

und man erkennt, daß $\tilde{T}_m$ eine untere Hessenbergmatrix mit den Elementen

$$\sigma_j := s_{j-1}\gamma_j \,, \quad j=2,\ldots,m,$$

überhalb der Diagonale ist. Nun ist aber $\tilde{T}_m$ symmetrisch und daher tridiagonal, (4.9) liefert somit die Rekursionsformel

$$(4.10) \qquad \sigma_{k+1} w_{k+1} = Aw_k - \rho_k w_k - \sigma_k w_{k-1} \,, \quad k=1,\ldots,m-1,$$

wobei $w_o := 0$, $\sigma_1 := \gamma_1$. Außerdem gilt $\rho_k = w_k^T A w_k$, wie aus (4.10) und der Orthonormalität der w_k folgt.

Nun können wir die Aussage des Satzes durch Induktion nach k beweisen.

Für k=0 erhalten wir mit $\overline{\gamma}_1 = \alpha_1$ und (4.5)

$$w_1 = c_1 v_1 + s_1 v_2 = (\alpha_1 v_1 + \beta_2 v_2)/\gamma_1 = Av_1/\gamma_1 \,, \quad \gamma_1 = \|Av_1\| \,,$$

und mit $\beta_1 = \|r_o\|$, $v_1 = r_o/\beta_1$, $p_{-1} = r_o$, $s_o = 1$

$$p_o = Ar_o = \|p_o\| w_1 \,, \quad \|p_o\| / \|p_{-1}\| = \gamma_1 = s_o \gamma_1 \,.$$

Sei die Behauptung bereits gezeigt für alle Indizes $\leq k-1$ und sei $1 \leq k < m$. Dann folgt

$$\rho_k = w_k^T A w_k = p_{k-1}^T A p_{k-1} / p_{k-1}^T p_{k-1} = \gamma_{k-1}^{OD}$$

und

$$\sigma_k w_{k-1} = (\|p_{k-1}\| / \|p_{k-2}\|^2) p_{k-2} = \delta_{k-1}^{OD} p_{k-2} / \|p_{k-1}\| \quad \text{falls } k \geq 2,$$

$$\sigma_k w_{k-1} = 0 = \delta_{k-1}^{OD} p_{k-2} / \|p_{k-1}\| \quad \text{falls } k=1.$$

Damit erhält man aus (4.10)

$$\sigma_{k+1} w_{k+1} = (A p_{k-1} - \gamma_{k-1}^{OD} p_{k-1} - \delta_{k-1}^{OD} p_{k-2}) / \|p_{k-1}\| = p_k / \|p_{k-1}\|$$

und unter Beachtung von $\|w_{k+1}\|=1$, $\sigma_{k+1} = s_k \gamma_{k+1} > 0$

$$p_k = \|p_k\| w_{k+1}, \quad \|p_k\| / \|p_{k-1}\| = s_k \gamma_{k+1} . \square$$

Aus $\bar{x} = x_m^L = x_o + \xi_1 w_1 + \cdots + \xi_m w_m$

$= x_m^{OD} = x_o + \alpha_o^{OD} p_o^{OD} + \cdots + \alpha_{m-1}^{OD} p_{m-1}^{OD}$ folgt mit diesem Satz

$$\alpha_k^{OD} = \xi_{k+1} / \|p_k^{OD}\|, \quad k=0,1,\ldots,m-1$$

und (4.7) ist gezeigt.

Schließlich notieren wir noch folgende Charakterisierung der x_k^{OD} bzw. x_k^c, die sich aus Satz(2.6)a) bzw. mit

$$A S_k = \bar{S}_k := [A r_o, A^2 r_o, \ldots, A^k r_o]$$

aus (4.1) ergibt:

(4.11) Für $k = 1,2,\ldots,m$ gilt:

a) x_k^{OD} ist eindeutig bestimmt durch die Forderungen

$$x_k^{OD} \in x_o + \bar{S}_k, \quad \bar{x} - x_k^{OD} \perp \bar{S}_k ;$$

b) falls T_k nichtsingulär, ist x_k^c eindeutig bestimmt durch die Forderungen

$$x_k^c \in x_o + S_k, \quad \bar{x} - x_k^c \perp \bar{S}_k .$$

5. Verbindungen zum MCR-Algorithmus

Einen weiteren dem CG-Verfahren verwandten Algorithmus zur Lösung von $Ax=b$ erhält man dadurch, daß in jedem Schritt Näherungen x_k berechnet werden, die das Residuum in folgendem Sinne minimieren:

$$(5.1)\quad \|b - Ax_k\| = \min_{x \in x_o + [r_o, Ar_o, \ldots, A^{k-1}r_o]} \|b - Ax\| .$$

Für positiv definite Matrizen A führt dies auf den "Conjugate Residual"-Algorithmus. Die Residuen $r_k=b-Ax_k$ sind nämlich parallel zu den CG-Suchrichtungen, wie Hestenes und Stiefel [16] zeigten, und somit A-konjugiert. Auf einfache Weise erhält man den CR-Algorithmus als Spezialfall $\mu=1$ der von Rutishauser [11] betrachteten Verallgemeinerung des CG-Verfahrens, die darin besteht, daß die Skalarprodukte u^Tv durch $u^TA^\mu v$ ersetzt werden. Die CR-Suchrichtungen d_k ergeben sich daher, startend mit $d_o=r_o$, aus der Rekursion

$$(5.2)\qquad d_{k+1} = r_{k+1} + \beta_k d_k ,$$

wobei für β_k die Wahl zwischen

$$(5.3)\qquad a)\ \beta_k = r_{k+1}^T A r_{k+1} / r_k^T A r_k$$

oder

$$b)\ \beta_k = - r_{k+1}^T A^2 d_k / d_k^T A^2 d_k$$

besteht, während die Schrittweiten mittels

$$\alpha_k = r_k^T A r_k / d_k^T A^2 d_k$$

berechnet werden (siehe etwa Reid [28]).
Für indefinite Matrizen ist der CR-Algorithmus jedoch ungeeignet, denn es kann $\alpha_k=0$, $r_k\neq 0$ auftreten, was bei Verwendung von (5.3)a) sofort zum Abbruch, bei (5.3)b) zu $d_{k+1}=0$ (siehe Chandra [6]) und damit zum Abbruch im nächsten Iterationsschritt führt. Fletcher [13] umging diese Schwierigkeit, indem er ausnutzte, daß sich die (ge-

eignet umnormierten) Suchrichtungen auch durch eine Lanczos-Rekursion erzeugen lassen, und erhielt so ein für indefinite Matrizen stabiles Verfahren, welches wir in Anlehnung an die Bezeichnungen Chandras MCR-Algorithmus nennen wollen:

(5.4) MCR-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$p_o = r_o = b - Ax_o \ , \ p_{-1} = 0 \ .$$

Für k=0,1,2,...

1) Falls $Ap_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst setze

2) $\alpha_k = r_k^T Ap_k / p_k^T A^2 p_k$,

$x_{k+1} = x_k + \alpha_k p_k$, $r_{k+1} = r_k - \alpha_k Ap_k$,

3)
$$\gamma_k = p_k^T A^3 p_k / p_k^T A^2 p_k \ , \quad \delta_k = \begin{cases} 0 & \text{für } k=0 \\ p_k^T A^2 p_k / p_{k-1}^T A^2 p_{k-1} & \text{für } k>0 \end{cases} ,$$

$$p_{k+1} = Ap_k - \gamma_k p_k - \delta_k p_{k-1} \ ,$$

$$Ap_{k+1} = A(Ap_k) - \gamma_k Ap_k - \delta_k Ap_{k-1} \ .$$

In der angegebenen Form benötigt der MCR-Algorithmus pro Iteration eine Matrixmultiplikation $A(Ap_k)$ und zirka 9n Multiplikationen.

Nun zeigt sich, daß ähnlich wie (5.2)

$$-\alpha_k p_{k+1} = r_{k+1} + \beta_k p_k \ , \ \beta_k = -r_{k+1}^T A^2 p_k / p_k^T A^2 p_k$$

gilt. Chandra [6] schlug eine Version vor, bei der falls $|\alpha_k| \leq \varepsilon$ p_{k+1} und Ap_{k+1} wie angegeben gebildet werden, andernfalls wird mit der umnormierten Suchrichtung

$$\tilde{p}_{k+1} = r_{k+1} + \beta_k p_k, \quad A\tilde{p}_{k+1} = Ar_{k+1} + \beta_k Ap_k$$

weitergerechnet, wodurch sich der Aufwand auf eine Matrixmultiplikation Ar_{k+1} und ungefähr 7n Multiplikationen reduziert. Diese ver-

kürzte Rekursion ist jedoch im allgemeinen nicht in jedem Schritt möglich, denn wie beim OD-Algorithmus (vgl. Satz(2.5)b)) kann man lediglich garantieren, daß auf $\alpha_k=0$ $\alpha_{k+1}\neq 0$ folgt, und im nächsten Abschnitt werden wir Beispiele angeben, bei denen stets $\alpha_{2k}=0$ gilt. Außerdem steht man vor dem Problem, ε geeignet zu wählen.

In [6] finden sich auch Abschätzungen, die aus der Minimierungseigenschaft (5.1) resultieren, und es zeigt sich, daß man vergleichbare Fehlerschranken wie beim OD-Algorithmus erhält, z.B.(vgl. Satz(3.8))

$$\frac{\|b - Ax_k\|}{\|b - Ax_o\|} \leq \frac{1}{T_{[k/2]}\left(\frac{\kappa^2 + 1}{\kappa^2 - 1}\right)} \leq 2\left(\frac{\kappa - 1}{\kappa + 1}\right)^{[k/2]}.$$

Man sieht ferner, daß der MCR-Algorithmus nach genau m Iterationen (mit der beim OD-Algorithmus definierten Zahl m) mit der exakten Lösung abbricht.

Der für uns wesentliche Zusammenhang zwischen OD- und MCR-Algorithmus, nämlich daß die Suchrichtungen der beiden Verfahren durch $p_k^{OD}=Ap_k^{MCR}$ gekoppelt sind, wurde von Fletcher [13] gezeigt und wird uns zur STOD-Version des OD-Algorithmus führen. Diese und einige weitere Relationen stellen wir zusammen in folgendem

(5.5) <u>Satz:</u> Für k = 0,1,...,m-1 gilt:

a) $p_k^{OD} = Ap_k^{MCR}$;

b) $[p_o^{MCR}, p_1^{MCR}, \ldots, p_k^{MCR}] = [r_o, Ar_o, \ldots, A^k r_o]$;

c) $r_k^{MCR} \in [p_o^{MCR}, \ldots, p_k^{MCR}]$;

d) $(r_j^{MCR})^T Ap_k^{MCR} = r_o^T Ap_k^{MCR}$ für $0 \leq j \leq k$;

e) $\alpha_k^{MCR} = r_o^T Ap_k^{MCR} / (p_k^{MCR})^T A^2 p_k^{MCR}$;

f) $(r_k^{OD})^T r_j^{MCR} = 0$ für $0 \leq j < k$.

Beweis: zu a) Induktion nach k:

Für k=0 gilt $p_o^{OD}=Ar_o=Ap_o^{MCR}$ wegen unserer Generalvoraussetzung $r_o=r_o^{OD}=r_o^{MCR}$. Sei sie Behauptung bereits für alle Indizes $\leq k$ bewiesen, und sei $k+1<m$; aus den Rekursionsformeln für p_{k+1}^{OD} und p_{k+1}^{MCR} folgt dann unmittelbar $p_{k+1}^{OD}=Ap_{k+1}^{MCR}$.

b) ergibt sich nun aus a) und Satz(2.4)b).

zu c)d)e) Aus

$$r_k^{MCR} = r_o - \alpha_o^{MCR}Ap_o^{MCR} - \cdots - \alpha_{k-1}^{MCR}Ap_{k-1}^{MCR}$$

erhält man zum einen c) wegen $p_o^{MCR}=r_o$ und $Ap_j^{MCR}=p_{j+1}^{MCR}+\gamma_j p_j^{MCR}+\delta_j p_{j-1}^{MCR}$, zum anderen d) wegen

$$(5.6) \qquad (p_j^{MCR})^T A^2 p_k^{MCR} = (p_j^{OD})^T p_k^{OD} = 0 \quad , \quad j \neq k \; ;$$

e) folgt aus d) und der Definition von α_k^{MCR}.

zu f) Induktion nach k:

Für k=0 ist nichts zu zeigen.

Gelte nun bereits $(r_k^{OD})^T r_j^{MCR}=0$ für $0\leq j<k$, und sei $k+1<m$. Wegen

$$(5.7) \qquad (r_{k+1}^{OD})^T r_j^{MCR} = (r_k^{OD})^T r_j^{MCR} - \alpha_k^{OD}(p_k^{MCR})^T A^2 r_j^{MCR}$$

folgt aus der Induktionsannahme, aus c) und (5.6) sofort

$$(r_{k+1}^{OD})^T r_j^{MCR} = 0 \qquad \text{für} \quad 0 \leq j < k \; .$$

Bleibt also noch der Fall j=k zu behandeln. Wegen c) gilt $r_k^{MCR}=\sum_{i=0}^{k}\sigma_i p_i^{MCR}$, und es ergibt sich mit Satz(2.4)e)

$$(r_k^{OD})^T r_k^{MCR} = \sum_{i=0}^{k}\sigma_i (r_k^{OD})^T A^{-1} p_i^{OD} = \sigma_k (r_k^{OD})^T p_{k-1}^{OD} \; ,$$

bzw.

$$(p_k^{MCR})^T A^2 r_k^{MCR} = \sigma_k (p_k^{MCR})^T A^2 p_k^{MCR} = \sigma_k (p_k^{OD})^T p_k^{OD}$$

nach (5.6). Aus (5.7) und der Definition von α_k^{OD} folgt jetzt

$$(r_{k+1}^{OD})^T r_k^{MCR} = \sigma_k ((r_k^{OD})^T p_{k-1}^{OD} - \alpha_k^{OD}(p_k^{OD})^T p_k^{OD}) = 0 \; . \; \square$$

6. Einiges über die Möglichkeit $\alpha_k^{OD}=0$ bzw. $\alpha_k^{MCR}=0$

Wie bereits erwähnt, kann man für den OD- bzw. MCR-Algorithmus nicht ausschließen, daß

$$\alpha_k^{OD} = 0 \qquad \text{bzw.} \qquad \alpha_k^{MCR} = 0$$

vorkommt, und lediglich garantieren, daß die Verfahren in jedem zweiten Iterationsschritt eine verbesserte Näherung liefern. Wir zeigen jetzt, wie sich dieses "auf der Stelle treten" im SYMMLQ-Algorithmus von Paige und Saunders äußert. Zunächst schicken wir die beiden Relationen

(6.1) a) $\alpha_k^{OD}(p_k^{OD})^T p_k^{OD} = r_o^T A^{-1} p_k^{OD} = r_o^T p_k^{MCR}$,

b) $\alpha_k^{MCR}(p_k^{MCR})^T A^2 p_k^{MCR} = r_o^T A p_k^{MCR} = r_o^T p_k^{OD}$

voraus, die aus den Sätzen(2.4)e)f),(5.5)a)e) folgen.

(6.2) Satz: Sei $0 \leqslant k < m$.

a) Es kann nicht gleichzeitig

$\alpha_k^{OD} = 0$ und $\alpha_k^{MCR} = 0$ gelten.

b) Es sind äquivalent:

(1) $\alpha_k^{MCR} = 0$

(2) $r_o^T p_k^{OD} = 0$

(3) T_{k+1} ist singulär.

c) Es sind äquivalent:

(1) $\alpha_k^{OD} = 0$

(2) $r_o^T p_k^{MCR} = 0$

(3) T_{k+1} ist nichtsingulär und $x_{k+1}^c = x_k^{OD}$.

Beweis: zu a) Annahme: $\alpha_k^{OD}=\alpha_k^{MCR}=0$ für ein k mit $0\leq k<m$. Falls $k\geq 2$, so folgt mit Satz(5.5)b)

$$[A^2r_o,\ldots,A^kr_o] = [A^2p_o^{MCR},A^2p_1^{MCR},\ldots,A^2p_{k-2}^{MCR}]$$

und wegen der A^2-Konjugiertheit der p_j^{MCR}

$$(6.3)\qquad (A^jr_o)^Tp_k^{MCR} = 0 \qquad \text{für } j=2,\ldots,k.$$

Aus unserer Annahme ergibt sich mit (6.1)

$$r_o^Tp_k^{MCR} = (Ar_o)^Tp_k^{MCR} = 0,$$

d.h. (6.3) gilt sogar für $j=0,1,\ldots,k$; man erhält also

$$p_k^{MCR} \in [r_o,Ar_o,\ldots,A^kr_o]^{\perp} = [p_o^{MCR},p_1^{MCR},\ldots,p_k^{MCR}]^{\perp} .$$

Es folgt $p_k^{MCR}=0$ und $p_k^{OD}=Ap_k^{MCR}=0$ im Widerspruch zu $k<m$.

zu b) Wegen (6.1)b) bleibt nur die Äquivalenz von (2) und (3) zu zeigen. Nach (4.2)c) und Satz(2.4)b) gilt:

$$[v_1,v_2,\ldots,v_{k+1}] = [r_o,Ar_o,\ldots,A^kr_o] = [r_o,p_o^{OD},p_1^{OD},\ldots,p_{k-1}^{OD}] ,$$

$$[Av_1,Av_2,\ldots,Av_{k+1}] = [p_o^{OD},p_1^{OD},\ldots,p_k^{OD}] .$$

Beachtet man die Definition von T_{k+1}, nämlich

$$T_{k+1} = V_{k+1}^TAV_{k+1} , \quad V_{k+1} = (v_1,v_2,\ldots,v_{k+1}) ,$$

so erkennt man sofort:

$$T_{k+1} \text{ singulär} \iff \text{Es gibt ein } y\in\mathbb{R}^{k+1} , y\neq 0, \text{ mit } V_{k+1}^TAV_{k+1}\,y = 0$$

$$\iff \text{Es gibt ein } x = \sum_{j=0}^{k}\sigma_jp_j^{OD} , \quad x\neq 0, \text{ mit } x\in[r_o,p_o^{OD},\ldots,p_{k-1}^{OD}]^{\perp}.$$

Wegen der Orthogonalität der p_j^{OD} erhalten wir daraus nun die gewünschte Äquivalenz.

zu c) Die Äquivalenz von (1) und (2) wird durch (6.1)a) geliefert, bleibt noch die von (1) und (3) zu zeigen.

"(1) $\Rightarrow$ (3)": Aus $\alpha_k^{OD}=0$ folgt zum einen $\alpha_k^{MCR}\neq 0$ nach Teil a) und somit die Nichtsingularität von T_{k+1} nach Teil b) dieses Satzes,

zum anderen $x_{k+1}^{OD}=x_k^{OD}$ und nach (4.11)a)

$$\bar{x} - x_k^{OD} \perp \bar{S}_{k+1} \quad .$$

Wegen $\bar{S}_k=[Ar_o,\ldots,A^k r_o]\subseteq[r_o,Ar_o,\ldots,A^k r_o]=S_{k+1}$ erhält man

$$x_k^{OD} \in x_o + S_{k+1}$$

und daher $x_{k+1}^{c}=x_k^{OD}$ nach (4.11)b).

"(3) $\Rightarrow$ (1)": Sei also T_{k+1} nichtsingulär und $x_{k+1}^{c}=x_k^{OD}$. (4.11)b) liefert $\bar{x}-x_k^{OD}\perp \bar{S}_{k+1}$, ferner gilt stets $x_k^{OD}\in x_o+\bar{S}_k\subseteq x_o+\bar{S}_{k+1}$, so daß (4.11)a) $x_k^{OD}=x_{k+1}^{OD}=x_k^{OD}+\alpha_k^{OD}p_k^{OD}$ und wegen $k<m$, $p_k\neq 0$ somit $\alpha_k^{OD}=0$ impliziert. □

Wir wollen nun zeigen, daß es eine ganze Klasse von Beispielen gibt, für die sowohl OD- als auch MCR-Algorithmus in dem Sinne "entartet" sind, daß in jeder zweiten Iteration Schrittweite Null auftritt. Es sei zunächst an die Darstellung (3.2) erinnert:

$$e_o = \sum_{j=1}^{m} \rho_j z_j \quad , \quad \rho_1,\ldots,\rho_m \neq 0 \ ,$$

mit orthonormalen Eigenvektoren $z_1,\ldots,z_m$, die zu paarweise verschiedenen Eigenwerten $\lambda_1,\ldots,\lambda_m$ von A gehören. Wir sprechen vom <u>Entartungsfall</u>, wenn A und e_o so beschaffen sind, daß

(6.4) a) $m/2 \in \mathbb{N}$;

b) $\lambda_j = -\lambda_{m+1-j}$, $j = 1,2,\ldots,m/2$;

c) $|\rho_j| = |\rho_{m+1-j}|$, $j = 1,2,\ldots,m/2$

gilt. Einen Vektor der Form $x=\sum_{j=1}^{m}\xi_j z_j$ nennen wir

$$\left\{\begin{array}{l}\underline{\text{gerade}}\\ \\ \underline{\text{ungerade}}\end{array}\right\}, \text{ falls } \xi_j = \left\{\begin{array}{l}\xi_{m+1-j}\\ \\ -\xi_{m+1-j}\end{array}\right\} \quad \text{für } j = 1,2,\ldots,m/2.$$

Offensichtlich gilt:

(6.5) a) $\alpha x + \beta y$ ist $\left\{\begin{matrix}\text{gerade}\\ \\ \text{ungerade}\end{matrix}\right\}$, falls x und y $\left\{\begin{matrix}\text{gerade}\\ \\ \text{ungerade}\end{matrix}\right\}$, $\alpha,\beta\in\mathbb{R}$;

b) $x^T y = 0$, falls x gerade und y ungerade;

und unter Berücksichtigung von (6.4)b):

c) Ax ist $\left\{\begin{matrix}\text{gerade}\\ \\ \text{ungerade}\end{matrix}\right\}$, falls x $\left\{\begin{matrix}\text{ungerade}\\ \\ \text{gerade}\end{matrix}\right\}$;

d) $x^T Ax = 0$, falls x gerade oder ungerade.

(6.6) Satz: Im Entartungsfall gilt:

a) $\alpha_{2k+1}^{OD} = \alpha_{2k}^{MCR} = 0$, $\alpha_{2k}^{OD} \neq 0$, $\alpha_{2k+1}^{MCR} \neq 0$, $0 \leq k < m/2$;

b) $[p_o^{OD}, p_2^{OD}, p_4^{OD}, \ldots, p_{2k}^{OD}] = [p_1^{MCR}, p_3^{MCR}, p_5^{MCR}, \ldots, p_{2k+1}^{MCR}]$

$= [Ar_o, A^3 r_o, A^5 r_o, \ldots, A^{2k+1} r_o]$, $0 \leq k < m/2$;

c) im SYMMLQ-Algorithmus existieren genau die x_k^c mit geradzahligem Index k;

d) $x_{2k}^{OD} = x_{2k}^c = x_k^{CRAIG}$, $x_{2k}^{MCR} = x_k^{CGN}$, $0 \leq k \leq m/2$,

dabei bezeichnen x_k^{CRAIG} bzw. x_k^{CGN} die Näherungen, welche Algorithmus (12.6) bzw. (12.2) ausgehend von $x_o^{CRAIG} = x_o^{CGN} = x_o$ liefert.

Beweis: Indem man eventuell einige z_j durch $-z_j$ ersetzt, darf wegen (6.4)c) o.B.d.A. angenommen werden, daß e_o gerade ist. Wir zeigen dann durch Induktion nach k, daß die Suchrichtungen p_k des OD-Algorithmus folgendes erfüllen:

(6.7) p_k ist $\left\{\begin{matrix}\text{gerade}\\ \\ \text{ungerade}\end{matrix}\right\}$, falls k $\left\{\begin{matrix}\text{gerade}\\ \\ \text{ungerade}\end{matrix}\right\}$, $0 \leq k \leq m$.

k=0: $p_o = A^2 e_o$ ist gerade nach (6.5)c).

Sei (6.7) bereits bewiesen für alle Indizes$\leq k$, und sei $k+1\leq m$. Mit (6.5)d) folgt $\gamma_k = p_k^T A p_k / p_k^T p_k = 0$ und daher

$$p_{k+1} = Ap_k - \delta_k p_{k-1} \,. \tag{6.8}$$

Aus der Induktionsannahme und mit (6.5)a)c) ergibt sich nun, daß zunächst Ap_k, p_{k-1} und deshalb auch p_{k+1} gerade bzw. ungerade sind, je nachdem ob $k+1$ gerade bzw. ungerade ist.

zu a) Wegen e_o gerade, $r_o = Ae_o$ ungerade folgt mit (6.7) und (6.5)b) $e_o^T p_{2k+1} = r_o^T p_{2k} = 0$, nach (6.1) also

$$\alpha_{2k+1}^{OD} = \alpha_{2k}^{MCR} = 0 \,, \quad 0 \leq k < m/2.$$

Der MCR-Algorithmus liefert stets nach genau m Schritten, der OD-Algorithmus im Entartungsfall wegen (6.4)b) und Satz(3.3)a) nach genau m-1 Schritten die exakte Lösung; beachtet man noch, daß keine zwei aufeinanderfolgenden Schrittweiten verschwinden können, so erhält man

$$\alpha_{2k}^{OD} \neq 0 \,, \; \alpha_{2k+1}^{OD} \neq 0 \,, \quad 0 \leq k < m/2.$$

zu b) Durch Induktion nach k zeigen wir, daß für $0\leq k<m/2$

$$\begin{aligned} [p_o^{OD}, p_2^{OD}, p_4^{OD}, \ldots, p_{2k}^{OD}] &\subseteq [Ar_o, A^3 r_o, A^5 r_o, \ldots, A^{2k+1} r_o] \,, \\ [p_1^{OD}, p_3^{OD}, p_5^{OD}, \ldots, p_{2k+1}^{OD}] &\subseteq [A^2 r_o, A^4 r_o, A^6 r_o, \ldots, A^{2k+2} r_o] \end{aligned} \tag{6.9}$$

erfüllt ist. Für k=0 ist dies richtig, denn es gilt $p_o^{OD} = Ar_o$ und wegen (6.8)($\delta_o = 0$!) $p_1^{OD} = A^2 r_o$. Der Schluß von k auf k+1 ergibt sich unmittelbar aus der Induktionsannahme (6.9) und aus (6.8). Da die p_j^{OD}, $0\leq j<m$, linear unabhängig sind folgt, daß man in (6.9) sogar Gleichheit hat und mit $p_j^{OD} = Ap_j^{MCR}$ die Behauptung.

zu c)d) Mit Teil a) und Satz(6.2)b)c) erkennt man, daß genau die x_k^c mit geradzahligem Index existieren und $x_{2k}^c = x_{2k}^{OD}$, $0\leq k\leq m/2$, gilt.

Wegen Teil a) und b) nehmen die Minimierungseigenschaften (2.6)a) bzw. (5.1) von OD- bzw. MCR-Algorithmus folgende Form an:

$$x_{2k}^{OD} = \arg\min_{x \in x_o + S_k^2} \|x - \bar{x}\| \,, \quad x_{2k}^{MCR} = \arg\min_{x \in x_o + S_k^2} \|b - Ax\|, \; 0 \leq k \leq m/2,$$

wobei

$$S_k^2 := [Ar_o, A^3 r_o, A^5 r_o, \ldots, A^{2k-1} r_o]$$

$$= [Ar_o, (A^2)Ar_o, (A^2)^2 Ar_o, \ldots, (A^2)^{k-1} Ar_o]$$

gesetzt wurde. Vergleich mit (12.8)d) und (12.4)d) liefert

$$x_{2k}^{OD} = x_k^{CRAIG}, \quad x_{2k}^{MCR} = x_k^{CGN}, \quad 0 \leq k \leq m/2. \square$$

Das Verhalten von OD- und MCR-Algorithmus im Entartungsfall könnte die Vermutung entstehen lassen, daß das Auftreten von Nullen in den OD- und MCR-Schrittweitenfolgen stets gekoppelt ist. Dies ist jedoch nicht so, wie folgendes Beispiel zeigt: Für

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} \sqrt{6} \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad x_o = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \bar{x} = \begin{bmatrix} \sqrt{6} \\ -1 \\ 1/3 \\ 1/2 \end{bmatrix}$$

ergibt sich nämlich

k	0	1	2	3
α_k^{OD}	9/20	-1/3	-7/108	5/36
α_k^{MCR}	1/2	0	-5/18	1/6
x_{k+1}^{OD}	$\frac{9}{20}\begin{bmatrix} \sqrt{6} \\ -1 \\ 3 \\ 2 \end{bmatrix}$	$\frac{1}{20}\begin{bmatrix} 47\sqrt{6}/3 \\ -29 \\ 7 \\ 18 \end{bmatrix}$	$\frac{1}{36}\begin{bmatrix} 31\sqrt{6} \\ -41 \\ 7 \\ 38 \end{bmatrix}$	$\begin{bmatrix} \sqrt{6} \\ -1 \\ 1/3 \\ 1/2 \end{bmatrix}$
x_{k+1}^{MCR}	$\frac{1}{2}\begin{bmatrix} \sqrt{6} \\ 1 \\ 1 \\ 1 \end{bmatrix}$		$\frac{5}{6}\begin{bmatrix} \sqrt{6} \\ -1 \\ 1/3 \\ 1 \end{bmatrix}$	

7. Stabile Versionen des OD-Verfahrens

In der Rechenpraxis zeigt sich, daß der OD-Algorithmus(2.2) nach einer Reihe von Iterationen instabil wird. Die Schwachstelle des Algorithmus liegt in der Berechnung der Schrittweite mittels

$$\alpha_k = r_k^T p_{k-1} / p_k^T p_k \,, \tag{7.1}$$

wie wir uns nun plausibel machen wollen. Ausgehend von der Minimierungseigenschaft (Satz(2.6)a))

$$\|x_{k+1} - \bar{x}\| = \min_{\alpha \in \mathbb{R}} \|x_k + \alpha p_k - \bar{x}\|$$

ergibt sich

$$\alpha_k = (\bar{x} - x_k)^T p_k / p_k^T p_k \tag{7.2}$$

als "natürliche" Formel für α_k. Diese ist freilich nicht auswertbar, und man weicht auf (7.1) aus, wobei ausgenutzt wird (vgl. Satz(2.4)e)), daß

$$p_k = Ap_{k-1} - \gamma_{k-1} p_{k-1} - \delta_{k-1} p_{k-2} \tag{7.3}$$

$$(\bar{x} - x_k)^T p_k = r_k^T p_{k-1} - \gamma_{k-1} (\bar{x} - x_k)^T p_{k-1} - \delta_{k-1} (\bar{x} - x_k)^T p_{k-2}$$

und $$(\bar{x} - x_k)^T p_{k-1} = (\bar{x} - x_k)^T p_{k-2} = 0 \quad \text{gilt.}$$

Seien nun $\alpha_k, \gamma_k, \delta_k, x_k, r_k, p_k$ die in Gleitpunktarithmetik tatsächlich berechneten Größen; an die Stelle von (7.3) tritt

$$p_k = Ap_{k-1} - \gamma_{k-1} p_{k-1} - \delta_{k-1} p_{k-2} - \|p_{k-1}\| \varepsilon_{k-1} \tag{7.4}$$

mit einem Fehlervektor ε_{k-1}. Schranken für $\|\varepsilon_{k-1}\|$ hängen von der speziellen Implementierung ab, so zeigte Paige [25], daß für eine von ihm empfohlene Version des Lanczos-Algorithmus

$$\|\varepsilon_{k-1}\| \leq \left(7 + j \frac{\| |A| \|}{\|A\|}\right) \|A\| \text{eps}$$

gilt, dabei ist j die maximale Anzahl von Null verschiedener Elemente in einer Zeile von A und eps die relative Maschinengenauigkeit.

Parlett [27] berichtet, daß in der Praxis bislang keine Abweichungen von

$$\|\varepsilon_{k-1}\| \leq \|A\| \text{ eps}$$

beobachtet wurden.
Infolge von Rundungsfehlern ist unvermeidlich, daß die verwendete Schrittweite α_k von der Schrittweite (7.2), sie sei mit $\tilde{\alpha}_k$ bezeichnet, die ja notwendig wäre, um den euklidischen Fehler längs der mit (7.4) berechneten Suchrichtung p_k exakt zu minimieren, abweicht, und wir wollen das Verhalten von $\Delta\alpha_k := \alpha_k - \tilde{\alpha}_k$ diskutieren. Dazu seien bereits einige, nämlich k_o Iterationsschritte ausgeführt und folgende vereinfachende Annahmen gemacht :

(7.5) a) Für $k > k_o$ sei (7.1) exakt erfüllt und r_k das tatsächliche Residuum von x_k;

b) für $k \geq k_o$ werde $x_k = x_{k-1} + \alpha_{k-1}p_{k-1}$ rundungsfehlerfrei berechnet, und es gelte die lokale Orthogonalität $p_k^T p_{k-1} = 0$.

Für $k>k_o$ folgt dann

$$(\bar{x} - x_k)^T p_{k-1} = (\bar{x} - x_{k-1} - (\tilde{\alpha}_{k-1} + \Delta\alpha_{k-1})p_{k-1})^T p_{k-1} = -\Delta\alpha_{k-1}\|p_{k-1}\|^2,$$

$$(\bar{x} - x_k)^T p_{k-2} = (\bar{x} - x_{k-2} - (\tilde{\alpha}_{k-2} + \Delta\alpha_{k-2})p_{k-2})^T p_{k-2} = -\Delta\alpha_{k-2}\|p_{k-2}\|^2,$$

und mit (7.1),(7.4) erhält man die Rekursion

(7.6)

$$\Delta\alpha_k\|p_k\| = -c_k^{(1)}\Delta\alpha_{k-1}\|p_{k-1}\| - c_k^{(2)}\Delta\alpha_{k-2}\|p_{k-2}\| + (\bar{x} - x_k)^T c_{k-1}\frac{\|p_{k-1}\|}{\|p_k\|},$$

$$c_k^{(1)} := \gamma_{k-1}\frac{\|p_{k-1}\|}{\|p_k\|},\quad c_k^{(2)} := \delta_{k-1}\frac{\|p_{k-2}\|}{\|p_k\|},$$

für die (auf Suchrichtungen mit der Länge 1 bezogenen) Schrittweitenfehler $\Delta\alpha_k\|p_k\|$. Die Annahmen (7.5) sind insofern berechtigt, als numerische Beispiele (siehe Abschnitt 18) zeigen, daß schon nach wenigen Iterationen $\Delta\alpha_k\|p_k\|$ praktisch durch die beiden ersten Summanden der rechten Seite von (7.6) beschrieben werden. Wie diese

Beispiele ferner belegen, können die Faktoren $c_k^{(1)}$, $c_k^{(2)}$ dafür sorgen, daß die beiden letzten Schrittweitenfehler sogar verstärkt zu $\Delta\alpha_k \| p_k \|$ beitragen, somit $|\Delta\alpha_k| \, \| p_k \|$ mit wachsendem k ansteigt, während

$$|\alpha_k| \, \| p_k \| \approx (\| \bar{x} - x_k \|^2 - \| \bar{x} - x_{k+1} \|^2)^{1/2}$$

zunächst abnimmt, bis schließlich $|\Delta\alpha_k| \, \| p_k \|$ in die Größenordnung von $|\alpha_k| \, \| p_k \|$ gerät, die berechnete Schrittweite α_k also für die Suchrichtung p_k unsinnig wird und die weiter gelieferten Näherungswerte für $\bar{x}$ unbrauchbar werden.

Die eben skizzierte Schwierigkeit läßt sich vermeiden, wenn man ausnutzt, daß für die Suchrichtungen des OD- und MCR-Algorithmus nach (5.5)a) $p_k^{OD} = Ap_k^{MCR}$ gilt. Mit (7.2) ergibt sich nämlich für α_k die Formel

$$\alpha_k = r_k^T p_k^{MCR} / (p_k^{MCR})^T A^2 p_k^{MCR} ,$$

die sich auswerten läßt, wenn man die MCR-Suchrichtung zu Verfügung hat. Wir schlagen deshalb eine Variante des OD-Verfahrens vor, bei der anstelle von p_k^{OD} stets p_k^{MCR} und Ap_k^{MCR} upgedated werden. Gegenüber dem Algorithmus(2.2) erhöht sich die Zahl der Multiplikationen um 2n, außerdem müssen zwei weitere Vektoren gespeichert werden. Dieser Mehraufwand wird jedoch durch das stark verbesserte numerische Verhalten gerechtfertigt, weshalb wir die neue Version auch als "Stabilen OD-Algorithmus" bezeichnen wollen. Schreibt man statt p_k^{MCR} einfach p_k, so erhalten wir den

(7.7) STOD-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$p_o = r_o = b - Ax_o, \; p_{-1} = 0.$$

Für k=0,1,2,...

1) Falls $Ap_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst setze

2) $\alpha_k = r_k^T p_k / p_k^T A^2 p_k$,

$$x_{k+1} = x_k + \alpha_k Ap_k, \; r_{k+1} = r_k - \alpha_k A(Ap_k),$$

3)
$$\gamma_k = p_k^T A^3 p_k / p_k^T A^2 p_k, \quad \delta_k = \begin{cases} 0 & \text{für } k = 0 \\ p_k^T A^2 p_k / p_{k-1}^T A^2 p_{k-1} & \text{für } k > 0 \end{cases},$$

$$p_{k+1} = Ap_k - \gamma_k p_k - \delta_k p_{k-1},$$

$$Ap_{k+1} = A(Ap_k) - \gamma_k Ap_k - \delta_k Ap_{k-1}.$$

Der Aufwand des STOD-Algorithmus mit einer Matrixmultiplikation $A\cdot(Ap_k)$, zirka 9n Multiplikationen (die drei Skalarprodukte $r_k^T p_k$, $(Ap_k)^T(Ap_k)$, $(Ap_k)^T(A(Ap_k))$ und sechs Skalar-Vektor-Multiplikationen $\alpha_k Ap_k$, $\alpha_k A(Ap_k)$, $\gamma_k p_k$, $\delta_k p_{k-1}$, $\gamma_k Ap_k$, $\delta_k Ap_{k-1}$) und Speicherplatz für die sieben Vektoren x_k, r_k, p_k, p_{k-1}, Ap_k, Ap_{k-1}, $A(Ap_k)$ ist äquivalent dem des MCR-Algorithmus in der Form (5.4). Wie in Abschnitt 5 angedeutet, schlug Chandra [6] eine Version des MCR-Algorithmus vor, bei der falls $|\alpha_k^{MCR}|>\varepsilon$ ($\varepsilon>0$ geeignet gewählt) von einer verkürzten Rekursion für die neunormierte Suchrichtung $\tilde{p}_{k+1}=-\alpha_k^{MCR} p_{k+1}$ Gebrauch gemacht wird.
Für den STOD-Algorithmus läßt sich etwas ähnliches realisieren. Falls $|\alpha_k|\leq\varepsilon$ benutzen wir die Form (7.7), falls $|\alpha_k|>\varepsilon$ rechnen wir mit $\tilde{p}_{k+1}=\alpha_k p_{k+1}$, $A\tilde{p}_{k+1}=\alpha_k Ap_{k+1}$ weiter. Wegen

$$\tilde{p}_{k+1} = \alpha_k Ap_k - (\gamma_k\alpha_k)p_k - (\delta_k\alpha_k)p_{k-1},$$

$$A\tilde{p}_{k+1} = A(\alpha_k Ap_k) - \gamma_k(\alpha_k Ap_k) - (\delta_k\alpha_k)Ap_{k-1}$$

und unter Beachtung von

$$x_{k+1} = x_k + \alpha_k Ap_k, \quad r_{k+1} = r_k - A(\alpha_k Ap_k)$$

erkennt man, daß die Berechnung von $\alpha_k(Ap_k)$, $A(\alpha_k Ap_k)$ genügt, während im STOD-Algorithmus $\alpha_k Ap_k$, $A(Ap_k)$ und $\alpha_k A(Ap_k)$ benötigt werden. Auf diese Weise läßt sich in den Iterationsschritten mit $|\alpha_k|>\varepsilon$ die Anzahl der Multiplikationen um etwa n senken. Die so erhaltene Modifikation des STOD-Algorithmus läßt sich folgendermaßen formulieren:

(7.8) MSTOD-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$p_o = r_o = b - Ax_o,\ p_{-1} = 0,\ \sigma_{-1} = 0.$$

Für k=0,1,2,...

1) Falls $Ap_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst,

2) berechne

$$\rho_k = p_k^T A^2 p_k,\ \alpha_k = r_k^T p_k / \rho_k.$$

3) Falls $|\alpha_k| \leq \epsilon$, setze

$$x_{k+1} = x_k + \alpha_k Ap_k,\ r_{k+1} = r_k - \alpha_k A(Ap_k),\ \sigma_k = 1;$$

falls $|\alpha_k| > \epsilon$, definiere neu

$$Ap_k := \alpha_k Ap_k,\ \rho_k := \alpha_k^2 \rho_k$$

und setze

$$x_{k+1} = x_k + Ap_k,\ r_{k+1} = r_k - A(Ap_k),\ \sigma_k = \alpha_k.$$

4) Setze

$$\gamma_k = p_k^T A^3 p_k / \rho_k,\ \delta_k = \begin{cases} 0 & \text{für } k = 0 \\ \rho_k / (\rho_{k-1}\sigma_k) & \text{für } k > 0 \end{cases},$$

$$p_{k+1} = Ap_k - (\gamma_k \sigma_k) p_k - (\delta_k \sigma_{k-1}) p_{k-1},$$

$$Ap_{k+1} = A(Ap_k) - \gamma_k Ap_k - \delta_k Ap_{k-1}.$$

Für die bislang betrachteten Algorithmen zur Lösung von Ax=b stellen wir die Zahl der Matrix-Vektor-Multiplikationen (A·x) pro Iteration, die ungefähre Anzahl M der Multiplikationen pro Iteration, sowie die Zahl N der zu speichernden Vektoren des $\mathbb{R}^n$ in folgender Tabelle zusammen:

	A·x	M	N
OD(2.2)	1	7n	5
STOD(7.7)	1	9n	7
MSTOD(7.8)	1	8n - 9n	7
MCR(5.4)	1	9n	7
MCR(Chandra)	1	7n - 9n	7
SYMMLQ†	1	9n	5

†Die Angaben beziehen sich auf die Version des SYMMLQ-Algorithmus, bei der nur die x_k^L-Größen berechnet werden.

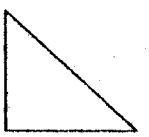
Bei der Formulierung der Algorithmen wurde als Stop-Kriterium stets die Abfrage $p_k=0$ bzw. $Ap_k=0$ verwendet, was man ohne zusätzliche Rechenarbeit testen kann, da $p_k^T p_k = \|p_k\|^2$ (falls p_k OD-Suchrichtung) bzw. $p_k^T A^2 p_k = \|Ap_k\|^2$ (falls p_k MCR-Suchrichtung) zur Verfügung steht. In der Praxis geht infolge von Rundungsfehlern die Orthogonalität der p_k jedoch rasch verloren, und man kann nicht einmal mehr erwarten, daß $\|p_k\|$ klein wird. Man ist daher in der Regel auf das Residuum angewiesen und bricht den Algorithmus ab, falls $\|r_k\|$ genügend klein ist. Zumindest in einem Teil der Iterationsschritte fällt daher noch die Berechnung des Skalarprodukts $r_k^T r_k$ an.

8. Prekonditionierte Varianten

Durch die Ergebnisse aus Abschnitt 3 wird nahegelegt, in Analogie zum prekonditionierten CG-Verfahren (siehe etwa [1,17,22,32]) prekonditionierte Versionen des OD-Verfahrens zu betrachten.
Sei Q eine nichtsinguläre n×n-Matrix mit den Eigenschaften:

(8.1) a) Q sei dünn besetzt, und Gleichungssysteme der Form

$$QQ^T q = p$$

seien leicht lösbar (z.B. Q =  eine untere Dreiecksmatrix);

b) $\kappa(Q^{-1}AQ^{-T}) \ll \kappa(A)$.

Mit $A' = Q^{-1}AQ^{-T}$, $b' = Q^{-1}b$, $x' = Q^T x$
ist das Gleichungssystem

(8.2) $$Ax = b$$

äquivalent zu

(8.3) $$A'x' = b'.$$

Statt auf (8.2) wollen wir nun das OD-Verfahren auf (8.3) anwenden, denn wegen $\kappa(A') \ll \kappa(A)$ ist nach Satz(3.8) dann nämlich schnellere Konvergenz zu erwarten. Ausgehend von dem Startwert x'_o $(=Q^T x_o)$ erhält man so eine Folge von Näherungswerten x'_k für die Lösung $\bar{x}'$ $(=Q^T\bar{x})$ von (8.3). Entscheidend ist jetzt, daß sich OD(bzw. STOD und MSTOD)-Algorithmus in einer Form schreiben lassen, welche die explizite Berechnung von A', b' vermeidet und außerdem gleich die auf das ursprüngliche System (8.2) bezogenen Größen $x_k = Q^{-T}x'_k$ und $r_k = b - Ax_k = Qr'_k$ liefert.
Transformiert man die von Algorithmus(2.2) (angewandt auf (8.3)) berechneten Vektoren x'_k, r'_k, p'_k mittels

$$x_k = Q^{-T}x'_k,\quad r_k = Qr'_k,\quad p_k = Qp'_k,$$

setzt

$$q_k := Q^{-T}Q^{-1}p_k$$

und beachtet, daß

$$r_k'^T p_{k-1}' = r_k^T Q^{-T} Q^{-1} p_{k-1} = r_k^T q_{k-1},$$

$$p_k'^T p_k' = p_k^T Q^{-T} Q^{-1} p_k = p_k^T q_k,$$

$$p_k'^T A' p_k' = p_k^T Q^{-T} Q^{-1} A Q^{-T} Q^{-1} p_k = q_k^T A q_k$$

gilt, so erhält man die folgende prekonditionierte Version des OD-Algorithmus:

(8.4) PCOD-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$p_{-1} = r_o = b - Ax_o$,

erhalte q_{-1} als Lösung von $QQ^T q_{-1} = p_{-1}$

und setze $p_o = Aq_{-1}$.

Für k=0,1,2,...

1) Falls $p_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst,

2) erhalte q_k als Lösung von $QQ^T q_k = p_k$.

3) Setze

$$\alpha_k = r_k^T q_{k-1} / p_k^T q_k,$$

$$x_{k+1} = x_k + \alpha_k q_k,\ r_{k+1} = r_k - \alpha_k A q_k$$

und

4)
$$\gamma_k = q_k^T A q_k / p_k^T q_k,\ \delta_k = \begin{cases} 0 & \text{für } k = 0 \\ p_k^T q_k / p_{k-1}^T q_{k-1} & \text{für } k > 0 \end{cases},$$

$$p_{k+1} = Aq_k - \gamma_k p_k - \delta_k p_{k-1}.$$

Entsprechend ergibt sich eine prekonditionierte Version des STOD-Algorithmus, wenn man die von (7.7) erzeugten Vektoren x_k', r_k', p_k' gemäß

$$x_k = Q^{-T} x_k',\ r_k = Q r_k',\ p_k = Q^{-T} p_k'$$

transformiert, außerdem

$$q_k := Q^{-T} Q^{-1} (Ap_k)$$

setzt und berücksichtigt, daß

$$r_k'^T p_k' = r_k^T Q^{-T} Q^T p_k = r_k^T p_k,$$

$$p_k'^T (A')^2 p_k' = p_k^T A Q^{-T} Q^{-1} A p_k = q_k^T A p_k,$$

$$p_k'^T (A')^3 p_k' = p_k^T A Q^{-T} Q^{-1} A Q^{-T} Q^{-1} A p_k = q_k^T A q_k$$

gilt.

(8.5) PCSTOD-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$r_o = b - Ax_o$, $p_{-1} = 0$,

erhalte p_o als Lösung von $QQ^T p_o = r_o$.

Für k=0,1,2,...

1) Falls $Ap_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst,

2) erhalte q_k als Lösung von $QQ^T q_k = Ap_k$.

3) Setze

$$\alpha_k = r_k^T p_k / q_k^T A p_k,$$

$$x_{k+1} = x_k + \alpha_k q_k,\ r_{k+1} = r_k - \alpha_k A q_k$$

und

4)

$$\gamma_k = q_k^T A q_k / q_k^T A p_k,\ \delta_k = \begin{cases} 0 & \text{für } k = 0 \\ q_k^T A p_k / q_{k-1}^T A p_{k-1} & \text{für } k > 0 \end{cases},$$

$$p_{k+1} = q_k - \gamma_k p_k - \delta_k p_{k-1},$$

$$Ap_{k+1} = Aq_k - \gamma_k Ap_k - \delta_k Ap_{k-1}.$$

Die Überlegungen, die vom STOD- zum MSTOD-Algorithmus führten, liefern auch eine modifizierte Fassung von (8.5):

(8.6) MPCSTOD-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$r_o = b - Ax_o$, $p_{-1} = 0$,

erhalte p_o als Lösung von $QQ^T p_o = r_o$.

Für k=0,1,2,...

1) Falls $Ap_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst,

2) erhalte q_k als Lösung von $QQ^T q_k = Ap_k$.

3) Berechne

$$\rho_k = q_k^T Ap_k, \quad \alpha_k = r_k^T p_k / \rho_k.$$

4) Falls $|\alpha_k| \leq \varepsilon$, setze

$$x_{k+1} = x_k + \alpha_k q_k, \quad r_{k+1} = r_k - \alpha_k Aq_k, \quad \sigma_k = 1;$$

falls $|\alpha_k| > \varepsilon$, definiere neu

$$q_k := \alpha_k q_k, \quad \rho_k := \alpha_k^2 \rho_k$$

und setze

$$x_{k+1} = x_k + q_k, \quad r_{k+1} = r_k - Aq_k, \quad \sigma_k = \alpha_k.$$

5) Setze

$$\gamma_k = \sigma_k q_k^T Aq_k / \rho_k, \quad \delta_k = \begin{cases} 0 & \text{für } k = 0 \\ (\rho_k \sigma_{k-1})/(\rho_{k-1}\sigma_k) & \text{für } k > 0 \end{cases},$$

$$p_{k+1} = q_k - \gamma_k p_k - \delta_k p_{k-1},$$

$$Ap_{k+1} = Aq_k - \gamma_k Ap_k - \delta_k Ap_{k-1}.$$

Aus den Minimierungs- und Konvergenzeigenschaften des OD-Verfahrens (angewandt auf (8.3)) gewinnt man durch Transformation auf die Größen des Ausgangssystems (8.2) sofort entsprechende Resultate für die prekonditionierten Versionen. Wir setzen

$$M := QQ^T,$$

$$\bar{S}_k := [(M^{-1}A)(M^{-1}r_o), (M^{-1}A)^2(M^{-1}r_o), \ldots, (M^{-1}A)^k(M^{-1}r_o)] ;$$

unter Beachtung, daß wegen

$$\|x'_k - \bar{x}'\|^2 = (x_k - \bar{x})^T M (x_k - \bar{x}) = \|e_k\|_M^2$$

die Fehler $e_k = \bar{x} - x_k$ nun in der M-Norm gemessen werden, erhalten wir aus (2.6)a) und (3.8) den folgenden

(8.7) Satz: Für die von den Algorithmen(8.4)-(8.6) ausgehend von x_o gelieferten Näherungswerte x_k, k=1,2,..., gilt:

a) $$\|e_k\|_M = \min_{x \in x_o + \bar{S}_k} \|x - \bar{x}\|_M ;$$

b) $$\frac{\|e_k\|_M}{\|e_o\|_M} \leq \frac{1}{T_{[(k+1)]/2}\left(\frac{\kappa'^2 + 1}{\kappa'^2 - 1}\right)} \leq 2 \left(\frac{\kappa' - 1}{\kappa' + 1}\right)^{[(k+1)/2]},$$

wobei $\kappa' := \kappa(Q^{-1}AQ^{-T}) = \kappa(M^{-1}A)$.

Gegenüber den OD-Algorithmen (2.2),(7.7),(7.8) ist bei den prekonditionierten Varianten (8.4)-(8.6) zusätzlich pro Iteration ein Gleichungssystem mit Koeffizientenmatrix QQ^T zu lösen, ferner erhöht sich die Zahl der zu speichernden Vektoren um eins. Außerdem ist zu berücksichtigen, daß man im allgemeinen Q erst aus A berechnen muß, und daß für die wesentlichen Elemente von Q Speicherplatz benötigt wird. Der Einsatz der prekonditionierten Typen ist daher nur gerechtfertigt, wenn man eine Matrix Q mit den Eigenschaften (8.1) findet und durch (8.1)b) die Anzahl der Iterationen derart gedrückt wird, daß die gesamte Rechenarbeit unter der des gewöhnlichen OD-Verfahrens bleibt.

Zu einer beliebigen symmetrischen Matrix A eine Prekonditionierungsmatrix Q mit (8.1) zu finden, ist ein recht schwieriges Problem. Möglicherweise lassen sich (in Analogie zur unvollständigen Cholesky-Zerlegung bei positiv definiten Matrizen) mit einer Art unvollständiger Bunch-Parlett-Zerlegung [4,5] brauchbare Resultate erzielen. Wegen der Notwendigkeit von Pivotsuche und dem möglichen Auftreten von 2×2-Pivotelementen liegen die Dinge für indefinite Matrizen allerdings um einiges komplizierter.

Mit einer speziellen Klasse indefiniter Matrizen, auf die sich die Ergebnisse, die Meijerink und van der Vorst [22] für M-Matrizen gewannen, unmittelbar übertragen lassen, beschäftigen wir uns im nächsten Abschnitt.

9. Unvollständige Faktorisierung von H-Matrizen

Sei $A=(a_{ij})$ eine reelle, zunächst nicht notwendig symmetrische $n\times n$-Matrix.

(9.1) Definition: a) A heißt M-Matrix, wenn es ein $s\in\mathbb{R}$ und eine nichtnegative Matrix $B\geq 0$ mit Spektralradius $\rho(B)<s$ gibt, so daß sich A in der Form $A=sI-B$ darstellen läßt.

b) A heißt H-Matrix, wenn die durch $\hat{a}_{ii}=|a_{ii}|$, $|\hat{a}_{ij}|=-|a_{ij}|$, $i,j=1,\ldots,n$, $i\neq j$, definierte Matrix $\hat{A}=(\hat{a}_{ij})$ eine M-Matrix ist.

Für die Eigenschaft "M-Matrix" gibt es zahlreiche äquivalente Charakterisierungen (siehe Berman und Plemmons [3]), so ist eine Matrix A mit $a_{ij}\leq 0$, $i\neq j$, eine M-Matrix genau dann, wenn $a_{ii}\geq 0$, $i=1,\ldots,n$, gilt und es eine positive Diagonalmatrix D gibt, so daß $D^{-1}AD$ strikt diagonaldominant ist. Damit erhält man

(9.2) Satz: A ist eine H-Matrix genau dann, wenn es eine Diagonalmatrix $D=\mathrm{diag}(d_1,\ldots,d_n)$ mit $d_i>0$, $i=1,\ldots,n$, gibt, so daß $D^{-1}AD$ strikt diagonaldominant ist, d.h.

$$|a_{ii}| > \sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}|d_j/d_i \,, \qquad i = 1,\ldots,n,$$

gilt.

In Verbindung mit dem Satz von Gerschgorin liefert (9.2) eine interessante Eigenschaft der H-Matrizen:

(9.3) Korollar: Sei A eine H-Matrix und $u:=|\{i\,|\,a_{ii}>0\}|$. Dann gilt: A besitzt genau u Eigenwerte (mehrfache Eigenwerte entsprechend ihrer algebraischen Vielfachheit gezählt) in der rechten Halbebene $\{z\,|\,\mathrm{Re}\ z > 0\}$, genau $n-u$ in der linken $\{z\,|\,\mathrm{Re}\ z < 0\}$.
Insbesondere ist jede H-Matrix nichtsingulär.

Von Meijerink und van der Vorst [22] stammt das folgende Resultat: Zu jeder M-Matrix A und jeder Menge $G \subseteq G_n := \{(i,j) \mid 1 \leq i,j \leq n\}$ mit $(i,i) \in G$, $i=1,\ldots,n$, existieren Matrizen

$$L = \begin{bmatrix} 1 & & & 0 \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix}, \qquad U = \begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ & 1 & \ddots & \vdots \\ & & \ddots & u_{n-1,n} \\ 0 & & & 1 \end{bmatrix},$$

$D = \operatorname{diag}(d_1,\ldots,d_n)$, $R = (r_{ij})$,

so daß A die unvollständige Zerlegung

$$A = LDU - R, \tag{9.4}$$

mit

$$l_{ji} = 0 \quad \text{falls} \quad i < j \quad \text{und} \quad (j,i) \notin G,$$

$$u_{ij} = 0 \quad \text{falls} \quad i < j \quad \text{und} \quad (i,j) \notin G,$$

$$r_{ij} = 0 \quad \text{falls} \quad (i,j) \in G,$$

besitzt.

Die Elemente von L,D,U,R sind eindeutig bestimmt und lassen sich rekursiv berechnen:

Für $i=1,\ldots,n$:

$$\text{(9.5) a)} \quad d_i = a_{ii} - \sum_{k=1}^{i-1} d_k l_{ik} u_{ki}, \quad r_{ii} = 0;$$

b) für $j = i+1,\ldots,n$:

$$\left.\begin{array}{ll} \text{falls } (j,i) \in G, & d_i l_{ji} \\ \text{falls } (j,i) \notin G, & -r_{ji} \end{array}\right\} = a_{ji} - \sum_{k=1}^{i-1} d_k l_{jk} u_{ki},$$

$$\left.\begin{array}{ll} \text{falls } (i,j) \in G, & d_i u_{ij} \\ \text{falls } (i,j) \notin G, & -r_{ij} \end{array}\right\} = a_{ij} - \sum_{k=1}^{i-1} d_k l_{ik} u_{kj},$$

$$u_{ij} = l_{ji} = r_{ij} = r_{ji} = 0 \quad \text{sonst.}$$

Manteuffel [21] erkannte, daß dieses Resultat auch für die umfassendere Klasse der H-Matrizen mit positiven Diagonalelementen gilt. Man sieht leicht, daß es sich auch auf beliebige H-Matrizen ausdehnen läßt (siehe auch Varga, Saff, Mehrmann [35]):

(9.6) Satz: Sei $A = (a_{ij})$ eine H-Matrix, $\hat{A} = (\hat{a}_{ij})$ mit $\hat{a}_{ii} = |\hat{a}_{ii}|$, $\hat{a}_{ij} = -|a_{ij}|$, $i \neq j$, die zugehörige M-Matrix. Dann besitzen A und $\hat{A}$ zu jeder Menge $G \subseteq G_n$ mit $(i,i) \in G$, $i = 1,\ldots,n$, Zerlegungen (8.4)

$$A = LDU - R, \quad \hat{A} = \hat{L}\hat{D}\hat{U} - \hat{R},$$

und es gilt für $i = 1,\ldots,n$:

a) $0 < \hat{d}_i \leq |d_i|$,

b) $\operatorname{sgn} d_i = \operatorname{sgn} a_{ii}$,

c) $\hat{l}_{ji} \leq -|l_{ji}|$, $\hat{u}_{ij} \leq -|u_{ij}|$, $j = i+1,\ldots,n$,

d) $|r_{ji}| \leq \hat{r}_{ji}$, $|r_{ij}| \leq \hat{r}_{ij}$, $j = i,\ldots,n$.

Beweis: $\hat{A}$ ist M-Matrix und besitzt daher die geforderte Zerlegung, weiter gilt $0<\hat{d}_i$, $i=1,\ldots,n$, nach Meijerink und van der Vorst [22]. Offensichtlich wird durch (9.5) auch für A die gewünschte Zerlegung definiert, falls man $d_i \neq 0$, $i=1,\ldots,n$, garantieren kann; insbesondere folgt ihre Existenz aus $|d_i| \geq \hat{d}_i$. Der Beweis des Satzes ist also erbracht, wenn nachgewiesen ist, daß für $i=1,\ldots,n$ die Aussage (A_i) gilt:

(A_i) (1) $\hat{d}_i \leq |d_i|$, $\operatorname{sgn} d_i = \operatorname{sgn} a_{ii}$,

(2) $|d_i l_{ji}| \leq -\hat{d}_i \hat{l}_{ji}$, $|d_i u_{ij}| \leq -\hat{d}_i \hat{u}_{ij}$, $j = i+1,\ldots,n$,

(3) $|r_{ji}| \leq \hat{r}_{ji}$, $|r_{ij}| \leq \hat{r}_{ij}$, $j = i,\ldots,n$.

(A_i) wird unter Verwendung von (9.5) (angewandt auf A bzw. $\hat{A}$) mittels Induktion nach i gezeigt. Wir beweisen jeweils nur die erste Ungleichung von (2) bzw. (3), die zweite folgt völlig analog; außerdem kann man sich bei (2) auf Indexpaare $(j,i) \in G$, bei (3) auf $(j,i) \notin G$ beschränken.

$i = 1$: (1) gilt wegen $d_1 = a_{11}$, $\hat{d}_1 = |a_{11}|$,

(2) wegen $|d_1 l_{j1}| = |a_{j1}| = -\hat{d}_1 \hat{l}_{j1}$, $(j,1) \in G$,

(3) wegen $|r_{j1}| = |a_{j1}| = \hat{r}_{j1}$, $(j,1) \notin G$.

Sei nun (A_k) bereits gezeigt für $1 \leq k < i$, und sei $i \leq n$. Dann gilt für $1 \leq k \leq i-1$, $i \leq j \leq n$ wegen (1), (2)

$$|d_k l_{jk} u_{ki}| \leq (\hat{d}_k / |d_k|)(\hat{d}_k \hat{l}_{jk} \hat{u}_{ki}) \leq \hat{d}_k \hat{l}_{jk} \hat{u}_{ki}$$

und somit

$$(9.7) \qquad \left| \sum_{k=1}^{i-1} d_k l_{jk} u_{ki} \right| \leq \sum_{k=1}^{i-1} \hat{d}_k \hat{l}_{jk} \hat{u}_{ki} \ .$$

Speziell für $j=i$ erhält man mit (9.5)a)

$$0 < \hat{d}_i = |a_{ii}| - \sum_{k=1}^{i-1} \hat{d}_k \hat{l}_{ik} \hat{u}_{ki} \leq |a_{ii}| - \left| \sum_{k=1}^{i-1} d_k l_{ik} u_{ki} \right|$$

$$\leq \left| a_{ii} - \sum_{k=1}^{i-1} d_k l_{ik} u_{ki} \right| = |d_i| ,$$

also (1). Für $j>i$ ergibt sich mit (9.5)b),(9.7):

$$\left.\begin{array}{ll} \text{falls } (j,i) \in G & |d_i l_{ji}| \\ \text{falls } (j,i) \notin G & |r_{ji}| \end{array}\right\} = \left| a_{ji} - \sum_{k=1}^{i-1} d_k l_{jk} u_{ki} \right|$$

$$\leq |a_{ji}| + \sum_{k=1}^{i-1} \hat{d}_k \hat{l}_{jk} \hat{u}_{ki} = \begin{cases} -\hat{d}_i \hat{l}_{ji} \\ \hat{r}_{ji} \end{cases} ,$$

also (2), (3). □

Sei im weiteren nun A stets eine symmetrische H-Matrix, $\hat{A}$ ist dann eine symmetrische M-Matrix, also positiv definit. Für die Menge $G \subseteq G_n$ verlangen wir:

$$(9.8) \qquad (i,i) \in G \quad \text{für} \quad i = 1,\ldots,n,$$

$$(i,j) \in G \quad \text{falls} \quad (j,i) \in G.$$

Zu jedem solchen G besitzt A nach dem letzten Satz eine unvollständige Cholesky-Zerlegung der Form:

$A = QSQ^T - R$, $q_{ij} = 0$ falls $(i,j) \notin G$, $r_{ij} = 0$ falls $(i,j) \in G$,

wobei

(9.9) $S = \mathrm{diag}(s_1,\ldots,s_n)$ mit $s_i = \mathrm{sgn}\, a_{ii}$, $i = 1,\ldots,n$,

$$Q = L|D|^{1/2}$$

gesetzt wurde. Q läßt sich als Prekonditionierungsmatrix für die Algorithmen des letzten Abschnitts verwenden, wenn man Q geeignet wählt. Nämlich einerseits klein genug, so daß sich Q ohne allzu großen Aufwand berechnen läßt, Q wenig Speicherplatz benötigt und zum Lösen von $QQ^Tq=p$ nicht zuviel Rechenarbeit investiert werden muß, andererseits aber groß genug, damit $QSQ^T \approx A$ hinreichend gut erfüllt ist und man daher $\kappa(A') \ll \kappa(A)$ erwarten kann.
Mit Hilfe der unvollständigen Cholesky-Zerlegung von $\hat{A}$

$$\hat{A} = \hat{Q}\hat{Q}^T - \hat{R},\ \hat{Q} = \hat{L}\hat{D}^{1/2},$$

läßt sich eine Abschätzung für $\kappa(A')$ gewinnen.

(9.10) <u>Satz:</u> Sei A eine symmetrische H-Matrix, $\hat{A}$ die zugehörige M-Matrix, $G \subseteq G_n$ eine Menge mit (9.8), sowie

$$A = QSQ^T - R,\ \hat{A} = \hat{Q}\hat{Q}^T - \hat{R},$$

$$A' = Q^{-1}AQ^{-T} = S - R',\ \hat{A}' = \hat{Q}^{-1}\hat{A}\hat{Q}^{-T} = I - \hat{R}'$$

die zu G gehörigen unvollständigen Cholesky-Zerlegungen und schließlich $\mu=\mu(G)$ der kleinste Eigenwert von $\hat{A}'$.

Dann gilt:

a) $0 \leqslant |Q^{-1}| \leqslant \hat{Q}^{-1},\ 0 \leqslant |R'| \leqslant \hat{R}'$;

b) $\rho(R') \leqslant \rho(\hat{R}') = 1 - \mu < 1$;

c) $\kappa(A') \leqslant \frac{2 - \mu}{\mu}$;

d) A und QSQ^T besitzen jeweils $u = |\{i|a_{ii} > 0\}|$ positive, n-u negative Eigenwerte;

e) $\mu(G)$ ist monoton in dem Sinne, daß

$$\mu(G^{(1)}) \leqslant \mu(G^{(2)})$$

für zwei Mengen $G^{(1)} \subseteq G^{(2)} \subseteq G_n$ mit (9.8).

<u>Beweis:</u> zu a) L und $\hat{L}$ sind untere Dreiecksmatrizen mit Einsen in der Diagonale und daher von der Form

$$L = I - T,\ \hat{L} = I - \hat{T},$$

mit echten unteren Dreiecksmatrizen T und $\hat{T}$; es folgt

$$L^{-1} = I + T + T^2 + \cdots + T^{n-1},\ \hat{L}^{-1} = I + \hat{T} + \hat{T}^2 + \cdots + \hat{T}^{n-1}.$$

Nach Satz(9.6)a)c)d) gilt

$$0 < \hat{D} \leqslant |D|,\quad 0 \leqslant |T| \leqslant \hat{T},\quad 0 \leqslant |R| \leqslant \hat{R},$$

man erhält $0 \leqslant |L^{-1}| \leqslant \hat{L}^{-1}$, sowie

$$0 \leqslant |Q^{-1}| = |\,|D|^{-1/2} L^{-1}| \leqslant \hat{D}^{-1/2} \hat{L}^{-1} = \hat{Q}^{-1},$$
$$0 \leqslant |R'| = |Q^{-1} R Q^{-T}| \leqslant \hat{Q}^{-1} \hat{R} \hat{Q}^{-T} = \hat{R}'.$$

zu b)c) Wir zitieren zunächst ein Resultat aus der Theorie der nichtnegativen Matrizen (siehe Varga [34]):

(9.11) i) Sei $B \geqslant 0$ eine nichtnegative Matrix. Dann ist $\rho(B)$ ein Eigenwert von B.

ii) Sei $0 \leqslant |C| \leqslant B$. Dann gilt $\rho(C) \leqslant \rho(B)$.

Die Matrix $\hat{A}' = \hat{Q}^{-1} \hat{A} \hat{Q}^{-T}$ ist positiv definit, und für ihre Eigenwerte μ_i gilt o.B.d.A.

$$0 < \mu = \mu_1 \leqslant \mu_2 \leqslant \cdots \leqslant \mu_n.$$

Die nichtnegative Matrix $\hat{R}' = I - \hat{A}'$ besitzt dann die Eigenwerte

$$1 - \mu_n \leqslant \cdots \leqslant 1 - \mu_2 \leqslant 1 - \mu_1 = 1 - \mu < 1,$$

und wegen (9.11)i) folgt $\rho(\hat{R}') = 1-\mu$. Mit $0 \leqslant |R'| \leqslant \hat{R}'$ und (9.11)ii) ergibt sich $\rho(R') \leqslant \rho(\hat{R}') = 1-\mu < 1$, also b).

Wegen $A' = S - R'$, $|S| = I$ gilt $0 \leqslant |A'| \leqslant I + |R'| \leqslant I + \hat{R}'$ und somit

(9.12) $$\rho(A') \leqslant 2 - \mu\ .$$

Aus $\hat{A}' = I - \hat{R}'$, $\rho(\hat{R}') < 1$ folgt $(\hat{A}')^{-1} = \sum_{k=0}^{\infty} (\hat{R}')^k$,

aus $SA' = I - SR'$, $|SR'| \leqslant |R'| \leqslant \hat{R}'$, $\rho(SR') \leqslant \rho(\hat{R}') < 1$ folgt

$$(A')^{-1} = (SA')^{-1} S = \Big(\sum_{k=0}^{\infty} (SR')^k\Big) S$$

und

$$|(A')^{-1}| \leqslant \sum_{k=0}^{\infty} (\hat{R}')^k = (\hat{A}')^{-1} .$$

Wir erhalten $\rho((A')^{-1}) \leqslant \rho((\hat{A}')^{-1}) = 1/\mu$ und mit (9.12) $\kappa(A') \leqslant (2-\mu)/\mu$, also c).

d) resultiert aus (9.9), dem Satz von Sylvester und Korollar(9.3);

e) wurde von Manteuffel [21] gezeigt.□

10. Verhalten bei singulären Matrizen

Bislang hatten wir das OD-Verfahren nur auf Gleichungssysteme

(10.1) $$Ax = b$$

mit nichtsingulärer Koeffizientenmatrix angewendet, wir wollen jetzt das Verhalten im singulären Fall klären. In diesem und dem folgenden Abschnitt sei daher für die $n\times n$-Matrix $A=A^T$ lediglich Symmetrie vorausgesetzt, die Möglichkeit, daß A singulär ist, also ausdrücklich miteingeschlossen. OD- und STOD-Algorithmus liefern auch im singulären Fall dieselben Näherungen, denn offensichtlich gilt weiterhin $p_k^{OD}=Ap_k^{MCR}$ und auch die Schrittweiten stimmen überein, wie sich als einfache Konsequenz aus Satz(10.5)f) ergibt. Da für die Praxis nur die STOD-Version in Frage kommt und außerdem auch das Verhalten des MCR-Algorithmus diskutiert werden soll, bedienen wir uns der Formeln aus (7.7), insbesondere sei p_k stets die MCR-Suchrichtung. Ferner wird das CG-Verfahren angesprochen, es sei vereinbart, daß in diesem Fall A als positiv semidefinit angenommen wird. Falls nötig unterscheiden wir die Größen der drei Algorithmen durch oben stehendes OD, MCR bzw. CG; wir nehmen an, daß alle drei Verfahren mit demselben Wert $x_o\in\mathbb{R}^n$ gestartet werden, der folgendermaßen zerlegt sei:

$$x_o = \tilde{x}_o + \tilde{\tilde{x}}_o,\ \tilde{x}_o\in R(A),\ \tilde{\tilde{x}}_o\in R(A)^{\perp} = N(A).$$

Entsprechend wird die rechte Seite von (10.1)

$$b = \tilde{b} + \tilde{\tilde{b}},\ \tilde{b}\in R(A),\ \tilde{\tilde{b}}\in N(A)$$

dargestellt; wir setzen

$$\bar{x} := \tilde{\tilde{x}}_o + A^+b,\ e_o := \bar{x} - x_o$$

und benutzen, daß mit $Ae_o=AA^+(\tilde{b}+\tilde{\tilde{b}})-Ax_o=\tilde{b}-Ax_o$

(10.2) $$r_o = b - Ax_o = \tilde{\tilde{b}} + Ae_o$$

gilt. e_o läßt sich wiederum nach Eigenvektoren von A entwickeln, wegen $e_o=A^+b-\tilde{x}_o\in R(A)$ kann dabei kein Eigenvektor zum Eigenwert 0 auftreten:

(10.3) $$e_o = \sum_{j=1}^{m} \rho_j z_j, \quad \rho_1,\ldots,\rho_m \neq 0,$$

mit orthonormalen Eigenvektoren $z_1,\ldots,z_m$, deren zugehörige Eigenwerte

(10.4) $$0 \neq \lambda_j \neq \lambda_k, \quad j,k = 1,\ldots,m, \; j \neq k,$$

erfüllen. Schließlich bezeichnen wir für $k \geq 1$ noch

$$S_k := [r_o, Ar_o, \ldots, A^{k-1} r_o],$$

$$\overline{S}_k := [Ar_o, A^2 r_o, \ldots, A^k r_o].$$

Die Größen des OD- bzw. MCR-Verfahrens besitzen folgende Eigenschaften:

(10.5) Satz: a) m ist der erste Index mit $Ap_m = 0$; zugleich ist m die kleinste ganze Zahl mit der Eigenschaft:

$Ar_o, A^2 r_o, \ldots, A^{m+1} r_o$ sind linear abhängig.

b) $p_j^T A^2 p_k = 0$ für $0 \leq j < k \leq m$.

c) $[Ap_o, Ap_1, \ldots, Ap_{k-1}] = \overline{S}_k$ für $1 \leq k \leq m$.

d) $\overline{S}_m = [z_1, z_2, \ldots, z_m]$.

e) $p_o, p_1, \ldots, p_{m-1}$ sind linear unabhängig, und es gilt:

$[p_o, p_1, \ldots, p_{k-1}] = S_k$ für $1 \leq k \leq m$,

$S_m \subseteq [\tilde{b}] \oplus [z_1, z_2, \ldots, z_m]$.

f) $(r_k^{OD})^T p_j = (r_k^{MCR})^T A p_j = 0$ für $0 \leq j < k \leq m$.

g) $Ar_m^{MCR} = 0$: x_m^{MCR} löst $A^2 x = Ab$.

h) Es sind äquivalent:

(1) $\tilde{b} = 0$, d.h. $Ax = b$ ist konsistent

(2) $p_m = 0$

(3) $r_m^{OD} = 0$: x_m^{OD} löst $Ax = b$.

i) Für $1 \leq k \leq m$ gilt

$$x_k^{OD} = \arg\min_{x \in x_o + \bar{S}_k} \|x - x_m^{OD}\|, \quad x_k^{MCR} = \arg\min_{x \in x_o + S_k} \|b - Ax\|.$$

j) Falls $\tilde{b} = 0$, gilt $x_m^{OD} = x_m^{MCR} = \bar{x} = \tilde{x}_o + A^+ b$.

Beweis: Die Vektoren Ap_k werden gerade durch eine Lanczos-Rekursion der Form (1.4) erzeugt, wenn man diese mit $f=Ar_o$ startet. a)-d) ergeben sich damit unmittelbar aus Satz(1.5), wenn mit (10.2)-(10.4)

$$Ar_o = A^2 e_o = \sum_{j=1}^{m} \lambda_j^2 p_j z_j, \quad \lambda_j^2 p_j \neq 0 \quad \text{für } j = 1,\ldots,m,$$

beachtet wird.
zu e) Die Inklusion $[p_o, p_1, \ldots, p_{k-1}] \subseteq S_k$ ist offensichtlich, für $1 \leq k \leq m$ folgt Gleichheit, denn nach a) und b) sind $Ap_o, Ap_1, \ldots, Ap_{m-1}$ und deshalb auch $p_o, p_1, \ldots, p_{m-1}$ linear unabhängig.

$$S_m \subseteq [\tilde{b}] \oplus [z_1, z_2, \ldots, z_m]$$

resultiert aus (10.2) und (10.3).
Um beim Beweis von f)-i) Schreibarbeit zu sparen, führen wir eine Matrix H ein und setzen $G := I + A - H$; für $H=I$ ergibt sich dann die Aussage für den OD-Algorithmus, für $H=A$ die entsprechende für den MCR-Algorithmus.
zu f) Induktion nach k:
Für k=0 ist nichts zu zeigen; sei die Behauptung bereits bewiesen für ein $k<m$. Es folgt

$$r_{k+1}^T H p_j = (r_k - \alpha_k G A p_k)^T H p_j = r_k^T H p_j - \alpha_k p_k^T A^2 p_j = 0$$

nach Induktionsannahme und b) für $0 \leq j < k$, nach Definition von α_k für $j=k$.
zu h) Die Implikation "(3) $\Rightarrow$ (1)" ist trivial.
"(1) $\Rightarrow$ (2)" Sei $\tilde{b}=0$, d.h. $b \in R(A)$. Einerseits gilt wegen $r_o = b - Ax_o \in R(A)$ und e)

$$p_m = Ap_{m-1} - \gamma_{m-1} p_{m-1} - \delta_{m-1} p_{m-2} \in [Ap_{m-1}, r_o, Ar_o, \ldots, A^{m-1} r_o] \subseteq R(A),$$

andererseits nach a) $Ap_m = 0$, d.h. $p_m \in N(A) = R(A)^{\perp}$, somit also $p_m = 0$.

Anstelle von "(2) $\Rightarrow$(3)" beweisen wir "$Hp_m=0 \Rightarrow Hr_m=0$", da stets $Ap_m=0$, ist dann auch gleich g) erledigt.

Wir nehmen also $Hp_m=Ap_m=0$ an; es folgt mit $Ap_j=p_{j+1}+\gamma_j p_j+\delta_j p_{j-1}$, $0\leq j<m$,

$$r_m = r_o - \sum_{j=0}^{m-1} \alpha_j GAp_j \in [r_o, GAp_o, GAp_1, \ldots, GAp_{m-1}]$$

$$\subseteq [p_o, Ap_o, Ap_1, \ldots, Ap_{m-1}, \underbrace{Ap_m}_{=0}]$$

$$\subseteq [p_o, Hp_o, Hp_1, \ldots, Hp_{m-1}, \underbrace{Hp_m}_{=0}]$$

und

$$H^2 r_m \in [Hp_o, Hp_1, \ldots, Hp_{m-1}],$$

mit f) schließlich $\|Hr_m\|^2=r_m^T H^2 r_m=0$.

zu i) Wegen $x_m=x_o+\sum_{j=0}^{m-1} \alpha_j Gp_j$ und b) ergibt sich

$$\|H(x_m - (x_o+\sum_{j=0}^{k-1} \xi_j Gp_j))\|^2 = \sum_{j=0}^{k-1} (\alpha_j-\xi_j)^2 p_j^T A^2 p_j + \sum_{j=k}^{m-1} \alpha_j^2 p_j^T A^2 p_j$$

für $1\leq k\leq m$, $\xi_o, \xi_1, \ldots, \xi_{k-1} \in \mathbb{R}$, also

$$x_k = \arg\min_{x \in x_o+GS_k} \|H(x_m - x)\|.$$

Im Falle des OD-Algorithmus sind wir fertig, im Falle des MCR-Algorithmus ist noch folgendes zu beachten: Wegen $\tilde{b}\in R(A)$, $\tilde{\tilde{b}}\in N(A)$ folgt $\tilde{b}-Ax_m^{MCR} \in R(A)$, mit g) aber $A(\tilde{b}-Ax_m^{MCR})=0$, daher $\tilde{b}-Ax_m^{MCR}\in R(A)\cap N(A)$, d.h.

$$\tilde{b} = Ax_m^{MCR} \tag{10.6}$$

und somit

$$\|b - Ax\|^2 = \|\tilde{\tilde{b}}\|^2 + \|\tilde{b} - Ax\|^2 = \|\tilde{\tilde{b}}\|^2 + \|A(x_m^{MCR} - x)\|^2.$$

zu j) Sei $\tilde{\tilde{b}}=0$. Nach h) bzw. (10.6) sind x_m^{OD} und x_m^{MCR} Lösungen von $Ax=b$; wegen $\tilde{x}_o\in R(A)$, $\overline{S}_m\subseteq R(A)$, $S_m\subseteq R(A)$ ($r_o\in R(A)$!) besitzen beide die Form

(10.7) $$x_m^{OD} = \tilde{\tilde{x}}_o + z_1,\ x_m^{MCR} = \tilde{\tilde{x}}_o + z_2,\ z_1 z_2 \in R(A).$$

Die Lösungsmannigfaltigkeit von $Ax=b$ ist aber gerade $A^+b+N(A)$, wegen $A^+b \in R(A)$, $\tilde{\tilde{x}}_o \in N(A)$ liefert ein Vergleich mit (10.7) $z_1=z_2=A^+b$, also

$$x_m^{OD} = x_m^{MCR} = \tilde{\tilde{x}}_o + A^+b = \bar{x}. \square$$

Wie wir eben gesehen haben, liegt im konsistenten Fall bei Abbruch des OD- bzw. MCR-Algorithmus $x_m^{OD}=x_m^{MCR}=\bar{x}$ vor; gilt jedoch $b \notin R(A)$, so zeigen beide Verfahren ein unterschiedliches Verhalten. Während nämlich der MCR-Algorithmus eine Lösung der Normalgleichungen liefert, ist dies beim OD-Algorithmus im allgemeinen nicht der Fall; wie einfachste Beispiele (siehe (10.9)) belegen, kann $\|Ar_m^{OD}\| / \|Ar_o\|$ beliebig groß werden. Das OD-Verfahren ist daher nur im konsistenten Fall brauchbar, übrigens ebenso wie das CG-Verfahren, welches falls $b \in R(A)$ Näherungen

(10.8) $$x_k^{CG} = \underset{x \in x_o + S_k}{\arg\min} \|x - \bar{x}\|_A$$

erzeugt und nach genau m Iterationen auch mit $x_m^{CG}=\bar{x}$ abbricht (vgl. dazu Elfving [10]).

(10.9) Beispiel: Wir wählen

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},\ x_o = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ b = \begin{bmatrix} 1 \\ \beta \end{bmatrix} \quad \text{mit} \quad \beta \in \mathbb{R} \setminus \{0\},$$

die CG-Suchrichtungen bezeichnen wir mit d_k. Ausgehend von

$$p_o = d_o = r_o = b - Ax_o = b$$

ergibt sich

$$\alpha_o^{MCR} = 1,\ x_1^{MCR} = \begin{bmatrix} 1 \\ \beta \end{bmatrix},\ r_1^{MCR} = \begin{bmatrix} 0 \\ \beta \end{bmatrix},\ Ar_1^{MCR} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$\alpha_o^{OD} = \alpha_o^{CG} = 1+\beta^2,\ x_1^{OD} = (1+\beta^2)\begin{bmatrix} 1 \\ 0 \end{bmatrix},\ x_1^{CG} = (1+\beta^2)\begin{bmatrix} 1 \\ \beta \end{bmatrix},$$

$$r_1^{OD} = r_1^{CG} = \beta\begin{bmatrix} -\beta \\ 1 \end{bmatrix},\ Ar_1^{OD} = Ar_1^{CG} = \begin{bmatrix} -\beta^2 \\ 0 \end{bmatrix},$$

sowie

$$p_1 = \begin{bmatrix} 0 \\ -\beta \end{bmatrix}, \quad d_1 = \begin{bmatrix} 0 \\ \beta+\beta^3 \end{bmatrix},$$

was wegen

$$Ap_1 = Ad_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

zum Abbruch der drei Verfahren führt. Während der MCR-Algorithmus eine Lösung der Normalgleichungen gefunden hat, gilt

$$\|Ar_1^{OD}\| / \|Ar_o\| = \|Ar_1^{CG}\| / \|Ar_o\| = \beta^2.$$

Es sei darauf hingewiesen, daß sich die Resultate, die wir in Abschnitt 3 für das OD-Verfahren erhalten haben, unmittelbar auf den singulären, aber konsistenten Fall übertragen lassen. Zum Beispiel führt (10.5)i) und (10.3) zu

$$\frac{\|x_k - \bar{x}\|}{\|x_o - \bar{x}\|} \leq \min_{p\in\overline{\Pi}_{k+1}} \max_{1\leq j\leq m} |p(\lambda_j)| \leq \frac{1}{T_{[(k+1)/2]}\left(\frac{\kappa^2+1}{\kappa^2-1}\right)},$$

wobei nun die "Konditionszahl" durch

$$\kappa := \kappa(A) = \max_{1\leq j\leq n} |\lambda_j| \; / \min_{\substack{1\leq j\leq n \\ \lambda_j\neq 0}} |\lambda_j|$$

definiert wird.

In Abschnitt 11 nützen wir aus, daß sich die Minimierungseigenschaften von OD-, CG- und MCR-Verfahren auch mit Hilfe orthogonaler Projektionen formulieren lassen. Dazu treffen wir einige Vereinbarungen, die bis zum Ende des nächsten Abschnitts gültig seien: Bei OD- und CG-Verfahren wird stets Konsistenz, also $b\in R(A)$ vorausgesetzt, ferner wird die Abkürzung $e_k := \bar{x}-x_k$ verwendet. Für das CG-Verfahren nehmen wir A als positiv semidefinit an, A besitzt dann genau eine positiv semidefinite Wurzel $A^{1/2}$, für die $AA^{1/2}=A^{1/2}A$ gilt (siehe etwa [37]); außerdem wird

$$\tilde{r}_o := A^{1/2}r_o,$$

$$\tilde{S}_k := [\tilde{r}_o, A\tilde{r}_o, \ldots, A^{k-1}\tilde{r}_o] = A^{1/2}S_k,$$

$$f_k := A^{1/2}e_k$$

gesetzt. Schließlich bezeichnen wir noch mit P_k, $\tilde{P}_k$, $\overline{P}_k$ jeweils

die Matrix der orthogonalen Projektion des $\mathbb{R}^n$ auf die Krylovunterräume S_k, $\tilde{S}_k$, $\bar{S}_k$.

(10.10) Satz: Für $k = 1,2,\ldots,m$ gilt:

a) $e_k^{OD} = (I - \bar{P}_k)e_o$, $r_k^{MCR} = (I - \bar{P}_k)r_o$,

$f_k^{CG} = (I - \tilde{P}_k)f_o$;

b) $$\frac{\|e_k^{OD}\|}{\|e_o\|} = \sin\sphericalangle(e_o,\bar{S}_k), \quad \frac{\|r_k^{MCR}\|}{\|r_o\|} = \sin\sphericalangle(r_o,\bar{S}_k),$$

$$\frac{\|f_k^{CG}\|}{\|f_o\|} = \frac{\|e_k^{CG}\|_A}{\|e_o\|_A} = \sin\sphericalangle(f_o,\tilde{S}_k).$$

Beweis: b) folgt mit (1.2) aus a).
zu a) Wir benötigen die Minimierungseigenschaften (10.5)i) und (10.8) der drei Verfahren. Die beiden Probleme

$$\min_{x\in x_o+\bar{S}_k} \|x - \bar{x}\| \qquad \text{und} \qquad \min_{w\in\bar{S}_k} \|e_o - w\|$$

besitzen jeweils eindeutig bestimmte Lösungen $x_k=x_k^{OD}$ bzw. $w_k=\bar{P}_k e_o$, die durch $w_k=x_k-x_o$ zusammenhängen, also $e_k^{OD}=e_o-w_k=(I-\bar{P}_k)e_o$. Entsprechend gilt für die Lösungen $x_k=x_k^{MCR}$ bzw. $w_k=\bar{P}_k r_o$ von

$$\min_{x\in x_o+S_k} \|b - Ax\| \qquad \text{und} \qquad \min_{w\in\bar{S}_k} \|r_o - w\|$$

$w_k=A(x_k-x_o)$ und daher $r_k^{MCR}=r_o-w_k=(I-\bar{P}_k)r_o$. Analog betrachtet man

$$\min_{x\in x_o+S_k} \|x - \bar{x}\|_A \qquad \text{und} \qquad \min_{w\in\tilde{S}_k} \|f_o - w\|,$$

nutzt $w_k=A^{1/2}(x_k-x_o)$ aus und erhält $f_k^{CG}=f_o-w_k=(I-\tilde{P}_k)f_o$. □

11. Schranken für die Fehlerkomponenten längs einzelner Eigenvektoren

Wegen Satz(10.10) stellen e_k, f_k bzw. r_k die "natürlichen" Fehler von OD-, CG- bzw. MCR-Verfahren dar, wobei die Startfehler nach (10.3), (10.2) die Entwicklungen

$$e_o = \sum_{j=1}^{m} \rho_j z_j, \quad f_o = \sum_{j=1}^{m} \rho_j \sqrt{\lambda_j}\, z_j, \quad r_o = \tilde{\tilde{b}} + \sum_{j=1}^{m} \rho_j \lambda_j z_j$$

aufweisen. Mit (10.5)d)e) erkennt man, daß für $k=1,\ldots,m$

$$\bar{S}_k \subseteq \bar{S}_m = [z_1, z_2, \ldots, z_m] =: Z_m$$

und falls $\tilde{b}=0$

$$\tilde{S}_k \subseteq \tilde{S}_m = A^{1/2} S_m = Z_m$$

gilt, damit führt (10.10)a) zu

$$e_k^{OD} \in Z_m, \quad f_k^{CG} \in Z_m, \quad r_k^{CG} - \tilde{\tilde{b}} \in Z_m, \quad k = 0,1,\ldots,m.$$

Die natürlichen Fehler lassen sich also stets als Linearkombinationen von $z_1, z_2, \ldots, z_m$ darstellen, und wir wollen nun obere Schranken für die Fehlerkomponenten längs der einzelnen Eigenvektoren z_j angeben. O.B.d.A. wird $j=1$ gewählt und

$$z := z_1, \quad \lambda := \lambda_1, \quad \rho := \rho_1$$

gesetzt, ferner definieren wir

(11.1)
$$\bar{\varepsilon}_k := \sin\sphericalangle(z, \bar{S}_k) = \|(I - \bar{P}_k)z\|,$$

$$\tilde{\varepsilon}_k := \sin\sphericalangle(z, \tilde{S}_k) = \|(I - \tilde{P}_k)z\|,$$

$$\varepsilon_k := \sin\sphericalangle(z, S_k) = \|(I - P_k)z\|.$$

(11.2) Lemma: Sei $U \subseteq \mathbb{R}^n$ ein linearer Unterraum, P die Matrix der orthogonalen Projektion auf U und $f \in \mathbb{R}^n$.
Dann gilt:
a) $|z^T(I - P)f| \leq \|(I - P)z\| \cdot \|(I - P)f\|$;

$$b)\ \|(I - P)f\| \leq |\eta| \cdot \|(I - P)z\| + \|\hat{f}\|,$$

$$\text{wobei } \eta := f^T z,\ \hat{f} := f - \eta z.$$

Beweis: P besitzt als orthogonale Projektion die Eigenschaften

$$(I - P)^T = I - P,\ (I - P)^2 = I - P,\ \|I - P\| \leq 1;$$

damit folgt

$$z^T(I - P)f = ((I - P)z)^T(I - P)f,$$

$$|z^T(I - P)f| \leq \|(I - P)z\| \cdot \|(I - P)f\|$$

und

$$\begin{aligned}\|(I-P)(\eta z+\hat{f})\|^2 &= \eta^2\|(I-P)z\|^2 + 2\eta((I-P)z)^T\hat{f} + \|(I-P)\hat{f}\|^2 \\ &\leq \eta^2\|(I-P)z\|^2 + 2|\eta| \cdot \|(I-P)z\| \cdot \|\hat{f}\| + \|\hat{f}\|^2 \\ &= (|\eta| \cdot \|(I-P)z\| + \|\hat{f}\|)^2. \square\end{aligned}$$

Dieses Lemma und Satz(10.10)a) implizieren folgendes

(11.3) Korollar: Für k = 1,2,...,m gilt:

$$a)\ |z^T e_k^{OD}| \leq \bar{\varepsilon}_k \|e_k^{OD}\|,$$

$$|z^T f_k^{CG}| \leq \tilde{\varepsilon}_k \|f_k^{CG}\| = \tilde{\varepsilon}_k \|e_k^{CG}\|_A,$$

$$|z^T r_k^{MCR}| \leq \bar{\varepsilon}_k \|r_k^{MCR}\|;$$

$$b)\ \|e_k^{CG}\|_A \leq |\rho|\sqrt{\lambda}\,\tilde{\varepsilon}_k + \Big(\sum_{j=2}^{m} \rho_j^2 \lambda_j\Big)^{1/2}.$$

In Verbindung mit (11.1) besagen die Abschätzungen (11.3)a), daß die Fehlerkomponente längs z relativ zur euklidischen Norm des natürlichen Fehlers mit wachsendem k in dem Maße abklingt, wie $\bar{S}_k$ bzw. $\tilde{S}_k$ eine gute Näherung für z enthält, d.h. wie schnell $\sphericalangle(z,\bar{S}_k)$ bzw. $\sphericalangle(z,\tilde{S}_k)$ gegen Null geht. Ein rasches Abklingen ist zum Beispiel gewährleistet, wenn λ gegenüber den restlichen Eigenwerten $\lambda_2,\ldots,\lambda_m$ isoliert liegt. Dies kann man aus dem folgenden Lemma ersehen, in welchem obere Schranken für $\tan\sphericalangle(z,U_k)$ angegeben werden, wobei wir etwas allgemeiner Krylovunterräume

$U_k := [f, Af, \ldots, A^{k-1}f]$ mit einem Anfangsvektor

$$f = \eta z + \sum_{j=2}^{m} \eta_j z_j \ , \ \eta \neq 0, \tag{11.4}$$

zulassen wollen. Die Situation, daß λ zu den größten bzw. kleinsten Eigenwerten unter den $\lambda_1, \ldots, \lambda_m$ gehört, wird in (11.5)a) behandelt; dieser Teil des Lemmas stammt von Saad [29], der übrigens die Winkel $\sphericalangle(z, U_k)$ benutzte, um Aussagen über das Konvergenzverhalten des Lanczos-Algorithmus zur Bestimmung der Eigenwerte einer symmetrischen Matrix zu gewinnen (siehe auch Parlett [27]). In Teil b) geben wir eine allgemeinere Abschätzung an, die sich im Prinzip bei beliebiger Anordnung der Eigenwerte anwenden läßt, aber natürlich hauptsächlich auf den Fall zielt, daß fast alle Eigenwerte in zwei relativ kleinen, möglichst weit voneinander entfernten Intervallen liegen, während λ und eventuell einige wenige weitere Eigenwerte zwischen diesen Intervallen anzutreffen sind.

(11.5) <u>Lemma:</u> Sei $l (\ll m)$ eine natürliche Zahl und $Z_l := [z_1, \ldots, z_l]$.

a) Sei $\alpha < \beta$. Falls $\lambda_{l+1}, \ldots, \lambda_m \in [\alpha, \beta]$ und entweder

$$\lambda_j > \beta, \quad j = 1,2,\ldots,l, \quad \text{(Fall I)}$$

oder

$$\lambda_j < \alpha, \quad j = 1,2,\ldots,l, \quad \text{(Fall II)}$$

erfüllt ist, so gilt für $k \geq l$

$$\tan\sphericalangle(z, U_k) \leq \frac{\sin\sphericalangle(f, Z_l)}{\cos\sphericalangle(f, z)} \ \frac{\nu_l}{|T_{k-l}(\frac{\beta+\alpha-2\lambda}{\beta-\alpha})|} \ ,$$

wobei

$$\nu_l := \begin{cases} 1 & \text{falls } l = 1 \\ \prod_{i=2}^{l} \frac{\lambda_i - \alpha}{|\lambda_i - \lambda|} & \text{falls } l > 1, \text{ Fall I} \\ \prod_{i=2}^{l} \frac{\beta - \lambda_i}{|\lambda - \lambda_i|} & \text{falls } l > 1, \text{ Fall II} \end{cases} .$$

b) Sei $\alpha < \beta < \gamma < \delta$ und $\beta - \alpha = \delta - \gamma$. Falls

$$\beta < \lambda_j < \gamma, \ j = 1,2,\ldots,l,$$

$$\lambda_{l+1},\ldots,\lambda_m \in M := [\alpha,\beta] \cup [\gamma,\delta]$$

erfüllt ist, so gilt für $k \geq l$

$$\tan\sphericalangle(z,U_k) \leq \frac{\sin\sphericalangle(f,Z_l)}{\cos\sphericalangle(f,z)} \frac{\nu_l}{|T_{[(k-l)/2]}(\tau)|} ,$$

wobei

$$\nu_l := \begin{cases} 1 & \text{falls } l = 1 \\ \max\limits_{l+1\leq j\leq m} \prod\limits_{i=2}^{l} \left|\frac{\lambda_j-\lambda_i}{\lambda-\lambda_i}\right| & \text{falls } l > 1 \end{cases} ,$$

$$\tau := \frac{(\alpha-\mu)^2 + (\beta-\mu)^2 - 2(\lambda-\mu)^2}{(\alpha-\mu)^2 - (\beta-\mu)^2} ,$$

$$\mu := \frac{\beta+\gamma}{2} .$$

Bemerkung: Natürlich gilt

$$\frac{\sin\sphericalangle(f,Z_l)}{\cos\sphericalangle(f,z)} \leq \tan\sphericalangle(f,z) = \tan\sphericalangle(z,U_l).$$

Beweis: Sei $k \geq l$ und p ein reelles Polynom, daß

(11.6) $\quad \operatorname{grad} p \leq k-1$, $p(\lambda) \neq 0$ und für $2 \leq j \leq l$ $p(\lambda_j) = 0$

erfülle. Dann folgt mit (11.4)

$$p(A)f = \eta p(\lambda) z + \sum_{j=l+1}^{m} \eta_j p(\lambda_j) z_j$$

und daraus wegen der Orthonormalität der z_j

$$z^T p(A)f = \eta p(\lambda), \quad \|p(A)f\|^2 = \eta^2 p(\lambda)^2 + \sum_{j=l+1}^{m} \eta_j^2 p(\lambda_j)^2 .$$

Wir beachten $\eta p(\lambda) \neq 0$, sowie (da $\operatorname{grad} p \leq k-1$) $p(A)f \in U_k$ und erhalten mit (1.1)

$$\tan^2\sphericalangle(z,U_k) = \min_{u\in U_k\setminus\{0\}} \tan^2\sphericalangle(z,u) \leq \tan^2\sphericalangle(z,p(A)f)$$

$$= \frac{\|p(A)f\|^2 - (z^T p(A)f)^2}{(z^T p(A)f)^2} = \sum_{j=l+1}^{m} \left(\frac{\eta_j p(\lambda_j)}{\eta p(\lambda)}\right)^2 \leq \sum_{j=l+1}^{m} \left(\frac{\eta_j}{\eta}\right)^2 \max_{l+1\leq j\leq m} \left(\frac{p(\lambda_j)}{p(\lambda)}\right)^2 .$$

Wegen (11.4) und (1.2) gilt aber

$$\cos^2\sphericalangle(f,z) = \frac{\eta^2}{\|f\|^2} \quad , \quad \sin^2\sphericalangle(f,Z_l) = \frac{1}{\|f\|^2} \sum_{j=l+1}^{m} \eta_j^2 ,$$

also ist

$$(11.7) \quad \tan\sphericalangle(z,U_k) \leq \frac{\sin\sphericalangle(f,Z_l)}{\cos\sphericalangle(f,z)} \max_{l+1\leq j\leq m} \frac{|p(\lambda_j)|}{|p(\lambda)|}$$

gezeigt. Die Aussagen des Lemmas ergeben sich nun aus (11.7) durch Einsetzen spezieller Polynome, nämlich Produkte aus

$$r(t) \equiv \begin{cases} 1 & \text{falls } l = 1 \\ \prod_{i=2}^{l} (t-\lambda_i) & \text{falls } l > 1 \end{cases}$$

und geeignet transformierten Tschebyscheffpolynomen.

zu a) Das Polynom

$$p(t) \equiv r(t)\, T_{k-l}(s(t)), \text{ wobei } s(t) \equiv \frac{\beta + \alpha - 2t}{\beta - \alpha} ,$$

erfüllt (11.6) (man beachte, daß $r(\lambda)\neq 0$ und wegen $|s(\lambda)|>1$ $T_{k-l}(s(\lambda))\neq 0$ gilt); mit $s([\alpha,\beta])=[-1,1]$ folgt $|T_{k-l}(s(\lambda_j))|\leq 1$, $l+1\leq j\leq m$, daher

$$\max_{l+1\leq j\leq m} |p(\lambda_j)| \leq \max_{l+1\leq j\leq m} |r(\lambda_j)| = \begin{cases} 1 & \text{falls } l=1 \\ \prod_{i=2}^{l} (\lambda_i-\alpha) & \text{falls } l>1, \text{ Fall I} \\ \prod_{i=2}^{l} (\beta-\lambda_i) & \text{falls } l>1, \text{ Fall II} \end{cases}$$

und mit (11.7) die Behauptung.

zu b) Wir wählen nun

$$p(t) \equiv r(t)\, T_{[(k-1)/2]}(h(t)),$$

wobei

$$h(t) \equiv \frac{q(\alpha) + q(\beta) - 2q(t)}{q(\alpha) - q(\beta)} , \quad q(t) \equiv (t - \mu)^2 .$$

Für die Parabel q gilt wegen $\mu = \frac{\beta+\gamma}{2} = \frac{\alpha+\delta}{2}$, $\beta<\lambda<\gamma$

$$q(\alpha) = q(\delta) > q(\beta) = q(\gamma) > q(\lambda),$$

sowie

$$q(\alpha) \geq q(t) \geq q(\beta) \ , \quad t \in M;$$

damit folgt einerseits $h(\tau)>1$ (also $p(\lambda)\neq 0$, (1.6) ist erfüllt), andererseits $h(M)=[-1,1]$ und

$$|T_{[(k-1)/2]}(h(\lambda_j))| \leq 1 \quad \text{für} \quad l+1\leq j\leq m.$$

Zusammen mit

$$\max_{l+1\leq j\leq m} |p(\lambda_j)| \leq \max_{l+1\leq j\leq m} |r(\lambda_j)| = \begin{cases} 1 & \text{falls } l=1 \\ \max\limits_{l+1\leq j\leq m} \prod\limits_{i=2}^{l} |\lambda_1-\lambda_i| & \text{falls } l>1 \end{cases}$$

liefert (11.7) die gewünschte Abschätzung. □

Im Rest dieses Abschnitts beschäftigen wir uns nur noch mit dem CG-Verfahren, der Index CG kann also weggelassen werden; die Matrix A ist stets positiv semidefinit und mit (10.4) gilt

$$\lambda = \lambda_1 > 0, \text{ sowie o.B.d.A. } 0 < \lambda_2 < \lambda_3 < \cdots \lambda_m.$$

(11.8) Lemma: Sei $U \subseteq [z_1,z_2,\ldots,z_m]$ ein linearer Unterraum mit $z \notin U^{\perp}$. Es gilt:

a) $$\sqrt{\frac{\lambda_2}{\lambda}} \tan\sphericalangle(z,U) \leq \tan\sphericalangle(z,A^{1/2}U) \leq \sqrt{\frac{\lambda_m}{\lambda}} \tan\sphericalangle(z,U);$$

b) $$\min\{1,\sqrt{\tfrac{\lambda_2}{\lambda}}\} \sin\sphericalangle(z,U) \leq \sin\sphericalangle(z,A^{1/2}U) \leq \max\{1,\sqrt{\tfrac{\lambda_m}{\lambda}}\} \sin\sphericalangle(z,U);$$

c) $$\min\{1,\sqrt{\tfrac{\lambda}{\lambda_m}}\} \cos\sphericalangle(z,U) \leq \cos\sphericalangle(z,A^{1/2}U) \leq \max\{1,\sqrt{\tfrac{\lambda}{\lambda_2}}\} \cos\sphericalangle(z,U).$$

Beweis: Sei $u\in U$ mit $u^Tz\neq 0$; wegen der Orthonormalität von $z_1,\ldots,z_m$ ergibt sich

$$u = (u^Tz)z + \sum_{j=2}^{m}(u^Tz_j)z_j,$$

$$\tan^2\sphericalangle(z,u) = \frac{\|u\|^2 - (u^Tz)^2}{(u^Tz)^2} = \frac{1}{(u^Tz)^2} \sum_{j=2}^{m} (u^Tz_j)^2 \ ,$$

sowie

$$A^{1/2}u = \sqrt{\lambda}(u^Tz)z + \sum_{j=2}^{m} \sqrt{\lambda_j}\,(u^Tz_j)z_j,\quad (A^{1/2}u)^Tz = \sqrt{\lambda}\,u^Tz \neq 0,$$

$$\tan^2\!<\!(z,A^{1/2}u) = \frac{\|A^{1/2}u\|^2 - \lambda(u^Tz)^2}{\lambda(u^Tz)^2} = \frac{1}{\lambda(u^Tz)^2}\sum_{j=2}^{m}\lambda_j(u^Tz_j)^2 .$$

Es folgt

$$\sqrt{\frac{\lambda_2}{\lambda}}\tan\!<\!(z,u) \leqslant \tan\!<\!(z,A^{1/2}u) \leqslant \sqrt{\frac{\lambda_m}{\lambda}}\tan\!<\!(z,u)$$

und mit - man beachte $z \notin U^{\perp}$ -

$$\tan\!<\!(z,U) = \min_{u\in U\setminus\{0\}} \tan\!<\!(z,u) = \min_{\substack{u\in U\\ u^Tz\neq 0}} \tan\!<\!(z,u),$$

$$\tan\!<\!(z,A^{1/2}U) = \min_{u\in U\setminus\{0\}} \tan\!<\!(z,A^{1/2}u) = \min_{\substack{u\in U\\ u^Tz\neq 0}} \tan\!<\!(z,A^{1/2}u)$$

die Behauptung a). Mit

$$\sin^2\phi = \frac{\tan^2\phi}{1+\tan^2\phi} \qquad \text{bzw. } \cos^2\phi = \frac{1}{1+\tan^2\phi}$$

und unter Berücksichtigung, daß für $\alpha>0$, $t\geqslant 0$

$$\min\{1,\alpha\}\,\frac{t}{1+t} \leqslant \frac{\alpha t}{1+\alpha t} \leqslant \max\{1,\alpha\}\,\frac{t}{1+t}$$

bzw.

$$\min\{1,\tfrac{1}{\alpha}\}\,\frac{1}{1+t} \leqslant \frac{1}{1+\alpha t} \leqslant \max\{1,\tfrac{1}{\alpha}\}\,\frac{1}{1+t}$$

gilt, erhalten wir die Abschätzungen b) bzw. c) unmittelbar aus a). □

Speziell für $U=S_k$ ($z^Tr_o=\rho\lambda\neq 0$, also $z\notin S_k^{\perp}$) liefert (11.8)b) folgendes

(11.9) Korollar: Für $k=1,2,\ldots,m$ gilt:

$$\min\{1,\sqrt{\frac{\lambda_2}{\lambda}}\}\,\varepsilon_k \leqslant \tilde{\varepsilon}_k \leqslant \max\{1,\sqrt{\frac{\lambda_m}{\lambda}}\}\,\varepsilon_k \leqslant \sqrt{\frac{\|A\|}{\lambda}}\,\varepsilon_k .$$

Aus (11.3) und (11.9), sowie mit $z^Tf_k=z^TA^{1/2}e_k=\sqrt{\lambda}z^Te_k$ folgt nun für $k=1,2,\ldots,m$ die Ungleichungskette

$$(11.10)\quad |z^T e_k| \leq \frac{1}{\sqrt{\lambda}}\,\tilde{\varepsilon}_k \|e_k\|_A \leq \tilde{\varepsilon}_k\Big(|\rho|\tilde{\varepsilon}_k + \Big(\sum_{j=2}^{m} \rho_j^2 \frac{\lambda_j}{\lambda}\Big)^{1/2}\Big)$$

$$\leq \tilde{\varepsilon}_k\Big(|\rho|\tilde{\varepsilon}_k + \sqrt{\frac{\lambda_m}{\lambda}}\,\tau\Big) \leq \varepsilon_k\Big(|\rho|\varepsilon_k + \tau\Big)\frac{\|A\|}{\lambda},$$

dabei wurde $\tau := \Big(\sum_{j=2}^{m} \rho_j^2\Big)^{1/2}$ gesetzt.

Wir wollen jetzt unsere Abschätzung (11.10) mit der Schranke vergleichen, die Stewart [31] für das CG-Verfahren herleitete. Es sei zunächst daran erinnert, daß wir im Krylovunterraum $\tilde{S}_k$ operierten und und so die Darstellung $f_k = (I - \tilde{P}_k) f_o$, $\tilde{P}_k$ eine orthogonale Projektion, ausnutzen konnten. Seien die CG-Suchrichtungen mit d_j bezeichnet und $D_k := (d_o, d_1, \ldots, d_{k-1})$, so geht Stewart dagegen von

$$e_k = (I - \hat{P}_k) e_o, \quad \text{wobei } \hat{P}_k := D_k (D_k^T A D_k)^{-1} D_k^T A,$$

aus. Nun ist jedoch $\hat{P}_k$ die Matrix einer Schrägprojektion auf S_k, und Stewart muß relativ komplizierte Abschätzungen ausführen, um zu folgendem Resultat zu gelangen:

Falls $\lambda - 2\varepsilon_k^2 \|A\|(1 + 2\chi_k) > 0$, so gilt

$$(11.11)\quad |z^T e_k| \leq 2\varepsilon_k\Big(|\rho|\varepsilon_k + \tau\Big)\Big(1 + \sqrt{\chi_k}\Big)\Big(1 + \frac{\|A\|}{\lambda - 2\varepsilon_k^2 \|A\|(1+2\chi_k)}\Big).$$

Dabei ist $\chi_k := \|A\| \cdot \|(U_k^T A U_k)^{-1}\|$ mit einer $n \times (k-1)$-Matrix U_k, die so gewählt wird, daß die Spalten von U_k zusammen mit $P_k z / \|P_k z\|$ eine Orthonormalbasis von S_k bilden. Vergleich des letzten Gliedes der Ungleichungskette (11.10) mit (11.11) zeigt, daß unsere Schranken besser sind (numerische Beispiele dazu findet man in Abschnitt 22).

II. VERFAHREN FÜR GLEICHUNGSSYSTEME MIT UNSYMMETRISCHER MATRIX

12. CG-Verfahren angewandt auf die Normalgleichungen.

Der Algorithmus von Craig

Eine naheliegende Möglichkeit, zu Verfahren für lineare Gleichungssysteme mit unsymmetrischer Matrix A zu gelangen, besteht darin, daß man zu Systemen mit A^TA bzw. AA^T als Koeffizientenmatrix übergeht und darauf die Algorithmen für den symmetrischen Fall anwendet. So leiten sich aus dem CG-Verfahren zwei Algorithmen ab, die nun kurz beschrieben werden sollen; wir brauchen uns nicht auf quadratische A zu beschränken und setzen A als reelle p×n-Matrix, sowie $b \in \mathbb{R}^p$ voraus. Das Gleichungssystem

$$Ax = b \tag{12.1}$$

besitzt nicht notwendig eine Lösung, wohl aber die Normalgleichungen

$$A^TAx = A^Tb,$$

denn diese sind wegen $A^Tb \in R(A^T)=R(A^TA)$ stets konsistent. Die Matrix A^TA ist positiv semidefinit, und Hestenes und Stiefel [16] schlugen vor, das CG-Verfahren auf die Normalgleichungen anzuwenden:

(12.2) CGN-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$r_o = b - Ax_o,\ d_o = A^Tr_o.$$

Für $k = 0,1,2,\ldots$

1) Falls $Ad_k = 0$: stop, x_k löst $A^TAx = A^Tb$.

Sonst setze

2) $\alpha_k = r_k^TAA^Tr_k/d_k^TA^TAd_k$,

$$x_{k+1} = x_k + \alpha_k d_k,\ r_{k+1} = r_k - \alpha_k Ad_k,$$

3) $\beta_k = r_{k+1}^T AA^T r_{k+1} / r_k^T AA^T r_k$,

$d_{k+1} = A^T r_{k+1} + \beta_k d_k$.

Der Startwert x_o wird in der Form

(12.3) $x_o = \tilde{x}_o + \tilde{\tilde{x}}_o$, $\tilde{x}_o \in R(A^TA) = R(A^T)$, $\tilde{\tilde{x}}_o \in N(A^TA) = N(A)$,

zerlegt; wir setzen ferner

$$S_k^2 := [A^T r_o, (A^TA)A^T r_o, (A^TA)^2 A^T r_o, \ldots, (A^TA)^{k-1} A^T r_o]$$

und bezeichnen mit $0<\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_r$ die von Null verschiedenen singulären Werte von A, d.h. die σ_j^2, $j=1,\ldots,r$, sind die von Null verschiedenen Eigenwerte von A^TA.

(12.4) Satz: Für die Größen des CGN-Algorithmus gilt:

a) Es gibt einen ersten Index l mit $d_l = 0$;

b) $\|A^T r_k\| > 0$, $\|Ad_k\| > 0$ für $0 \leq k < l$;

c) $A^T r_l = 0$: $x_l = \bar{x} := \tilde{\tilde{x}}_o + A^+ b$ löst $A^TAx = A^Tb$;

d) $x_k = \arg\min_{x \in x_o + S_k^2} \|x - \bar{x}\|_{A^TA}$ für $1 \leq k \leq l$;

e) $$\frac{\|x_k - \bar{x}\|_{A^TA}}{\|x_o - \bar{x}\|_{A^TA}} \leq \min_{p \in \Pi_k} \max_{1 \leq j \leq r} |p(\sigma_j^2)| \leq \left(T_k\left(\frac{\left(\frac{\sigma_r}{\sigma_1}\right)^2 + 1}{\left(\frac{\sigma_r}{\sigma_1}\right)^2 - 1}\right)\right)^{-1},$$

für $0 \leq k \leq l$;

f) $$\frac{r_k^T AA^T AA^T r_k}{r_k^T AA^T r_k} = \frac{1}{\alpha_k} + \frac{\beta_{k-1}}{\alpha_{k-1}} \quad \text{für } 1 \leq k < l;$$

g) die umnormierten Residuen

$$q_o := r_o, \quad q_k := \frac{1}{\alpha_o \alpha_1 \cdots \alpha_{k-1}} r_k \quad \text{für } 1 \leq k \leq l,$$

gehorchen der Rekursion

(12.5) $$q_{k+1} = -(AA^T q_k - \gamma_k q_k + \delta_k q_{k-1}) \quad , \quad 0 \leq k < l,$$

wobei

$$\gamma_k = \frac{q_k^T AA^T AA^T q_k}{q_k^T AA^T q_k} \quad , \quad \delta_k = \begin{cases} 0 & \text{für } k=0 \\ \dfrac{q_k^T AA^T q_k}{q_{k-1}^T AA^T q_{k-1}} & \text{für } k>0 \end{cases} ;$$

h) l ist die kleinste ganze Zahl mit der Eigenschaft:
$A^T r_o, (A^T A)A^T r_o, (A^T A)^2 A^T r_o, \ldots, (A^T A)^l A^T r_o$ sind linear abhängig;

i) l ist die Anzahl der Eigenräume von $A^T A$, in denen $A^T r_o$ von Null verschiedene Komponenten besitzt.

<u>Beweis:</u> a)-f) ergeben sich unmittelbar aus den entsprechenden Eigenschaften des CG-Verfahrens, siehe etwa [1,10,28].
zu g) Wegen $r_1 = r_o - \alpha_o A d_o$, $d_o = A^T r_o$ folgt $1/\alpha_o = \gamma_o$ und $q_1 = -AA^T q_o + \gamma_o q_o$. Für $1 \leq k < l$ gilt

$$r_{k+1} = r_k - \alpha_k A d_k = r_k - \alpha_k (AA^T r_k + \beta_{k-1} A d_{k-1})$$

$$= -\alpha_k (AA^T r_k - (\frac{1}{\alpha_k} + \frac{\beta_{k-1}}{\alpha_{k-1}}) r_k + \frac{\beta_{k-1}}{\alpha_{k-1}} r_{k-1})$$

und Division durch $\alpha_o \alpha_1 \cdots \alpha_k$ liefert (12.5), wenn man $\delta_k = \beta_{k-1}/\alpha_{k-1}^2$, sowie (nach f)) $\gamma_k = 1/\alpha_k + \beta_{k-1}/\alpha_{k-1}$ beachtet.
zu h)i) Nach b) und c) ist l der erste Index mit $A^T r_l = 0$; wegen g) gehorchen die (geeignet normierten) $A^T r_k$ aber einer Lanczos-Rekursion, welche die Form (1.4) besitzt, wenn man dort als symmetrische Matrix $A^T A$ wählt und $f = A^T r_o$ setzt. h) und i) ergeben sich daher aus Satz(1.5)d)e). □

Für den Rest dieses Abschnitts sei nun vorausgesetzt, daß (12.1) konsistent ist, es gelte also $b \in R(A)$. In diesem Fall kann man zur Lösung von (12.1) ein Verfahren benutzen, welches zuerst von Craig [8] angegeben wurde:

(12.6) Craig-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$r_o = b - Ax_o, \quad A^T d_o = A^T r_o.$$

Für $k = 0,1,2,\ldots$

1) Falls $A^T d_k = 0$: stop, x_k löst $Ax = b$.

Sonst setze

2) $\alpha_k = r_k^T r_k / d_k^T A A^T d_k$,

$x_{k+1} = x_k + \alpha_k A^T d_k$, $r_{k+1} = r_k - A(\alpha_k A^T d_k)$,

3) $\beta_k = r_{k+1}^T r_{k+1} / r_k^T r_k$,

$A^T d_{k+1} = A^T r_{k+1} + \beta_k A^T d_k$.

Faddejew und Faddejewa [12] erkannten, daß dieser Algorithmus in engem Zusammenhang mit dem CG-Verfahren steht. Wegen $b \in R(A) = R(AA^T)$ ist nämlich das Gleichungssystem

(12.7) $$AA^T y = b$$

konsistent, außerdem ist seine Koeffizientenmatrix positiv semidefinit. Wir zerlegen x_o gemäß (12.3) und wählen ein $y_o \in \mathbb{R}^p$ mit $\tilde{x}_o = A^T y_o$; ausgehend von diesem y_o als Startwert liefert das CG-Verfahren, angewandt auf (12.7), Näherungen y_k, zugehörige Residuen r_k und Suchrichtungen d_k, die über $d_o = r_o$, $d_{k+1} = r_{k+1} + \beta_k d_k$ zusammenhängen. Diese r_k, d_k, β_k zusammen mit den

$$x_k := \tilde{\tilde{x}}_o + A^T y_k, \quad k = 0,1,\ldots,$$

sind aber gerade die Größen aus Algorithmus(12.6).

(12.8) Satz: Für das Verfahren von Craig gilt:

a) Es gibt einen ersten Index l mit $d_l = 0$;

b) $\|r_k\| > 0$, $\|A^T d_k\| > 0$ für $0 \leq k < l$;

c) $r_l = 0$: $x_l = \bar{x} := \tilde{\tilde{x}}_o + A^+ b$ löst $Ax = b$;

d) $x_k = \arg\min_{x \in x_o + S_k^2} \|x - \bar{x}\|$ für $1 \leq k \leq l$;

e) $$\frac{\|x_k - \bar{x}\|}{\|x_o - \bar{x}\|} \leq \min_{p \in \Pi_k} \max_{1 \leq j \leq r} |p(\sigma_j^2)| \leq \left(T_k\left(\frac{\left(\frac{\sigma_r}{\sigma_1}\right)^2 + 1}{\left(\frac{\sigma_r}{\sigma_1}\right)^2 - 1}\right)\right)^{-1}$$

für $0 \leq k \leq l$;

f) $d_k^T A A^T d_j = 0$ für $0 \leq k < j \leq l$;

g) $$\beta_k = -\frac{r_{k+1}^T A A^T d_k}{d_k^T A A^T d_k}$$ für $0 \leq k < l$;

h) $$\alpha_k = \frac{r_k^T d_k}{d_k^T A A^T d_k}$$ für $0 \leq k < l$;

i) $$\frac{r_k^T A A^T r_k}{r_k^T r_k} = \frac{1}{\alpha_k} + \frac{\beta_{k-1}}{\alpha_{k-1}}$$ für $1 \leq k < l$;

j) die umnormierten Richtungen

$$q_o := d_o = r_o,\ q_k := \frac{1}{\alpha_o \alpha_1 \cdots \alpha_{k-1}} d_k \quad \text{für } 1 \leq k \leq l,$$

erfüllen die Rekursion (12.5);

k) $x_{k+1} = \hat{\rho}_{k+1}(\hat{\sigma}_{k+1} A^T r_k + x_k) + (1 - \hat{\rho}_{k+1}) x_{k-1}$ für $0 \leq k < l$,

wobei $x_{-1} := 0$, $\hat{\sigma}_{k+1} = r_k^T r_k / r_k^T A A^T r_k$ und

(12.9) $$\hat{\rho}_{k+1} = \begin{cases} 1 & \text{für } k = 0 \\ \left(1 - \frac{\hat{\sigma}_{k+1}}{\hat{\sigma}_k} \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} \frac{1}{\hat{\rho}_k}\right)^{-1} & \text{für } k > 0 \end{cases}.$$

Beweis: a)-i) ergeben sich wiederum aus den entsprechenden Eigenschaften des CG-Verfahrens vermöge des skizzierten Zusammenhangs, wir wollen lediglich auf c) etwas näher eingehen: Der CG-Algorith-

mus liefert die Lösung $y_1=\tilde{\tilde{y}}_o+(AA^T)^+b$ von (12.7), wobei

$$y_o = \tilde{y}_o + \tilde{\tilde{y}}_o,\ \tilde{y}_o \in R(AA^T),\ \tilde{\tilde{y}}_o \in N(AA^T) = N(A^T),$$

im Craig-Algorithmus erhält man daher

$$x_1 = \tilde{\tilde{x}}_o + A^T y_1 = \tilde{\tilde{x}}_o + A^T(AA^T)^+ b = \tilde{\tilde{x}}_o + A^+ b.$$

zu j) Wir vereinbaren $\beta_{-1}:=0$; für $0\leqslant k<1$ folgt

$$d_{k+1} = r_{k+1} + \beta_k d_k = -\alpha_k AA^T d_k + r_k + \beta_k d_k$$

$$= -\alpha_k AA^T d_k + (1 + \beta_k)d_k - \beta_{k-1}d_{k-1}.$$

Division dieser Gleichung durch $\alpha_o\alpha_1\cdots\alpha_k$ ergibt die Behauptung, wenn man beachtet, daß wegen f)

$$r_{k+1}^T AA^T d_k = d_k^T AA^T d_k - \alpha_k d_k^T AA^T AA^T d_k$$

und mit g)

$$\frac{1 + \beta_k}{\alpha_k} = \frac{d_k^T AA^T AA^T d_k}{d_k^T AA^T d_k} = \gamma_k, \quad 0 \leqslant k < 1,$$

sowie nach Definition der β_j, α_j

$$\frac{\beta_{k-1}}{\alpha_{k-1}\alpha_k} = \frac{1}{\alpha_{k-1}^2} \frac{d_k^T AA^T d_k}{d_{k-1}^T AA^T d_{k-1}} = \delta_k, \quad 1 \leqslant k < 1,$$

gilt.

zu k) Wegen $A^T d_o = A^T r_o$,

$$A^T d_k = A^T r_k + \beta_{k-1} A^T d_{k-1} = A^T r_k + \frac{\beta_{k-1}}{\alpha_{k-1}} (x_k - x_{k-1}),\ 1 \leqslant k < 1,$$

folgt für $0\leqslant k<1$

$$x_{k+1} = x_k + \alpha_k A^T d_k = \hat{\rho}_{k+1}(\hat{\sigma}_{k+1} A^T r_k + x_k) + (1 - \hat{\rho}_{k+1}) x_{k-1},$$

wobei $\hat{\sigma}_1=\alpha_o$, $\hat{\rho}_1=1$, sowie für $k>0$

$$\hat{\sigma}_{k+1} = \left(\frac{1}{\alpha_k} + \frac{\beta_{k-1}}{\alpha_{k-1}}\right)^{-1} \left(= \frac{r_k^T r_k}{r_k^T AA^T r_k} \text{ nach i)}\right),$$

$$\hat{\rho}_{k+1} = 1 + \frac{\alpha_k}{\alpha_{k-1}} \beta_{k-1}$$

gesetzt wurde. Mit $\hat{\rho}_{j+1}\hat{\sigma}_{j+1}=\alpha_j$, $j=0,1,\ldots$, erkennt man, daß die $\hat{\rho}_{k+1}$ auch durch (12.9) definiert werden. □

Stimmt der Startwert $x_o=x_o^{CGN}=x_o^{CRAIG}$ der Algorithmen(12.2) und (12.6) überein, so gilt:

(12.10) Korollar: a) Craig- und CGN-Algorithmus brechen nach derselben Zahl l von Iterationen ab und liefern die Lösung $\bar{x} = \tilde{\tilde{x}}_o + A^+ b$ von $Ax = b$;

b)
$$r_k^{CGN} = \frac{\alpha_o^{CGN} \cdots \alpha_{k-1}^{CGN}}{\alpha_o^{CRAIG} \cdots \alpha_{k-1}^{CRAIG}} d_k^{CRAIG} \quad \text{für } 1 \leq k \leq l;$$

c) $\|x_k^{CRAIG} - \bar{x}\| \leq \|x_k^{CGN} - \bar{x}\|$,

$\|b - Ax_k^{CGN}\| \leq \|b - Ax_k^{CRAIG}\|$ für $0 \leq k \leq l$.

Beweis: a)b) ergeben sich aus (12.4)g)c) und (12.8)j)c); c) folgt aus (12.4)d) und (12.8)d). □

Nun sei A als nichtsinguläre $n\times n$-Matrix vorausgesetzt und die Lösung von (12.1) wieder mit $\bar{x}$ bezeichnet; für diesen Fall wurde von Amara und Nedelec† jüngst folgendes Verfahren vorgeschlagen:

(12.11) Algorithmus (Amara und Nedelec):

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$q_{-1} = r_o = b - Ax_o, \quad \hat{p}_o = A^T r_o.$$

Für $k = 0,1,2,\ldots$

1) Falls $\hat{p}_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst setze

2) $\lambda_k = \|\hat{p}_k\|$, $p_k = \hat{p}_k/\lambda_k$,

---

†Amara,M., Nedelec,J.-C.: Résolution de système matriciel indéfini par une décomposition sur une double suite orthogonale. Mitteilung von J.L.Lions.

$$\hat{q}_k = \begin{cases} Ap_k & \text{für } k = 0 \\ Ap_k - \lambda_k q_{k-1} & \text{für } k > 0 \end{cases},$$

$$\mu_k = \|\hat{q}_k\|, \quad q_k = \hat{q}_k/\mu_k,$$

3) $\alpha_k = r_k^T q_{k-1}/\lambda_k,$

$$x_{k+1} = x_k + \alpha_k p_k, \quad r_{k+1} = r_k - \alpha_k A p_k,$$

4) $\hat{p}_{k+1} = A^T q_k - \mu_k p_k.$

Amara und Nedelec zeigten, daß ihr Algorithmus wohldefiniert ist und nach l ($\leq n$) Iterationsschritten mit $x_l = \bar{x}$ abbricht, ferner gilt:

(12.12) $$p_k^T p_j = q_k^T q_j = \delta_{kj} \quad \text{für } 0 \leq k,j < l.$$

Wir vergleichen jetzt die Näherungen x_k^{CRAIG} und x_k^{AN}, welche die Verfahren (12.6) und (12.11) ausgehend vom gemeinsamen Startwert x_o ($\neq \bar{x}$) erzeugen, und gelangen zu folgendem Resultat:

(12.13) Satz: Für $0 \leq k < l$ gilt:

a) $$r_k^T A^{-T} p_j = \begin{cases} \alpha_k & \text{für } j = k \\ 0 & \text{für } 0 \leq j < k \end{cases};$$

b) $$\mu_k \hat{p}_{k+1} = A^T A p_k - (p_k^T A^T A p_k) p_k - \mu_{k-1} \lambda_k p_{k-1}$$

(dabei wurde $\mu_{-1} := 0$, $p_{-1} := 0$ gesetzt).

c) Die Abbruchindizes der Algorithmen (12.6) und (12.11) stimmen überein, und es gilt:

$$x_k^{CRAIG} = x_k^{AN} \quad \text{für } 0 \leq k \leq l.$$

Beweis: zu a) Induktion nach k:

Für k=0 gilt $r_o^T A^{-T} p_o = \alpha_o$ wegen $p_o = A^T q_{-1}/\lambda_o$. Sei $1 \leq k < l$ und die Behauptung bereits bewiesen für k-1; es folgt mit (12.12)

$$r_k^T A^{-T} p_j = r_{k-1}^T A^{-T} p_j - \alpha_{k-1} p_{k-1}^T p_j = 0 \quad \text{für } 0 \leq j < k$$

und weiter

$$r_k^T A^{-T} \hat{p}_k = r_k^T A^{-T}(A^T q_{k-1} - \mu_{k-1} p_{k-1}) = r_k^T q_{k-1} = \alpha_k \lambda_k .$$

zu b) Wir multiplizieren

$$\mu_k q_k = A p_k - \lambda_k q_{k-1} \quad (\text{falls } k > 0) \text{ bzw. } \mu_o q_o = A p_o$$

von links mit A^T, benutzen $A^T q_j = \hat{p}_{j+1} + \mu_j p_j$ und erhalten so eine Rekursion der Gestalt

$$\mu_k \hat{p}_{k+1} = A^T A p_k - \nu_k p_k - \mu_{k-1} \lambda_k p_{k-1}, \quad 0 \leq k < l,$$

dabei gilt $\nu_k = p_k^T A^T A p_k$ wegen (12.12).

zu c) Beachtet man (12.8)h) und setzt

$$s_o = A^T r_o, \quad s_k = \frac{1}{\alpha_o \alpha_1 \cdots \alpha_{k-1}} A^T d_k, \quad 1 \leq k \leq l,$$

so erkennt man, daß die Größen des Craig-Algorithmus

$$(12.14) \quad x_{k+1}^{CRAIG} - x_k^{CRAIG} = \alpha_k A^T d_k = \frac{r_k^T d_k}{d_k^T A A^T d_k} A^T d_k = \frac{r_k^T A^{-T} s_k}{s_k^T s_k} s_k$$

für $0 \leq k < l$ erfüllen. Wegen (12.8)j) gilt für $0 \leq k < l$

$$s_{k+1} = -(A^T A s_k - \gamma_k s_k + \delta_k s_{k-1}),$$

$$\text{wobei } \gamma_k = \frac{s_k^T A^T A s_k}{s_k^T s_k}, \quad \delta_k = \begin{cases} 0 & \text{für } k = 0 \\ \dfrac{s_k^T s_k}{s_{k-1}^T s_{k-1}} & \text{für } k > 0 \end{cases}.$$

Vergleich mit der Rekursion aus Teil b) liefert folgenden Zusammenhang zwischen den Suchrichtungen der Algorithmen(12.6) und (12.11):

$$p_k = (-1)^k \frac{s_k}{\|s_k\|}, \quad \mu_k \|\hat{p}_{k+1}\| = \frac{\|s_{k+1}\|}{\|s_k\|} \qquad \text{für } 0 \leq k < l;$$

insbesondere stimmen die Abbruchindizes beider Verfahren überein und wegen Teil a), sowie (12.14) gilt $x_k^{CRAIG} = x_k^{AN}$ für $0 \leq k \leq l$. □

13. Der STOD-Algorithmus für Matrizen der Form A = S - N

In diesem und den folgenden Abschnitten sei A stets eine unsymmetrische n×n-Matrix, die wir gemäß

$$A = M - N,\ M := (A + A^T)/2 = M^T,\ N := (A^T - A)/2 = -N^T,$$

aufspalten, ferner wird für $k \geq 1$

$$S_k := [r_o, Ar_o, \ldots, A^{k-1}r_o]$$

gesetzt. Versucht man - im Sinne einer Verallgemeinerung des CG- bzw. CR-Verfahrens für positiv definite Matrizen - Näherungen $x_k \in x_o + S_k$ zu berechnen, welche die Galerkin-Bedingung

$$r_k = b - Ax_k \perp S_k \tag{13.1}$$

erfüllen bzw. die Minimierungseigenschaft

$$x_k = \arg \min_{x \in x_o + S_k} \|b - Ax\| \tag{13.2}$$

besitzen, so gelangt man zu Algorithmen, die im allgemeinen in jedem Iterationsschritt sämtliche bisherigen Suchrichtungen benötigen (siehe [2,30,38]). Für Matrizen A mit M=I fanden Concus, Golub [7] und Widlund [36], daß sich die x_k aus (13.1) in einfacher Weise berechnen lassen; ihr Verfahren (siehe (14.7)) basiert auf der Tatsache, daß man in diesem Spezialfall die Krylovsequenz $r_o, Ar_o, A^2r_o, \ldots$ mittels einer Lanczos-Rekursion der Form

$$\alpha_k v_{k+1} = Nv_k - \beta_k v_{k-1},\ v_o = r_o, \tag{13.3}$$

orthogonalisieren kann. Im weiteren werden wir zeigen, daß sich auch MCR- und STOD-Algorithmus auf solche A=I-N übertragen lassen: Ähnlich wie in (13.3) erzeugen wir Vektoren p_k, die einerseits als Suchrichtungen für die Berechnung der x_k aus (13.2) dienen, andererseits verwenden wir die $A^T p_k$ als Suchrichtungen für ein Verfahren mit der Minimierungseigenschaft

$$x_k = \arg \min_{x \in x_o + A^T S_k} \|x - \bar{x}\|.$$

Ganz entsprechend kann man bei schiefsymmetrischem A (also M=0) vorgehen, im folgenden sei daher A von der Gestalt

$$A = S - N,\ S = sI,\ N = -N^T,$$

wobei s=0 oder s=1 zu setzen ist. Man beachte, daß

$$AA^T = A^TA = S - N^2$$

gilt und N mit A, A^T und AA^T vertauschbar ist; ferner wird von

$$y^TN^{2k-1}y = 0,\ y \in \mathbb{R}^n,\ k = 1,2,\ldots,$$

Gebrauch gemacht.
Wir formulieren nun die Rekursion zur Berechnung der p_k (im übrigen sei der triviale Fall $A^Tr_o=0$ stets ausgeschlossen):

(13.4) Lanczos-Algorithmus:

Start: Setze $p_o = r_o$, $p_{-1} = 0$.

Für k = 0,1,2,...:

Falls $A^Tp_k = 0$: stop,

andernfalls setze

$$p_{k+1} = Np_k - \delta_k p_{k-1} \text{ mit } \delta_k = \begin{cases} 0 & \text{für } k=0 \\ -p_k^TAA^Tp_k / p_{k-1}^TAA^Tp_{k-1} & \text{für } k>0 \end{cases}.$$

(13.5) Satz: Für die Größen aus (13.4) gilt:

a) Es gibt einen ersten Index m, $1 \leq m \leq n$ mit $A^Tp_m = 0$;

b) $p_j^TAA^Tp_k = p_j^TA^TAp_k = 0$ für $0 \leq j < k \leq m$;

c) $p_{k+1} = N^2p_k - (\delta_k+\delta_{k+1})p_k - \delta_k\delta_{k-1}p_{k-2}$ für $0 \leq k \leq m-2$,
wobei $p_{-2} := 0$, $\delta_{-1} := 0$;

d) i) $[p_o,p_2,p_4,\ldots,p_{2k}] = [r_o,N^2r_o,N^4r_o,\ldots,N^{2k}r_o]$, $0 \leq k < \frac{m}{2}$;

ii) $[p_1,p_3,p_5,\ldots,p_{2k+1}] = [Nr_o,N^3r_o,N^5r_o,\ldots,N^{2k+1}r_o]$,
$0 \leq k < \frac{m-1}{2}$;

iii) $[p_o,p_1,\ldots,p_{k-1}] = [r_o,Nr_o,\ldots,N^{k-1}r_o] = S_k$, $1 \leq k \leq m$;

e) m ist die kleinste ganze Zahl mit der Eigenschaft:
$A^T r_o, NA^T r_o, N^2 A^T r_o, \ldots, N^m A^T r_o$ sind linear abhängig;

f) m ist die Anzahl der Eigenräume von $N=S-A$, in denen $A^T r_o$ von Null verschiedene Komponenten hat:

$$A^T r_o = \sum_{j=1}^{m} \rho_j z_j , \quad \rho_1, \ldots, \rho_m \in \mathbb{C} \setminus \{0\}, \tag{13.6}$$

dabei sind $z_1, \ldots, z_m$ orthonormale Eigenvektoren zu paarweise verschiedenen Eigenwerten $i\lambda_1, \ldots, i\lambda_m$ ($\in i\mathbb{R}$) von N;

g) i) $\{\lambda_1, \lambda_2, \ldots, \lambda_m\} = \{-\lambda_1, -\lambda_2, \ldots, -\lambda_m\}$;

ii) $l := \left|\{|\lambda_1|, |\lambda_2|, \ldots, |\lambda_m|\}\right| = [(m+1)/2]$ ist die Anzahl der Eigenräume von $A^T A$, in denen $A^T r_o$ von Null verschiedene Komponenten besitzt.

<u>Beweis:</u> zu a)b) Sei $\tilde{m} \geq 1$ mit $A^T p_k \neq 0$, $0 \leq k < \tilde{m}$; per Induktion nach k zeigen wir $p_j^T AA^T p_k = 0$, $0 \leq j < k \leq \tilde{m}$. Für $k=1$ gilt wegen $N=-N^T$ $p_o^T AA^T p_1 = p_o^T (S-N^2) N p_o = 0$. Sei nun $1 \leq k < \tilde{m}$ und die Behauptung bereits bewiesen für alle Indizes $\leq k$. Es folgt

$$p_j^T AA^T p_{k+1} = p_j^T (S-N^2) N p_k - \delta_k p_j^T AA^T p_{k-1} = 0 \quad \text{für} \quad j = k, k-1,$$

dabei wird für $j=k$ die Schiefsymmetrie von N und für $j=k-1$

$$p_{k-1}^T AA^T N p_k = -(Np_{k-1})^T AA^T p_k = -(p_k + \delta_{k-1} p_{k-2})^T AA^T p_k = -p_k^T AA^T p_k$$

ausgenutzt. Für $0 \leq j < k-1$ ergibt sich mit der Induktionsannahme

$$p_j^T AA^T p_{k+1} = p_j^T AA^T (Np_k - \delta_k p_{k-1}) = -(Np_j)^T AA^T p_k$$
$$= -(p_{j+1} + \delta_j p_{j-1})^T AA^T p_k = 0.$$

Insbesondere sind $A^T p_o, A^T p_1, \ldots, A^T p_{\tilde{m}-1}$ linear abhängig, also gibt es einen ersten Index $m \leq n$ mit $A^T p_m = 0$, und wir wählen $\tilde{m} = m$.

zu c) $p_{k+2} = N(Np_k - \delta_k p_{k-1}) - \delta_{k+1} p_k = N^2 p_k - \delta_k (p_k + \delta_{k-1} p_{k-2}) - \delta_{k+1} p_k$
$$= N^2 p_k - (\delta_k + \delta_{k+1}) p_k - \delta_k \delta_{k-1} p_{k-2}.$$

zu d) Mit c) und $p_o=r_o$ zeigt man

$$[p_o,p_2,\ldots,p_{2k}] \subseteq [r_o,N^2r_o,\ldots,N^{2k}r_o] \quad \text{für } 0 \leq k < \frac{m}{2},$$

$$[p_1,p_3,\ldots,p_{2k+1}] \subseteq [Nr_o,N^3r_o,\ldots,N^{2k+1}r_o] \quad \text{für } 0 \leq k < \frac{m-1}{2},$$

und daraus folgt

$$[p_o,p_1,\ldots,p_{k-1}] \subseteq [r_o,Nr_o,\ldots,N^{k-1}r_o] \quad \text{für } 1 \leq k \leq m;$$

dabei gilt nun jeweils Gleichheit, denn mit $A^Tp_o,\ldots,A^Tp_{m-1}$ sind auch $p_o,p_1,\ldots,p_{m-1}$ linear unabhängig.

zu e) Die Teile a)b) und d)iii) liefern einerseits die lineare Unabhängigkeit von $A^Tr_o,NA^Tr_o,\ldots,N^{m-1}A^Tr_o$, andererseits mit $Np_k=p_{k+1}+\delta_kp_{k-1}$, $0\leq k<m$:

$$N^mA^Tr_o = A^TN(N^{m-1}r_o) \in [A^TNp_o,A^TNp_1,\ldots,A^TNp_{m-1}]$$

$$\subseteq [A^Tp_o,A^Tp_1,\ldots,A^Tp_{m-1},\underbrace{A^Tp_m}_{=0}] = [A^Tr_o,NA^Tr_o,\ldots,N^{m-1}A^Tr_o]$$

zu f)g) Zur Unterscheidung bezeichnen wir zunächst den Abbruchindex der Lanczos-Rekursion mit $\tilde{m}$, und m sei die Anzahl der Komponenten aus (13.6); eine solche Darstellung von A^Tr_o existiert, denn wegen $N=-N^T$ besitzt der $\mathbb{C}^n$ eine Orthonormalbasis aus Eigenvektoren z_j von N und die zugehörigen Eigenwerte haben die Gestalt $i\lambda_j$, $\lambda_j\in\mathbb{R}$. Wir bemerken, daß die (mit den konjugiert komplexen Komponenten von z_j gebildeten) Vektoren $\bar{z}_j$ jeweils Eigenvektoren von N zu den Eigenwerten $\overline{i\lambda}_j=-i\lambda_j$ sind. Mit (13.6), $A^Tr_o\in\mathbb{R}^n$ folgt

$$A^Tr_o = \sum_{j=1}^{m} \rho_jz_j = \sum_{j=1}^{m} \bar{\rho}_j\bar{z}_j$$

und daraus

$$\{\lambda_1,\lambda_2,\ldots,\lambda_m\} = \{-\lambda_1,-\lambda_2,\ldots,-\lambda_m\}; \tag{13.7}$$

insbesondere sind genau $l=[(m+1)/2]$ der $\lambda_1^2,\ldots,\lambda_m^2$ paarweise verschieden. Da z_j und $\bar{z}_j$ Eigenvektoren von $A^TA=S-N^2$ zum Eigenwert $s+\lambda_j^2$ sind, ist l gerade die Anzahl der Eigenräume von A^TA, in denen A^Tr_o von Null verschiedene Komponenten hat.

Es bleibt noch $\tilde{m}=m$ zu zeigen: Nach e) ist $\tilde{m}$ der Grad eines reellen Polynoms $p\neq 0$ kleinsten Grades mit

$$p(N)A^T r_o = \sum_{j=1}^{m} \rho_j p(i\lambda_j) z_j = 0,$$

d.h. mit $p(i\lambda_j)=0$ für $1 \leq j \leq m$. Ein solches ist aber

$$p(t) \equiv \prod_{j=1}^{m} (t - i\lambda_j)$$

(wegen (13.7) sind die Koeffizienten von p reell!), und es folgt $\tilde{m}=m$. □

Mit den orthogonalen Suchrichtungen $A^T p_k$ bilden wir in Analogie zum STOD-Algorithmus(7.7) für symmetrische Matrizen den

(13.8) <u>STOD-Algorithmus (für Matrizen A = S - N):</u>

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$p_o = r_o = b - Ax_o, \quad p_{-1} = 0.$$

Für k = 0,1,2,...

1) Falls $A^T p_k = 0$: stop.

Sonst setze

2) $\alpha_k = r_k^T p_k / p_k^T AA^T p_k$,

$x_{k+1} = x_k + \alpha_k A^T p_k$, $r_{k+1} = r_k - \alpha_k AA^T p_k$,

3) $$\delta_k = \begin{cases} 0 & \text{für } k = 0 \\ -p_k^T AA^T p_k / p_{k-1}^T AA^T p_{k-1} & \text{für } k > 0 \end{cases},$$

$$p_{k+1} = Np_k - \delta_k p_{k-1}.$$

Erste Eigenschaften dieses Verfahrens notieren wir in folgendem

(13.9) <u>Satz:</u> a) $r_k^T p_j = 0$ für $0 \leq j < k \leq m$.

b) $r_k^T p_j = r_o^T p_j$ für $0 \leq k \leq j \leq m$.

c) $\alpha_{2k-1} = 0$, $x_{2k} = x_{2k-1}$ für $1 \leq k \leq m/2$.

d) Es sind äquivalent:

(1) $b \in R(A)$

(2) $p_m = 0$

(3) $r_m = 0$.

Beweis: zu a) Induktion nach k:
Für k=0 ist nichts zu zeigen; gilt die Behauptung bereits für ein k<m, so folgt

$$r_{k+1}^T p_j = r_k^T p_j - \alpha_k p_k^T A A^T p_j = 0 \qquad \text{für } 0 \leqslant j \leqslant k,$$

wobei (13.5)b) und (falls j=k) die Definition von α_k benutzt wird.
b) ergibt sich aus

(13.10) $$r_k = r_o - \alpha_o A A^T p_o - \cdots - \alpha_{k-1} A A^T p_{k-1}$$

und (13.5)b); c) erhalten wir mit b) und (13.5)d)ii).
zu d) "(3) $\Rightarrow$ (1)" ist trivial.
"(1) $\Rightarrow$ (2)": $b \in R(A)$ impliziert $r_o \in R(A)$, also $S_k \subseteq R(A)$, $k \geq 1$. Wegen (13.5)d)iii),a) gilt dann

$$p_m = N p_{m-1} - \delta_{m-1} p_{m-2} \in [r_o, N r_o, \ldots, N^m r_o] = S_{m+1}$$

und $p_m \in R(A) \cap N(A^T)$, d.h. $p_m = 0$.
"(2) $\Rightarrow$ (3)": Mit (13.10), $N p_k = p_{k+1} + \delta_k p_{k-1}$, $0 \leqslant k < m$, $p_m = 0$ folgt

$$\begin{aligned} r_m &\in [r_o, (S-N^2)p_o, (S-N^2)p_1, \ldots, (S-N^2)p_{m-1}] \\ &\subseteq [p_o, p_1, \ldots, p_{m-1}] \end{aligned}$$

und damit $r_m^T r_m = 0$ wegen a). □

Aus (13.9)d) ist ersichtlich:
Bei nichtsingulärem A (für S=I ist dies stets der Fall) liefert der Algorithmus(13.8)

$$x_m = \bar{x} := A^{-1} b;$$

bei singulärem A (z.B. S=0, n ungerade) müssen wir Konsistenz von Ax=b voraussetzen: Dann folgt (mit $\tilde{\tilde{x}}_o$ aus der Zerlegung (12.3))

$$x_m \in x_o + R(A^T) = \tilde{\tilde{x}}_o + R(A^T),$$

also $$x_m = \bar{x} := \tilde{\tilde{x}}_o + A^+ b.$$

Damit formulieren wir einen weiteren

(13.11) Satz: Es gilt:

a) $$x_{2k-1} = \arg\min_{x \in x_o + A^T S_{2k-1}} \|x - \bar{x}\| = \arg\min_{x \in x_o + S_k^2} \|x - \bar{x}\|$$

für $1 \leq k \leq l$ $(= [(m+1)/2])$, wobei

$$S_k^2 = [A^T r_o, (A^T A)A^T r_o, \ldots, (A^T A)^{k-1} A^T r_o];$$

b) bei selbem Startwert x_o stimmen die Näherungen der Algorithmen (13.8) und (12.6) überein:

$$x_{2k} = x_{2k-1} = x_k^{CRAIG} \quad \text{für } 1 \leq k \leq l.$$

Beweis: zu a) Aus der Orthogonalität der $A^T p_k$ und $x_m = \bar{x}$ folgt

$$x_{2k-1} = \arg\min_{x \in x_o + A^T [p_o, p_1, \ldots, p_{2k-2}]} \|x - \bar{x}\| ;$$

wegen (13.9)c), (13.5)d)i) können wir uns bei der Minimierung auf

$$x \in x_o + A^T [p_o, p_2, p_4, \ldots, p_{2k-2}]$$
$$= x_o + [A^T r_o, N^2 A^T r_o, N^4 A^T r_o, \ldots, N^{2k-2} A^T r_o] = x_o + S_k^2$$

beschränken.
b) ergibt sich aus a) und (12.8)d). □

Bemerkung: Vergleich der Rekursion (12.5), welcher die in (12.8)j) definierten Vektoren q_k des Craig-Algorithmus gehorchen, mit der Rekursion (13.5)c) (man beachte, daß $\delta_{2k} + \delta_{2k+1} = p_{2k}^T AA^T N^2 p_{2k} / p_{2k}^T AA^T p_{2k}$ gilt!) zeigt:

$$p_{2k} = q_k \quad \text{für} \quad 0 \leq k \leq l;$$

(13.11)b) läßt sich auch mit Hilfe dieses Zusammenhangs beweisen.

Auf Grund der Verbindung zum Craig-Algorithmus gilt die Abschätzung (12.8)e) in entsprechender Form für den STOD-Algorithmus(13.8). Wir wollen an dieser Stelle lediglich auf den eigentlich interessanten Fall S=I eingehen und nehmen dazu o.B.d.A.

$$0 \leq |\lambda_1| < |\lambda_2| < \cdots < |\lambda_l| \leq \|N\| = \Lambda$$

an.

(13.12) Korollar: Im Falle S=I, N≠0 gilt für die Fehler $e_k=\bar{x}-x_k$ des STOD-Algorithmus:

$$\frac{\|e_{2k}\|}{\|e_o\|} = \frac{\|e_{2k-1}\|}{\|e_o\|} \leq \min_{p\in\Pi_k} \max_{1\leq j\leq l} |p(1+\lambda_j^2)| \leq \min_{p\in\Pi_k} \max_{1+\lambda_1^2\leq t\leq 1+\lambda_1^2} |p(t)|$$

$$= \left(T_k\left(\frac{2+\lambda_1^2+\lambda_1^2}{\lambda_1^2-\lambda_1^2}\right)\right)^{-1} \leq \left(T_k\left(\frac{2+\Lambda^2}{\Lambda^2}\right)\right)^{-1} \quad \text{für } 1 \leq k \leq l.$$

Beweis: (13.11)a) liefert

$$\|e_{2k-1}\| = \min_{p\in\Pi_k} \|p(A^TA)e_o\| ,$$

(13.6) die Entwicklung

$$e_o = \sum_{j=1}^{m} \frac{\rho_j}{1+\lambda_j^2} z_j ,$$

und mit (13.5)g), (1.3) folgt die Behauptung. □

14. Ein verallgemeinerter STOD-Algorithmus für positiv reelle Matrizen. Zusammenhänge

In diesem Abschnitt sei vorausgesetzt, daß $A=M-N$ positiv reell, d.h. $M=(A+A^T)/2$ positiv definit, ist; insbesondere existiert die Cholesky-Zerlegung $M=QQ^T$. In Art einer Prekonditionierung führen wir vermöge $A' = Q^{-1}AQ^{-T}$, $b' = Q^{-1}b$, $x' = Q^T x$

(14.1) $$Ax = b$$

über in das äquivalente Gleichungssystem

(14.2) $$A' x' = b' ,$$

dessen Koeffizientenmatrix nun wegen unserer speziellen Wahl von Q die Gestalt

$$A' = I - N' , \quad N' := Q^{-1}NQ^{-T} = -N'^T$$

aufweist. Man kann also Algorithmus(13.8) auf (14.2) anwenden; transformieren wir gemäß

$$x_k = Q^{-T}x'_k , \quad r_k = Qr'_k , \quad p_k = Q^{-T}p'_k$$

wieder auf die Größen des Ausgangssystems (14.1), setzen

$$q_k := M^{-1}A^T p_k$$

und beachten (13.9)c), so ergibt sich ein verallgemeinertes STOD-Verfahren:

(14.3) GSTOD-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$r_o = b - Mx_o + Nx_o$, $p_{-1} = 0$;

erhalte p_o als Lösung von $Mp_o = r_o$

und setze $A^T p_o = r_o + Np_o$.

Für $k = 0,1,2,\ldots$

1) Falls $A^T p_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst,

2) erhalte q_k als Lösung von $Mq_k = A^T p_k$.

3) Setze

$$\alpha_k = \begin{cases} r_k^T p_k / q_k^T A^T p_k & \text{falls } k \text{ gerade} \\ 0 & \text{falls } k \text{ ungerade} \end{cases},$$

$$x_{k+1} = x_k + \alpha_k q_k, \quad r_{k+1} = r_k - \alpha_k (A^T p_k - N q_k)$$

und

4) $$\delta_k = \begin{cases} 0 & \text{für } k = 0 \\ -q_k^T A^T p_k / q_{k-1}^T A^T p_{k-1} & \text{für } k > 0 \end{cases},$$

$$p_{k+1} = q_k - p_k - \delta_k p_{k-1},$$

$$A^T p_{k+1} = N q_k - \delta_k A^T p_{k-1}.$$

Sei m ($\leq n$) der Abbruchindex dieses Verfahrens, wir vereinbaren ferner:

$$l := [(m+1)/2],$$

$$v_o := M^{-1} r_o, \quad K := M^{-1} N,$$

$$\bar{S}_k := [v_o, Kv_o, \ldots, K^{k-1} v_o],$$

$$\Lambda := \|N'\| = \rho(N') = \rho(M^{-1}N).$$

Im übrigen sei der triviale Fall $\Lambda = 0$ stets ausgeschlossen, außerdem sollen alle betrachteten Algorithmen mit demselben $x_o \neq \bar{x}$ gestartet werden.
Aus (13.9)c), (13.11)a), (13.12) leiten sich folgende Eigenschaften des GSTOD-Algorithmus her:

(14.4) <u>Satz:</u> a) $x_{2k} = x_{2k-1}$ für $1 \leq k \leq m/2$.

b) $$x_k = \underset{x \in x_o + M^{-1} A^T \bar{S}_k}{\arg\min} \|x - \bar{x}\|_M \quad \text{für } 1 \leq k \leq m.$$

c) $$\frac{\|e_k\|_M}{\|e_o\|_M} \leq \left(T_{[(k+1)/2]}\left(\frac{2+\Lambda^2}{\Lambda^2}\right)\right)^{-1} \leq 2\left(\frac{\sqrt{1+\Lambda^2}-1}{\sqrt{1+\Lambda^2}+1}\right)^{[(k+1)/2]}$$

für $0 \leq k \leq m$.

Die Näherungen aus (14.3) lassen sich auch auf andere Weise berechnen: Indem wir den Craig-Algorithmus(12.6) auf das speziell prekonditionierte System (14.2) anwenden, die gestrichenen Größen gemäß

$$x_k = Q^{-T}x'_k \;,\; r_k = Qr'_k \;,\; p_k = QA'^{T}d'_k$$

transformieren und

$$q_k := M^{-1}p_k,\; w_k := M^{-1}r_k$$

setzen, gelangen wir zum

(14.5) SPC-Craig-Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$r_o = b - Mx_o + Nx_o$;

erhalte w_o als Lösung von $Mw_o = r_o$

und setze $p_o = r_o + Nw_o$.

Für $k = 0,1,2,\ldots$

1) Falls $p_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst,

2) erhalte q_k als Lösung von $Mq_k = p_k$

und setze $\alpha_k = r_k^T w_k / p_k^T q_k$,

$x_{k+1} = x_k + \alpha_k q_k$, $r_{k+1} = r_k - \alpha_k(p_k - Nq_k)$,

3) erhalte w_{k+1} als Lösung von $Mw_{k+1} = r_{k+1}$

und setze $\beta_k = r_{k+1}^T w_{k+1} / r_k^T w_k$,

$p_{k+1} = r_{k+1} + Nw_{k+1} + \beta_k p_k$.

Die Näherungen des GSTOD-Algorithmus werden zur Unterscheidung mit x_k^{OD} bezeichnet; wir halten fest:

(14.6) Satz: Für die Näherungen x_k des SPC-Craig-Algorithmus gilt:

a) $x_k = x_{2k-1}^{OD} = x_{2k}^{OD}$ für $1 \leq k \leq l$ $(= [(m+1)/2])$;

b) $x_{k+1} = \hat{\rho}_{k+1}(\hat{\sigma}_{k+1} M^{-1} A^T w_k + x_k) + (1 - \hat{\rho}_{k+1}) x_{k-1}$,

$0 \leq k < l$,

wobei $x_{-1} := 0$, $\hat{\sigma}_{k+1} = r_k^T w_k / w_k^T A M^{-1} A^T w_k$ und

$$\hat{\rho}_{k+1} = \begin{cases} 1 & \text{für } k = 0 \\ (1 - \dfrac{\hat{\sigma}_{k+1}}{\hat{\sigma}_k} \dfrac{r_k^T w_k}{r_{k-1}^T w_{k-1}} \dfrac{1}{\hat{\rho}_k})^{-1} & \text{für } k > 0 \end{cases}.$$

Beweis: a) folgt aus (13.11)b) und b) aus (12.8)k). □

(14.3) und (14.5) sind also mathematisch äquivalent; beide Algorithmen stehen zudem in engem Zusammenhang mit dem Verfahren von Concus, Golub [7] und Widlund [36], welches (für den Spezialfall M=I) bereits im letzten Abschnitt angesprochen wurde und das für allgemeine positiv reelle Matrizen folgende Form besetzt:

(14.7) Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$r_o = b - Mx_o + Nx_o, \quad x_{-1} = r_{-1} = 0.$$

Für k = 0,1,2,...

1) Falls $r_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst,

2) erhalte v_k als Lösung von $Mv_k = r_k$,

3) setze

$$\rho_{k+1} = \begin{cases} 1 & \text{für } k = 0 \\ (1 + \dfrac{r_k^T v_k}{r_{k-1}^T v_{k-1}} \dfrac{1}{\rho_k})^{-1} & \text{für } k > 0 \end{cases},$$

$$x_{k+1} = x_{k-1} + \rho_{k+1}(v_k + x_k - x_{k-1}),$$

$$r_{k+1} = r_{k-1} + \rho_{k+1}(Nv_k - r_{k-1}).$$

Der Abbruchindex dieses Verfahrens ist (wie beim GSTOD-Algorithmus, vgl. (13.5)e)) durch die maximale Dimension von $\bar{S}_k$, $k \geq 1$, gegeben und daher gleich m. Die Näherungen x_k, $1 \leq k \leq m$, sind durch die Galerkin-Bedingung

(14.8) $$r_k = A(\bar{x} - x_k) \perp \bar{S}_k, \quad x_k \in x_o + \bar{S}_k,$$

eindeutig bestimmt; Widlund [36] leitete die Abschätzung

$$(14.9) \qquad \frac{\|e_k\|_M}{\|e_o\|_M} \leq \sqrt{1+\Lambda^2}\,(T_{[k/2]}\,(\tfrac{2+\Lambda^2}{\Lambda^2}))^{-1},\quad 0 \leq k \leq m,$$

für die Fehler $e_k=\bar{x}-x_k$ her.

Wir zeigen nun:

(14.10) Satz: Für die Näherungen aus (14.7) gilt:

a) $x_{2k} = x^{OD}_{2k-1} = x^{OD}_{2k}$ für $1 \leq k \leq m/2$;

b) $$\frac{1}{\sqrt{1+\Lambda^2}}\|e_{k+1}\|_M \leq \|e_k\|_M \leq \sqrt{1+\Lambda^2}\,\|e_{k-1}\|_M$$

für $1 \leq k < m$.

Beweis: zu a) Aus der Minimierungseigenschaft (14.4)b) des GSTOD-Algorithmus folgt

$$\bar{x} - x^{OD}_{2k} \perp A^T\bar{S}_{2k},$$

und wegen (14.4)a), $M^{-1}A^T=I+K$ gilt

$$x^{OD}_{2k} = x^{OD}_{2k-1} \in x_o + (I+K)\bar{S}_{2k-1} \subseteq x_o + \bar{S}_{2k}.$$

Also erfüllt x^{OD}_{2k} die Bedingung (14.8), und wir erhalten

$$x^{OD}_{2k} = x^{OD}_{2k-1} = x_{2k}.$$

zu b) Wie der Übergang von (14.1) zu (14.2) zeigt, dürfen wir o.B.d.A. $A=I-N$, $N=-N^T$, annehmen; sei $1\leq k\leq m$. Aus (14.8) folgt speziell

$$e_k^TA^T(x_o - x_k) = e_k^TA^T(x_o - x_{k-1}) = 0$$

und damit

$$e_k^Te_k = e_k^T(I+N)e_k = e_k^TA^T(e_o+x_o-x_k) = e_k^TA^T(e_o+x_o-x_{k-1})$$

$$= e_k^TA^Te_{k-1} \leq \|e_k\|\,\|I+N\|\,\|e_{k-1}\|,$$

also $\|e_k\| \leq \sqrt{1+\rho(N)^2}\,\|e_{k-1}\|$. □

(14.11) Korollar: Für $1 \leq k < m/2$ gilt:

$$\frac{1}{\sqrt{1+\Lambda^2}} \| e^{OD}_{2k+1} \|_M \leq \| e_{2k+1} \|_M \leq \sqrt{1+\Lambda^2} \| e^{OD}_{2k-1} \|_M .$$

Nach (14.10)a), (14.6)a) stimmen die Näherungen $x_0, x_2, x_4, \ldots$ des Verfahrens (14.7) mit den von GSTOD- bzw. SPC-Graig-Algorithmus berechneten überein; gegenüber diesen stellen die in (14.7) zusätzlich gelieferten $x_1, x_3, x_5, \ldots$ nach (14.11) für kleine Λ keine wesentlich verbesserten Näherungen dar. Man beachte, daß pro Iteration von (14.3) und (14.7), sowie pro "halber" Iteration von (14.5) jeweils ein Gleichungssystem mit Koeffizientenmatrix M zu lösen ist. Wegen der Fehlerschranken (14.4)c), (14.9) sind die betrachteten Verfahren in der Regel daher nur für schwach unsymmetrische, positiv reelle Matrizen sinnvoll; als Maß für die Unsymmetrie von A dient dabei gerade der Spektralradius $\Lambda = \rho(M^{-1}N)$ von $M^{-1}N$.

Wir wollen noch auf einen weiteren Zusammenhang hinweisen: Seien zunächst A, B nichtsinguläre n×n-Matrizen, und mit $G := I - B^{-1}A$ wird das iterative Verfahren

$$x_{k+1} = Gx_k + B^{-1}b, \tag{14.12}$$

sowie die "doppelte" Methode

$$x_{k+1} = G^2 x_k + GB^{-1}b + B^{-1}b,$$

die sich durch Zusammenfassen zweier Schritte von (14.12) ergibt, zur Lösung von $Ax=b$ gebildet. Hageman, Luk und Young [15] zeigten, daß in gewissen Fällen die beschleunigten Versionen

$$x_{k+1} = \rho_{k+1}(\sigma_{k+1}\delta_k + x_k) + (1 - \rho_{k+1})x_{k-1}, \tag{14.14}$$

$$\delta_k := Gx_k + B^{-1}b - x_k \quad (= B^{-1}(b - Ax_k)),$$

von (14.12) bzw.

$$x_{k+1} = \hat{\rho}_{k+1}(\hat{\sigma}_{k+1}\hat{\delta}_k + x_k) + (1 - \hat{\rho}_{k+1})x_{k-1}, \tag{14.15}$$

$$\hat{\delta}_k := G^2 x_k + GB^{-1}b + B^{-1}b - x_k,$$

von (14.13) "virtuell" äquivalent sind, d.h. für die Näherungen x_k aus (14.14) und $\hat{x}_k$ aus (14.15) gilt:

$$x_{2k} = \hat{x}_k \quad \text{für } k = 0,1,2,\ldots \ .$$

Nun sei A positiv reell und B=M gewählt, also $G=M^{-1}N=K$. Man erkennt, daß die Klasse der Verfahren (14.14) auch den Algorithmus(14.7) von Concus, Golub und Widlund enthält, und dieser ist - wie eines der Resultate aus [15] besagt - virtuell äquivalent zu (14.15), wenn mit $W:=M^{1/2}(I+K)^{-1}$

$$\hat{\sigma}_{k+1} = \hat{\delta}_k W^T W \hat{\delta}_k / \hat{\delta}_k^T W^T W (I - K^2)\hat{\delta}_k,$$

$$\text{(14.16)} \qquad \hat{\rho}_{k+1} = \begin{cases} 1 & \text{für } k = 0 \\ \left(1 - \dfrac{\hat{\sigma}_{k+1}}{\hat{\sigma}_k} \dfrac{\hat{\delta}_k^T W^T W \hat{\delta}_k}{\hat{\delta}_{k-1}^T W^T W \hat{\delta}_{k-1}} \dfrac{1}{\hat{\rho}_k}\right)^{-1} & \text{für } k > 0 \end{cases}$$

gewählt wird. Wir können hier einen einfachen Beweis für diesen Zusammenhang geben: Es gilt nämlich

$$\hat{\delta}_k = (I+K)M^{-1}(b-Ax_k) = M^{-1}A^T w_k, \text{ wobei } r_k=b-Ax_k,\ w_k=M^{-1}r_k,$$

$$W = M^{1/2}A^{-T}M, \quad W^TW = MA^{-1}MA^{-T}M,$$

$$\hat{\delta}_k^T W^T W \hat{\delta}_k = w_k^T M w_k = r_k^T w_k,$$

sowie mit $M(I-K^2)=A^TM^{-1}A$

$$\hat{\delta}_k^T W^T W (I-K^2)\hat{\delta}_k = w_k^T MA^{-T}M(I-K^2)M^{-1}A^T w_k = w_k^T AM^{-1}A^T w_k;$$

also ist durch (14.15), (14.16) gerade die Rekursion (14.6)b) des SPC-Craig-Verfahrens gegeben, und letzteres ist nach (14.10)a), (14.6)a) virtuell äquivalent zu Algorithmus(14.7).

15. Ein verallgemeinerter MCR-Algorithmus für positiv reelle Matrizen. Zusammenhänge

Zunächst werden wieder Matrizen der speziellen Form

$$A = S - N,\ S = 0 \quad \text{oder} \quad S = I,\ N = -N^T,$$

betrachtet. Basierend auf dem Lanczos-Algorithmus (13.4) bilden wir nun - die Minimierungseigenschaft (13.2) im Visier - den

(15.1) MCR-Algorithmus (für Matrizen $A = S - N$):

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$p_o = r_o = b - Ax_o,\ p_{-1} = 0.$$

Für $k = 0,1,2,\ldots$

1) Falls $Ap_k = 0$: stop.

Sonst setze

2) $\alpha_k = r_k^T Ap_k / p_k^T A^T Ap_k,$

$$x_{k+1} = x_k + \alpha_k p_k,\ r_{k+1} = r_k - \alpha_k Ap_k,$$

3) $$\delta_k = \begin{cases} 0 & \text{für } k = 0 \\ -p_k^T A^T Ap_k / p_{k-1}^T A^T Ap_{k-1} & \text{für } k > 0 \end{cases},$$

$$p_{k+1} = Np_k - \delta_k p_{k-1}.$$

Man beachte, daß $A^TA = AA^T$ und insbesondere $p_k^T A^T Ap_k = p_k^T AA^T p_k$ gilt; natürlich stimmt der Abbruchindex m des MCR-Algorithmus mit dem des STOD-Algorithmus (13.8) überein. Wir formulieren nun die wesentlichen Eigenschaften von (15.1):

(15.2) Satz: a) $p_j^T A^T Ap_k = 0$ für $0 \leq j < k \leq m$.

b) $r_k^T Ap_j = 0$ für $0 \leq j < k \leq m$.

c) $r_k^T Ap_j = r_o^T Ap_j$ für $0 \leq k \leq j \leq m$.

d) Im Fall $S = 0$ gilt $\alpha_{2k} = 0$ für $0 \leq k < m/2$.

e) $$A^T r_m = 0: \; x_m = \bar{x} := \begin{cases} A^{-1}b & \text{falls } A \text{ nichtsingulär} \\ \tilde{\tilde{x}}_o + A^+ b & \text{falls } A \text{ singulär} \end{cases}.$$

f) $$x_k = \underset{x \in x_o + S_k}{\arg\min} \|x - \bar{x}\|_{A^T A} \quad \text{für } 1 \leq k \leq m.$$

Beweis: a) wurde aus (13.5)b) übernommen. b) ergibt sich durch Induktion nach k, wobei man

$$r_{k+1}^T A p_j = r_k^T A p_j - \alpha_k p_k^T A^T A p_j$$

und a) ausnutzt. c) folgt aus

$$r_k = r_o - \alpha_o A p_o - \cdots - \alpha_{k-1} A p_{k-1}$$

und a); im Fall A=-N erhalten wir mit c) und (13.5)d)i)

$$r_{2k}^T A p_{2k} = -r_o^T N p_{2k} = 0,$$

also d).
zu e) Für $0 \leq k < m$ gilt

$$A^T A p_k = A^T (S-N) p_k = S A^T p_k - A^T p_{k+1} - \delta_k A^T p_{k-1},$$

$$A A^T p_k = A (S+N) p_k = S A p_k + A p_{k+1} + \delta_k A p_{k-1};$$

wegen $A^T p_m = A p_m = 0$ führt dies zu

$$\begin{aligned} A A^T r_m &= A (A^T r_o - \alpha_o A^T A p_o - \cdots - \alpha_{m-1} A^T A p_{m-1}) \\ &\in [A A^T p_o, A A^T p_1, \ldots, A A^T p_{m-1}] \\ &\subseteq [A p_o, A p_1, \ldots, A p_{m-1}], \end{aligned}$$

und mit b) folgt $\|A^T r_m\| = r_m^T A A^T r_m = 0$.
Bei nichtsingulärem A (insbesondere im Fall S=I) ist daher $x_m = \bar{x}$ die Lösung von Ax=b. Falls A=-N singulär ist, stellt x_m immerhin eine Lösung der Normalgleichungen dar; d) in Verbindung mit (13.5)d)ii) zeigt (mit $\tilde{\tilde{x}}_o$ aus der Zerlegung (12.3))

$$x_m \in x_o + R(N) = \tilde{\tilde{x}}_o + R(A^T),$$

und es resultiert $x_m = \tilde{\tilde{x}}_o + A^+ b$.
f) gilt wegen a) und (13.5)d)iii). □

Unter Berücksichtigung von (15.2)d), (13.5)d)ii) liefert ein Vergleich der Minimierungseigenschaften (15.2)f) und (12.4)d) folgendes

(15.3) Korollar: Bei schiefsymmetrischen Matrizen $A = -N$ stimmen die Näherungen aus den Algorithmen (15.1) und (12.2) überein:

$$x_{2k+1} = x_{2k} = x_k^{CGN} \quad \text{für } 0 \leq k \leq m/2.$$

Im Fall S=0 sind also STOD- und Craig-, sowie MCR- und CGN-Algorithmus mathematisch äquivalent, d.h. es liegt ein Analogon zum Entartungsfall bei symmetrischen Matrizen vor (vgl. Satz(6.6)). Man beachte auch, daß für schiefsymmetrische Matrizen Bedingungen der Form (6.4)b)c) stets erfüllt sind: Das Spektrum liegt auf der imaginären Achse symmetrisch zum Nullpunkt, und bei der Entwicklung reeller Vektoren nach orthonormalen Eigenvektoren stimmen die Beträge der zu einem konjugiert komplexen Paar von Eigenwerten gehörigen Komponenten überein.
Ein unterschiedliches Verhalten zeigt sich dagegen im Fall S=I: Nach (13.11)b) hatte zwar der STOD-Ansatz auf das Verfahren von Craig geführt, der MCR-Algorithmus dagegen erzeugt Näherungen, die nie auf der Stelle treten, d.h. $x_{k+1} \neq x_k$ für $0 \leq k < m$.

(15.4) Satz: Sei $A = I - N$, $N^T = -N$, und $\Lambda := \|N\| > 0$; $i\lambda_1, i\lambda_2, \ldots, i\lambda_m$ seien die Eigenwerte (von N) zu (13.6). Dann gelten für die Residuen des MCR-Algorithmus(15.1) folgende Abschätzungen:

a) $$\frac{\|r_k\|}{\|r_o\|} \leq \min_{p \in \Pi_k} \max_{1 \leq j \leq m} |p(1-i\lambda_j)|$$

$$\leq \min_{p \in \Pi_k} \max_{-\Lambda \leq t \leq \Lambda} |p(1+it)| =: m_k < \left(T_{[k/2]}\left(\frac{2+\Lambda^2}{\Lambda^2}\right)\right)^{-1}$$

für $1 \leq k \leq m$;

b) $$\frac{\|r_{k+1}\|}{\|r_k\|} \leq m_1 = \frac{\Lambda}{\sqrt{1+\Lambda^2}} \quad \text{(insbesondere } \alpha_k \neq 0\text{) für } 0 \leq k < m;$$

c) $$\frac{\|r_{k+2}\|}{\|r_k\|} \leq m_2 = 1 - \frac{1}{\sqrt{1+\Lambda^2}} \quad \text{für } 0 \leq k \leq m-2.$$

Beweis: zu a) Wir schreiben (15.2)f) in der Form

$$\|r_k\| = \min_{x \in x_o + S_k} \|b - Ax\| = \min_{p \in \Pi_k} \|p(A) r_o\| ;$$

(13.5)f) liefert die Entwicklung

$$r_o = \sum_{j=1}^{m} \frac{\rho_j}{1 + i\lambda_j} z_j ,$$

wobei $z_1, \ldots, z_m$ orthonormale Eigenvektoren von A zu den Eigenwerten $1-i\lambda_1, \ldots, 1-i\lambda_m$ darstellen, und es ergeben sich - man beachte noch

$$m_k \leq m_{2[k/2]} < \left(T_{[k/2]}\left(\frac{2+\Lambda^2}{\Lambda^2}\right)\right)^{-1}$$

(siehe (16.17)b)) - die gewünschten Abschätzungen.
zu b)c) Die Werte für m_1 und m_2 wurden aus Satz(16.17)f) entnommen. Für k=0 gelten die Behauptungen wegen a); sei daher $0<k<m$. Fassen wir $x_k = \hat{x}_o$ als neuen Startwert auf, so folgt mit

$$\hat{x}_o \in x_o + S_k, \quad \hat{r}_o = b - A\hat{x}_o \in r_o + AS_k \subseteq r_o + S_{k+1}, \quad A\hat{r}_o \in S_{k+2}$$

und (15.2)f)

$$\|r_{k+j}\| = \min_{x \in x_o + S_{k+j}} \|b-Ax\| \leq \min_{x \in \hat{x}_o + [\hat{r}_o, A^{j-1}\hat{r}_o]} \|b-Ax\| = \|\hat{r}_j\|,$$

also

$$\frac{\|r_{k+j}\|}{\|r_k\|} \leq \frac{\|\hat{r}_j\|}{\|\hat{r}_o\|} \leq m_j \qquad \text{für } j = 1,2. \square$$

Bemerkung: Die benutzte obere Schranke für m_k ist im Sinne

$$\lim_{k\to\infty} (m_k)^{1/k} = \lim_{k\to\infty} \frac{1}{|T_k(\frac{1}{i\Lambda})|^{1/k}} = \lim_{k\to\infty} \frac{1}{\left(T_{[k/2]}\left(\frac{2+\Lambda^2}{\Lambda^2}\right)\right)^{1/k}}$$

asymptotisch exakt (siehe (16.17)c)).

Es sei nun an Abschnitt 5 erinnert: Dort haben wir skizziert, daß die Rekursion

(15.5) $$d_{k+1} = r_{k+1} + \beta_k d_k ,$$

mit welcher die Suchrichtungen d_k des CR-Algorithmus für positiv definite Matrizen erzeugt werden, bei indefinitem A im allgemeinen nicht mehr brauchbar ist. Das liegt am möglichen Auftreten der

Schrittweite $\alpha_k=0$, die dann beim Weiterrechnen mit (15.5) den vorzeitigen Abbruch des Verfahrens auslöst. Beim MCR-Algorithmus (15.1) für unsere speziellen Matrizen A=I-N wird durch Satz(15.4)b) stets $\alpha_k \neq 0$ garantiert, und es ist daher zu erwarten, daß man in diesem Fall auch mit einer Rekursion der Gestalt (15.5) auskommt. In der Tat besitzen die gemäß

$$d_o := r_o,\ d_k := \alpha_{k-1}p_k \quad \text{für} \quad 1 \leq k \leq m,$$

umnormierten Suchrichtungen folgende Eigenschaften:

(15.6) Satz: Für $0 \leq k < m$ gilt:

a) $r_k^T A d_k = r_k^T r_k$;

b) $d_{k+1} = r_{k+1} + \beta_k d_k$, wobei $\beta_k := -r_{k+1}^T r_{k+1} / r_k^T r_k$;

c) $\beta_k = -r_{k+1}^T A^T A d_k / d_k^T A^T A d_k$.

d) Für $0 \leq j, k \leq m$ gilt:

$$r_j^T A r_k = \begin{cases} 0 & \text{falls } j > k \\ r_k^T r_k & \text{falls } j = k \\ 2r_j^T r_k = 2r_k^T r_k & \text{falls } j < k \end{cases}.$$

Beweis: Es wird noch $\alpha_{-1}:=1$, $\beta_{-1}:=0$, $d_{-1}:=0$ vereinbart.
zu a)b) Wir zeigen

$$d_k = r_k + \beta_{k-1} d_{k-1}, \quad 0 \leq k \leq m,$$

durch Induktion nach k. Für k=0 ist dies trivial richtig; sei jetzt $0 \leq k < m$ und die Behauptung bereits bewiesen für alle Indizes $\leq k$. Mit (15.2)b) und A=I-N folgt

$$\alpha_{j-1} r_j^T A p_j = r_j^T A d_j = r_j^T A (r_j + \beta_{j-1} d_{j-1})$$

$$= r_j^T A r_j = r_j^T r_j \qquad \text{für } 0 \leq j \leq k,$$

also insbesondere a). Wegen

$$r_{k+1}^T r_{k+1} = (r_k - \alpha_k A p_k)^T (r_k - \alpha_k A p_k) = r_k^T r_k - \alpha_k r_k^T A p_k$$

ergibt sich weiter

$$-\beta_k = 1 - \alpha_k r_k^T A p_k / r_k^T r_k = 1 - \alpha_k / \alpha_{k-1},$$

und außerdem gilt - wegen

$$\delta_k = -\frac{p_k^T A^T A p_k}{p_{k-1}^T A^T A p_{k-1}} = -\frac{r_k^T A p_k}{\alpha_k}\,\frac{\alpha_{k-1}}{r_{k-1}^T A p_{k-1}} = -\frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}\,\frac{\alpha_{k-2}}{\alpha_k}$$

$$= \beta_{k-1}\,\frac{\alpha_{k-2}}{\alpha_k}$$

(falls $k>0$) bzw. $\delta_o=\beta_{-1}=0$ -

$$\beta_{k-1}d_{k-1} = \alpha_k\delta_k p_{k-1}.$$

Damit erhalten wir schließlich

$$d_{k+1} = \alpha_k p_{k+1} = \alpha_k(p_k - Ap_k - \delta_k p_{k-1}) = r_{k+1} - r_k + \alpha_k p_k - \alpha_k\delta_k p_{k-1}$$

$$= r_{k+1} - d_k + \beta_{k-1}d_{k-1} + (\alpha_k/\alpha_{k-1})d_k - \alpha_k\delta_k p_{k-1}$$

$$= r_{k+1} + \beta_k d_k.$$

zu c) Aus b) und (15.2)a) folgt

$$0 = d_{k+1}^T A^T A d_k = r_{k+1}^T A^T A d_k + \beta_k d_k^T A^T A d_k.$$

zu d) Sei $0\leq j,k\leq m$: Für $j=k$ ist die Behauptung wegen $A=I-N$ offensichtlich, falls $j>k$ führt b) und (15.2)b) zu

$$r_j^T A r_k = r_j^T A(d_k - \beta_{k-1}d_{k-1}) = 0.$$

Im Fall $j<k$ liefert das eben Gezeigte $r_j^T A^T r_k = r_k^T A r_j = 0$, und unter Beachtung von $A+A^T=2I$ ergibt sich

$$r_j^T A r_k = r_j^T(A + A^T)r_k = 2r_j^T r_k;$$

aus (15.2)b) folgt

$$r_k^T r_k = r_k^T(r_j - \alpha_j Ap_j - \cdots - \alpha_{k-1}Ap_{k-1}) = r_k^T r_j = r_j^T r_k. \square$$

Mit Teil a) und b) des letzten Satzes erhalten wir eine äquivalente Version von (15.1), nämlich den folgenden

(15.7) <u>Algorithmus (für Matrizen A = I - N):</u>

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$d_o = r_o = b - Ax_o.$$

Für $k = 0,1,2,\ldots$

1) Falls $r_k=0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst setze

2) $\alpha_k = r_k^T r_k / d_k^T A^T A d_k$,

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

3) $\beta_k = -r_{k+1}^T r_{k+1} / r_k^T r_k$,

$d_{k+1} = r_{k+1} + \beta_k d_k$.

Die Verfahren dieses Abschnitts lassen sich natürlich auch zur Lösung von $Ax=b$ mit positiv reellem $A=M-N$, $N=-N^T$, $M=QQ^T$ positiv definit, einsetzen, indem man sie auf das speziell prekonditionierte System

$$A' x' = b' \quad , \quad A' = Q^{-1} A Q^{-T} \quad , \quad b' = Q^{-1} b$$

anwendet. Mit

$$x_k = Q^{-T} x_k', \quad r_k = Q r_k', \quad p_k = Q^{-T} p_k' \text{ und } q_k := M^{-1} A p_k$$

resultiert so aus (15.1) ein verallgemeinertes MCR-Verfahren:

(15.8) <u>GMCR-Algorithmus:</u>

Start: Wähle $x_o \in \mathbb{R}$ und setze

$r_o = b - Mx_o + Nx_o$, $p_{-1} = Ap_{-1} = 0$;

erhalte p_o als Lösung von $Mp_o = r_o$

und setze $Ap_o = r_o - Np_o$.

Für $k = 0,1,2,\ldots$

1) Falls $Ap_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst,

2) erhalte q_k als Lösung von $Mq_k = Ap_k$.

3) Setze

$\alpha_k = r_k^T q_k / q_k^T A p_k$,

$x_{k+1} = x_k + \alpha_k p_k$, $r_{k+1} = r_k - \alpha_k A p_k$

und

4) $$\delta_k = \begin{cases} 0 & \text{für } k = 0 \\ -q_k^T A p_k / q_{k-1}^T A p_{k-1} & \text{für } k > 0 \end{cases},$$

$$p_{k+1} = p_k - q_k - \delta_k p_{k-1},$$

$$Ap_{k+1} = Nq_k - \delta_k Ap_{k-1}.$$

Bei der Übertragung von (15.7) transformieren wir gemäß

$$x_k = Q^{-T}x'_k,\ r_k = Qr'_k,\ d_k = Q^{-T}d'_k,$$

setzen

$$q_k := M^{-1}Ad_k,\ w_k := M^{-1}r_k$$

und erhalten folgenden

(15.9) Algorithmus:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$r_o = b - Mx_o + Nx_o$;

erhalte d_o als Lösung von $Md_o = r_o$

und setze $Ad_o = r_o - Nd_o$, $w_o = d_o$.

Für $k = 0,1,2,\ldots$

1) Falls $r_k = 0$: stop, $x_k = \bar{x}$ ist Lösung von $Ax = b$.

Sonst,

2) erhalte q_k als Lösung von $Mq_k = Ad_k$.

3) Setze

$$\alpha_k = r_k^T w_k / q_k^T Ad_k,$$

$$x_{k+1} = x_k + \alpha_k d_k,\ r_{k+1} = r_k - \alpha_k Ad_k$$

und

4) $w_{k+1} = w_k - \alpha_k q_k$,

$$\beta_k = -r_{k+1}^T w_{k+1} / r_k^T w_k,$$

$$d_{k+1} = w_{k+1} + \beta_k d_k,$$

$$Ad_{k+1} = r_{k+1} - Nw_{k+1} + \beta_k Ad_k.$$

Aus (15.2)f) bzw. (15.4)a) leiten sich entsprechende Eigenschaften der beiden Verfahren (15.8) und (15.9) ab:

(15.10) Satz: Sei $\Lambda := \rho(M^{-1}N) > 0$ und m der Abbruchindex.

Für $1 \leq k \leq m$ gilt:

a) $$x_k = \arg\min_{x \in x_o + \overline{S}_k} \|b - Ax\|_{M^{-1}},$$

wobei $\overline{S}_k := [M^{-1}r_o, (M^{-1}N)M^{-1}r_o, \ldots, (M^{-1}N)^{k-1}M^{-1}r_o]$;

b) $$\frac{\|r_k\|_{M^{-1}}}{\|r_o\|_{M^{-1}}} < (T_{[k/2]}(\frac{2+\Lambda^2}{\Lambda^2}))^{-1} \leq 2\left(\frac{\sqrt{1+\Lambda^2} - 1}{\sqrt{1+\Lambda^2} + 1}\right)^{[k/2]}.$$

Man beachte, daß die angegebene Fehlerschranke im wesentlichen mit der des GSTOD-Algorithmus(14.3) übereinstimmt (siehe (14.4)c)).

In der folgenden Tabelle stellen wir Operationszahlen und Speicherplatzbedarf der behandelten Verfahren zur Lösung von Ax=b mit positiv reellem A zusammen. Die einzelnen Algorithmen wurden gerade so formuliert, daß anstelle von A·x die in der Regel billigeren Matrix-Vektor-Produkte N·x zu bilden sind. Eine Multiplikation mit M ist höchstens bei der Berechnung von r_o nötig, aber natürlich sind Gleichungssysteme mit M als Koeffizientenmatrix zu lösen ($M^{-1}\cdot x$). Es werden jeweils die Anzahlen der pro Iteration (bzw. pro halber Iteration beim SPC-Craig-Algorithmus) anfallenden Operationen aufgeführt:

	$M^{-1}\cdot x$	$N\cdot x$	ungefähre Zahl der Multiplikationen	Zahl der zu speichernden Vektoren des $\mathbb{R}^n$
GSTOD(14.3)	1	1	4,5n	8
SPC-Craig(14.5)	1	1	2,5n	5
Algorithmus(14.7)	1	1	3n	6
GMCR(15.8)	1	1	6n	8
Algorithmus(15.9)	1	1	7n	6

Wir wollen nun noch auf Zusammenhänge zwischen den Algorithmen dieses Abschnitts und weiteren verallgemeinerten CG-Verfahren für unsymmetrische A hinweisen. Young und Jea [38] setzen die Existenz einer Hilfsmatrix Z mit der Eigenschaft, daß ZA positiv reell ist, voraus und definieren dann durch die Galerkin-Bedingung

$$Zr_k \perp S_k = [r_o, Ar_o, \ldots, A^{k-1}r_o]$$

(man beachte, daß dies für $Z=I$ bzw. $Z=A^T$ auf (13.1) bzw. (13.2) führt) Näherungen x_k, welche mit dem nachstehenden Algorithmus berechnet werden:

(15.11) ORTHODIR:

Start: Wähle $x_o \in \mathbb{R}^n$ und setze

$$q_o = r_o = b - Ax_o.$$

Für $k = 0,1,2,\ldots$

1) Falls $q_k = 0$: stop, x_k löst $Ax = b$.

Sonst setze

2) $\alpha_k = r_k^T Z^T q_k / q_k^T ZAq_k$

$$x_{k+1} = x_k + \alpha_k q_k,\ r_{k+1} = r_k - \alpha_k Aq_k,$$

3)

$$\gamma_{k,i} = -\frac{q_i^T ZA^2 q_k + \sum_{j=0}^{i-1} \gamma_{k,j}\, q_i^T ZAq_j}{q_i^T ZAq_i}, \quad 0 \leq i \leq k,$$

$$q_{k+1} = Aq_k + \gamma_{k,k}q_k + \gamma_{k,k-1}q_{k-1} + \cdots + \gamma_{k,o}q_o.$$

Die Suchrichtungen q_k sind im Sinne von

(15.12) $$q_j^T ZAq_k = 0 \qquad \text{für } 0 \leq j < k$$

semiorthogonal; falls ZA nicht symmetrisch ist, gilt jedoch in der Regel $q_j^T ZAq_k \neq q_k^T ZAq_j$.

Schritt 3) von (15.11) benötigt Information aus sämtlichen vorhergehenden Iterationen; um das zu vermeiden, wurden die "abgeschnittenen" Varianten ORTHODIR(s), s eine natürliche Zahl, vorgeschlagen, bei denen stets

$$\gamma_{k,i} := 0 \quad \text{für } 0 \leq i \leq k-s$$

gesetzt wird. Im allgemeinen sind diese Verfahren natürlich nicht äquivalent zu ORTHODIR, und es ist möglich, daß sie vorzeitig oder aber nie abbrechen; eventuell sind deshalb Restarts durchzuführen. Wir betrachten jetzt wieder die speziellen Matrizen $A = sI - N$, $s=0$ oder $s=1$, $N=-N^T$, und wählen $Z:=A^T$. Vergleicht man die Rekursion

$$p_{k+1} = -(Ap_k - sp_k + \delta_k p_{k-1}) \quad \text{für } 0 \leq k < m,\ p_o = r_o,$$

der die Suchrichtungen von (15.1) gehorchen, mit Schritt 3) von (15.11) und berücksichtigt (15.12), so zeigt sich, daß für $0 \leq k < m$

$$\gamma_{k,k} = -s,\ \gamma_{k,k-1} = -\delta_k,\ \gamma_{k,i} = 0 \quad \text{für } 0 \leq i \leq k-2,$$

$$q_{k+1} = (-1)^{k+1} p_{k+1}$$

gilt. Wir sind damit auf folgendes Resultat gestoßen:

(15.13) Satz: MCR-Algorithmus(15.1), ORTHODIR(15.11) und die abgeschnittenen Versionen ORTHODIR(s), $s \geq 2$, sind äquivalent falls $Z^T = A = S - N$.

Axelsson [2] diskutiert die Möglichkeit, die Größen

$$(15.14) \qquad x_k = \arg\min_{x \in x_o + S_k} \|b - Ax\|$$

mit Hilfe von Vektoren d_k zu berechnen, die - in Anlehnung an das CR-Verfahren für positiv definite Matrizen - ausgehend von $d_o = r_o$ durch

$$(15.15) \quad d_{k+1} = r_{k+1} + \beta_k d_k,\ \beta_k = -r_{k+1}^T A^T A d_k / d_k^T A^T A d_k$$

erzeugt werden. (15.15) ist nicht für beliebige Matrizen brauchbar (bei schiefsymmetrischem A oder im Entartungsfall bei symmetrischem A gilt $r_1 = r_o$, $\beta_o = -1$, $d_1 = 0$), und Axelsson setzt A als positiv reell voraus. Im allgemeinen sind die Ad_k nicht orthogonal (lediglich $d_{k+1}^T A^T A d_k = 0$ wird durch die Wahl von β_k garantiert!), und die Rekursion der x_k nimmt die Gestalt

$$(15.16) \qquad x_{k+1} = x_k + \sum_{j=0}^{k} \alpha_{k,j} d_j$$

an. Es wurden wiederum abgeschnittene Varianten vorgeschlagen, die in (15.16) (und entsprechend bei der Minimierung (15.14)) nur die letzten s Suchrichtungen berücksichtigen. Vergleicht man mit Algorithmus(15.7) und beachtet die Aussagen von Satz(15.6), so ergibt sich:

(15.17) Satz: Für die speziellen Matrizen $A = I - N$, $N = -N^T$, sind Algorithmus(15.7) und die Axelsson'schen Algorithmen, $s \geq 1$, äquivalent.

16. Über ein Approximationsproblem

Sei $\Lambda>0$ und $E_\Lambda := \{z = 1 + it \in \mathbb{C} \mid t \in [-\Lambda, \Lambda]\}$; Ziel dieses Abschnitts ist es, Aussagen über die Größen

$$(16.1) \qquad \min_{p \in \Pi_k} \max_{z \in E_\Lambda} |p(z)|$$

zu gewinnen, welche in (15.4) im Zusammenhang mit dem MCR-Algorithmus aufgetaucht sind. Nun läßt sich E_Λ als entartete Ellipse mit den Brennpunkten $1 \pm i\Lambda$ und großer Halbachse Λ auffassen, und wir wollen uns zunächst mit dem allgemeineren Problem

$$(16.2) \qquad \min_{p \in \Pi_k'} \max_{z \in E(d,c,a)} |p(z)| \quad (=: m_k)$$

beschäftigen. Dabei sei

$$\Pi_k' := \{p(z) \equiv 1 + \sigma_1 z + \sigma_2 z^2 + \cdots + \sigma_k z^k \mid \sigma_1, \ldots, \sigma_k \in \mathbb{C}\},$$

$d, c \in \mathbb{C}$, $a>0$, und mit $E(d,c,a)$ der Rand der in der komplexen Ebene gelegenen Ellipse mit Mittelpunkt d, Brennpunkten $d \pm c$ und großer Halbachse a bezeichnet. Natürlich muß $a \geq |c|$ erfüllt sein, wobei uns - in Hinblick auf (16.1) - besonders der Grenzfall $a=|c|$ interessiert, bei dem die Ellipse zu einer Strecke degeneriert.
Liegt 0 innerhalb oder auf der Ellipse $E(d,c,a)$, so ist (16.2) trivial: Wie man mit Hilfe des Maximumprinzips für holomorphe Funktionen einsieht, gilt $m_k=1$ und $p \equiv 1$ ist Optimalpolynom (sogar das einzige, falls 0 echt innerhalb der Ellipse anzutreffen ist). Im folgenden nehmen wir deshalb stets an, daß der Nullpunkt außerhalb der Ellipse liegt.
Man beachte, daß beim Übergang von (16.1) nach (16.2) die Menge der zulässigen Polynome erweitert wurde; dadurch können wir bequem auf die Theorie der komplexen linearen Tschebyscheff-Approximation zurückgreifen. Dort werden Probleme der Form

$$(16.3) \qquad \min_{v \in V} \max_{z \in E} |f(z) - v(z)|$$

behandelt, wobei E ein kompakter Raum, f ein Element des komplexen Vektorraums

$$C(E) := \{f : E \to \mathbb{C} \mid f \text{ stetig}\}$$

und $V \subset C(E)$ ein linearer Teilraum der Dimension k ist. Zu (16.3)

existiert stets eine Minimallösung, die zudem eindeutig ist, falls V der Haarschen Bedingung gnügt, d.h. jedes $v \in V \setminus \{0\}$ höchstens k-1 Nullstellen in E besitzt. Mit $E := E(d,c,a)$, $f \equiv 1$ und

$$V := \{p \mid p+1 \in \Pi_k'\}$$

identifizieren wir (16.2) als eine Approximationsaufgabe der Gestalt (16.3); ferner ist die Haarsche Bedingung erfüllt, denn da 0 außerhalb der Ellipse liegt, gilt insbesondere $0 \notin E$. Durch Übertragung der Resultate aus der Approximationstheorie (siehe etwa Meinardus [23]) - mit

$$D(p) := \{z \in E \mid |p(z)| = \max_{u \in E} |p(u)|\}$$

wird die Menge der Extremalpunkte von p bezeichnet - erhalten wir somit:

(16.4) Satz: a) $p \in \Pi_k'$ ist genau dann Optimalpolynom von (16.2), wenn für alle $\sigma_1, \sigma_2, \ldots, \sigma_k \in \mathbb{C}$

$$\min_{z \in D(p)} \operatorname{Re}\left(\overline{p(z)}(\sigma_1 z + \sigma_2 z^2 + \cdots + \sigma_k z^k)\right) \leq 0$$

gilt.

b) Es existiert genau eine Optimallösung $p_k \in \Pi_k'$ von (16.2).

c) $D_k := D(p_k)$ enthält mindestens k+1 Punkte.

Für den Fall, daß die Ellipse symmetrisch zur reellen Achse liegt, also

(16.5) $$z \in E(d,c,a) \Rightarrow \bar{z} \in E(d,c,a)$$

erfüllt ist, läßt sich anfügen:

(16.6) Korollar: Es gilt $p_k \in \Pi_k$, d.h. die Probleme (16.2) und

$$\min_{p \in \Pi_k} \max_{z \in E(d,c,a)} |p(z)|$$

sind äquivalent.

Beweis: Sei $\bar{p}_k \in \Pi_k'$ das Polynom mit den konjugiert komplexen Koeffizienten von p_k. Wegen $|\bar{p}_k(z)| = |p_k(\bar{z})|$ und (16.5) ist $\bar{p}_k$ ebenfalls Optimallösung von (16.2); nach (16.4)b) ist eine solche aber eindeutig bestimmt, es folgt $p_k = \bar{p}_k$ und somit $p_k \in \Pi_k$. □

Die Optimalpolynome p_k sind bislang nur für spezielle Ellipsen bekannt [20]:
Für den Kreis (c=0) gilt

$$p_k(z) \equiv \left(\frac{d-z}{d}\right)^k ,$$

(im weiteren sei daher c=0 stets ausgeschlossen); sind beide Brennpunkte von E(d,c,a) reell, so ist p_k gerade durch das umnormierte Tschebyscheff-Polynom

$$t_k(z) \equiv \frac{T_k\left(\frac{d-z}{c}\right)}{T_k\left(\frac{d}{c}\right)} \tag{16.7}$$

gegeben.
Bei allgemeiner Ellipse ($d, c \in \mathbb{C}$) liefert (16.7) - wie jedes Polynom aus Π_k' - eine obere Schranke für den Optimalwert m_k von (16.2):

$$m_k \leq \max_{z \in E(d,c,a)} |t_k(z)| =: \tilde{\tau}_k . \tag{16.8}$$

t_k führt aber auch zu einer Abschätzung nach unten, nämlich

$$\underset{\sim}{\tau}_k := \min_{z \in E(d,c,a)} |t_k(z)| \leq m_k , \tag{16.9}$$

wie Manteuffel [20] mittels des Satzes von Rouché nachwies. Manteuffel zeigte ferner, daß im Fall a>|c| der nichtentarteten Ellipse $(\underset{\sim}{\tau}_k)^{1/k}$ und $(\tilde{\tau}_k)^{1/k}$ für $k \to \infty$ gegen denselben Grenzwert konvergieren, die Tschebyscheff-Schranken also in diesem Sinne asymptotisch exakt sind. Aus der Darstellung $T_k(w)=\cosh(k \cosh^{-1} w)$ ist ersichtlich [20], daß T_k die Ellipse $E_\alpha^o := E(0,1,\alpha)$, $\alpha \geq 1$, auf die (k-fach durchlaufene) Ellipse $E^o_{T_k(\alpha)}$ abbildet, und da die in (16.7) vorgeschaltete Transformation w=(d-z)/c gerade E(d,c,a) in $E^o_{\alpha_o}$, $\alpha_o := a/|c|$, überführt, ergibt sich

$$\left\{ \begin{matrix} \tilde{\tau}_k \\ \\ \underset{\sim}{\tau}_k \end{matrix} \right\} = \frac{(\alpha_o + \sqrt{\alpha_o^2-1})^k \pm (\alpha_o - \sqrt{\alpha_o^2-1})^k}{2\,|T_k(\frac{d}{c})|} . \tag{16.10}$$

Man erkennt, daß die Tschebyscheff-Schranken um so mehr auseinanderklaffen, je näher der Entartungsfall rückt; für $\alpha_o=1$ gilt $\underset{\sim}{\tau}_k=0$ und (16.9) ist trivial.

Bei der Suche nach einer anderen unteren Schranke für m_k hilft uns ein Resultat von Bernstein weiter, das eine Abschätzung für das Anwachsen des maximalen Betrags eines Polynoms beim Übergang von $[-1,1]$ nach E_α^o angibt. Für unsere Zwecke formulieren wir etwas allgemeiner folgendes

(16.11) Lemma: Sei $p(w) \equiv \sigma_o + \sigma_1 w + \cdots + \sigma_k w^k$, $\sigma_o, \sigma_1, \ldots, \sigma_k \in \mathbb{C}$, und $1 \leq \hat{\alpha} \leq \alpha$. Es gilt:

$$\max_{w \in E_\alpha^o} |p(w)| \leq \left(\frac{\alpha + \sqrt{\alpha^2 - 1}}{\hat{\alpha} + \sqrt{\hat{\alpha}^2 - 1}} \right)^k \max_{w \in E_{\hat{\alpha}}^o} |p(w)|.$$

Der Beweis für den Fall $\hat{\alpha}=1$ (siehe etwa [23]) läßt sich unmittelbar übertragen:
Die Transformation

$$w = \frac{1}{2}(v + \frac{1}{v}) \tag{16.12}$$

bildet sowohl den Kreis $|v|=r$ (≥ 1), als auch $|v|=1/r$ auf die Ellipse E_α^o mit $\alpha=\frac{1}{2}(r+\frac{1}{r})$ ab, wobei

$$r = \alpha + \sqrt{\alpha^2 - 1} \tag{16.13}$$

gerade die Summe der beiden Halbachsenlängen von E_α^o darstellt. Seien jetzt $r \geq \hat{r} \geq 1$ die auf diese Weise den gegebenen $\alpha \geq \hat{\alpha} \geq 1$ zugeordneten Werte, und sei

$$q(v) \equiv v^k p(\tfrac{1}{2}(v + \tfrac{1}{v})).$$

Für $|v|=1/\hat{r}$ gilt dann

$$|q(v)| \leq \frac{1}{\hat{r}^k} \max_{w \in E_{\hat{\alpha}}^o} |p(w)|; \tag{16.14}$$

wegen grad $p \leq k$ ist q wieder ein Polynom und insbesondere holomorph, nach dem Maximumprinzip ist (16.14) sogar für $|v| \leq 1/\hat{r}$ erfüllt, also speziell für $|v|=1/r$. Es folgt

$$\max_{w \in E_\alpha^o} |p(w)| \leq r^k \max_{|v|=\frac{1}{r}} |q(v)| \leq (\frac{r}{\hat{r}})^k \max_{w \in E_{\hat{\alpha}}^o} |p(w)|. \square$$

Der Normierungspunkt 0 liegt außerhalb von $E(d,c,a)$, sein Bild d/c unter der Transformation $w=(d-z)/c$ daher außerhalb von $E_{\alpha_o}^o$, und wir bezeichnen mit α_1 $(>\alpha_o \geq 1)$ die große Halbachse der durch $d/c \in E_{\alpha_1}^o$

definierten Ellipse. Für dieses α_1 gilt - die richtige Wahl eines der beiden Zweige der Wurzelfunktion vorausgesetzt -

$$(16.15)\quad \alpha_1 + \sqrt{\alpha_1^2-1} = \left|\frac{d}{c} + \sqrt{(\frac{d}{c})^2-1}\,\right| > 1 > \left|\frac{d}{c} - \sqrt{(\frac{d}{c})^2-1}\,\right|,$$

wie man sich anhand der Abbildungseigenschaften von (16.12) und mit (16.13) klarmacht. Anwendung von Lemma(16.11) mit $\alpha=\alpha_1$, $\hat{\alpha}=\alpha_o$, $p(w)\equiv p_k(d-cw)$ ergibt

$$1 = p_k(0) = p(\frac{d}{c}) \leq \max_{w\in E^o_{\alpha_1}} |p(w)| \leq \left(\frac{\alpha_1 + \sqrt{\alpha_1^2-1}}{\alpha_o + \sqrt{\alpha_o^2-1}}\right)^k m_k;$$

zusammen mit (16.8), (16.10), (16.15) resultiert also:

(16.16) <u>Satz:</u> Sei $\alpha_o = a/|c| \geq 1$. Es gilt:

$$a)\quad \left(\frac{\alpha_o + \sqrt{\alpha_o^2-1}}{\left|\frac{d}{c} + \sqrt{(\frac{d}{c})^2-1}\,\right|}\right)^k \leq m_k \leq \frac{T_k(\alpha_o)}{|T_k(\frac{d}{c})|} = \tilde{\tau}_k \quad \text{für } k = 1,2,\ldots;$$

$$b)\quad \lim_{k\to\infty} (m_k)^{1/k} = \lim_{k\to\infty} (\tilde{\tau}_k)^{1/k} = \frac{\alpha_o + \sqrt{\alpha_o^2-1}}{\left|\frac{d}{c} + \sqrt{(\frac{d}{c})^2-1}\,\right|}.$$

Wir kehren nun zu unserem Ausgangsproblem zurück und setzen für den Rest des Abschnitts $d=1$, $c=i\Lambda$, $a=\Lambda$. Man beachte, daß $E_\Lambda = E(1,i\Lambda,\Lambda)$ (16.5) erfüllt und somit Korollar(16.6) greift, d.h. (16.1) und (16.2) besitzen dasselbe Optimalpolynom $p_k\in\Pi_k$. Für p_k bzw. m_k, $k\in\mathbb{N}$, gilt:

(16.17) <u>Satz:</u> a)
$$p_k(z) \neq t_k(z) \equiv \frac{T_k(\frac{1-z}{i\Lambda})}{T_k(\frac{1}{i\Lambda})} \in \Pi_k.$$

$$b)\quad \left(\frac{\Lambda}{1 + \sqrt{1+\Lambda^2}}\right)^k \leq m_k < \frac{1}{|T_k(\frac{1}{i\Lambda})|},$$

wobei $|T_k(\frac{1}{i\Lambda})| = T_{k/2}(\frac{2+\Lambda^2}{\Lambda^2})$ falls k gerade.

$$c)\quad \lim_{k\to\infty} (m_k)^{1/k} = \lim_{k\to\infty} \frac{1}{|T_k(\frac{1}{i\Lambda})|^{1/k}} = \frac{\Lambda}{1 + \sqrt{1+\Lambda^2}}.$$

d) i) Die Menge D_k der Extremalpunkte von p_k enthält genau k+1 Elemente;

ii) $1 \pm i\Lambda \in D_k$;

iii) $1 \in D_k \Leftrightarrow$ k gerade.

e) Die Nullstellen von p_k liegen in $\{z \mid \mathrm{Re}\, z \geq 1\}$.

f) i) $m_1 = \dfrac{\Lambda}{\sqrt{1+\Lambda^2}}$, $p_1(z) \equiv 1 - \dfrac{z}{1+\Lambda^2}$;

ii) $m_2 = 1 - \dfrac{1}{\sqrt{1+\Lambda^2}}$,

$$p_2(z) \equiv 1 - \left(1 + \frac{1}{\sqrt{1+\Lambda^2}}\right) \frac{z}{\sqrt{1+\Lambda^2}} + \frac{z^2}{1+\Lambda^2} .$$

Beweis: zu a) t_k ist ein reelles Polynom (also aus Π_k), denn T_k ist (un-)gerade für (un-)gerades k; ferner gilt

(16.18)
$$\mu_k := \frac{1}{T_k(\frac{1}{i\Lambda})} \in \begin{cases} \mathbb{R}\setminus\{0\} & \text{falls k gerade} \\ i\mathbb{R}\setminus\{0\} & \text{falls k ungerade} \end{cases} .$$

Für E_Λ wird die Parametrisierung $z=1-it$, $t\in[-\Lambda,\Lambda]$, gewählt, wegen

$$t_k(1 - it) \equiv \mu_k\, T_k(\tfrac{t}{\Lambda}) \equiv \mu_k \cos(k \cos^{-1}(\tfrac{t}{\Lambda}))$$

sind die Extremalpunkte von t_k durch

$$z_j := 1 - i\theta_j,\ \theta_j := \Lambda \cos \tfrac{j\pi}{k},\ j = 0,1,\ldots,k,$$

gegeben, und es gilt

$$t_k(z_j) = (-1)^j \mu_k \quad \text{für } 0 \leq j \leq k;$$

für später sei noch festgehalten, daß die θ_j symmetrisch zu 0 liegen:

(16.19)
$$\theta_{k-j} = -\theta_j \quad \text{für } 0 \leq j \leq k.$$

Um $t_k \neq p_k$ nachzuweisen, genügt es wegen (16.4)b) ein Polynom q mit grad $q \leq k$ und $q(0)=0$ zu finden, welches

(16.20)
$$\mathrm{Re}\,((-1)^j\, \bar{\mu}_k\, q(z_j)) > 0,\ j = 0,1,\ldots,k,$$

erfüllt. Wir setzen q an in der Form

$$q(z) \equiv \beta_o + \beta_1(1-z) + \beta_2(1-z)^2 + \cdots + \beta_k(1-z)^k,$$

$$\beta_o, \beta_1, \ldots, \beta_k \in \mathbb{R},$$

und fordern - um $q(0)=0$ sicherzustellen -

$$\beta_o + \beta_1 + \beta_2 + \cdots + \beta_k = 0. \tag{16.21}$$

1. Fall: Sei $k=2l$, $l \in \mathbb{N}$, gerade.

Nach (16.18) ist $\mu_k = \overline{\mu}_k \in \mathbb{R}\setminus\{0\}$ und für (16.20) ist lediglich $\operatorname{Re} q(z_j)$ von Bedeutung. Für $t \in \mathbb{R}$ stellt

$$\operatorname{Re} q(1-it) \equiv \beta_o - \beta_2 t^2 + \beta_4 t^4 - \cdots + (-1)^l \beta_{2l} (t^2)^l$$

ein reelles Polynom vom Grad $\leq l$ in t^2 dar, und die Koeffizienten $\beta_o, \beta_2, \beta_4, \ldots, \beta_{2l} \in \mathbb{R}$ können so festgelegt werden, daß

$$r(t) \equiv \operatorname{Re} q(1-it) \equiv \mu_k \prod_{j=1}^{l} (t^2 - \xi_j^2)$$

gilt. $\xi_1, \xi_2, \ldots, \xi_l$ wählen wir dabei so, daß

$$\theta_o > \xi_1 > \theta_1 > \xi_2 > \cdots > \xi_{l-1} > \theta_{l-1} > \xi_l > \theta_l = 0$$

erfüllt ist; $r(t)$ besitzt dann nämlich die $2l$ einfachen Nullstellen $\xi_1, \xi_2, \ldots, \xi_l, -\xi_l, \ldots, -\xi_2, -\xi_1$, die - man beachte (16.19) - zwischen den θ_j, $j=0,\ldots,k$, liegen. Es folgt

$$\operatorname{sgn} \operatorname{Re} q(z_j) = \operatorname{sgn} r(\theta_j) = (-1)^j \operatorname{sgn} \mu_k \qquad \text{für } 0 \leq j \leq k$$

und damit

$$(-1)^j \mu_k \operatorname{Re} q(z_j) > 0 \qquad \text{für } 0 \leq j \leq k,$$

also (16.20). Wir müssen nur noch (16.21) garantieren und können dies etwa durch die Wahl

$$\beta_1 = -\beta_o - \beta_2 - \beta_4 - \cdots - \beta_{2l}, \quad \beta_3 = \beta_5 = \cdots = \beta_{2l-1} = 0$$

erreichen.

2. Fall: Sei $k=2l-1$, $l \in \mathbb{N}$, ungerade.

Nun gilt $\mu_k = i\tilde{\mu}_k$, $\tilde{\mu}_k \in \mathbb{R}$, und wegen

$$\mathrm{Re}\,((-1)^j\,\bar{\mu}_k\,q(z_j)) = (-1)^j\,\tilde{\mu}_k\,\mathrm{Im}\,q(z_j)$$

ist in Hinblick auf (16.20) $\mathrm{Im}\,q(z_j)$ ausschlaggebend. Es werden daher $\beta_1,\beta_3,\beta_5,\ldots,\beta_{2l-1} \in \mathbb{R}$ durch die Forderung

$$\mathrm{Im}\,q(1-it) \equiv t(\beta_1-\beta_3t^2+\beta_5t^4- \cdots + (-1)^{l-1}\beta_{2l-1}(t^2)^{l-1})$$

$$\equiv \tilde{\mu}_k t \prod_{j=1}^{l-1} (t^2 - \xi_j^2)$$

definiert, wobei man jetzt durch

$$\theta_o > \xi_1 > \theta_1 > \cdots > \xi_{l-1} > \theta_{l-1} > 0 > \theta_l$$

und mit (16.19)

$$\mathrm{sgn}\,\mathrm{Im}\,q(1-i\theta_j) = (-1)^j\,\mathrm{sgn}\,\tilde{\mu}_k \quad \text{für } 0 \leq j \leq k$$

sicherstellt. Wählen wir

$$\beta_o = -\beta_1-\beta_3- \cdots -\beta_{2l-1},\ \beta_2 = \beta_4 = \cdots = \beta_{2l-2} = 0,$$

so ist auch (16.21) erfüllt, und wir haben ein q mit den gewünschten Eigenschaften gefunden.

b) und c) ergeben sich durch Anwendung von Satz(16.16) auf unseren Spezialfall (man beachte $m_k<\tilde{\tau}_k$ wegen $t_k \neq p_k$); ferner gilt

$$|T_k(\tfrac{1}{i\Lambda})| = \tfrac{1}{2}((\tfrac{1}{\Lambda} + \sqrt{\tfrac{1}{\Lambda^2}+1}\,)^k + (\tfrac{1}{\Lambda} - \sqrt{\tfrac{1}{\Lambda^2}+1}\,)^k) \quad \text{für } k \geq 0,$$

und mit

$$(\tfrac{1}{\Lambda} \pm \sqrt{\tfrac{1}{\Lambda^2}+1}\,)^2 = \mu \pm \sqrt{\mu^2-1}\,, \quad \mu := \frac{2+\Lambda^2}{\Lambda^2},$$

führt dies bei geradem k zu

$$|T_k(\tfrac{1}{i\Lambda})| = T_{k/2}(\mu).$$

zu d) Nach (16.4)c) besitzt p_k mindestens k+1 Extremalpunkte, und wir nehmen an, es gäbe sogar mehr als k+1. Nach eventuellem Weglassen der Endpunkte $1\pm i\Lambda$ von E_Λ findet man dann k Stück der Form $1+i\theta_j$, wobei

$$\Lambda > \theta_1 > \theta_2 > \cdots > \theta_k > \Lambda$$

gilt. Das durch

$$|p_k(1+it)|^2 \equiv p_k(1+it)p_k(1-it) \equiv q(t)$$

definierte reelle Polynom q ist höchstens vom Grade 2k und nimmt

in den θ_j, $j=1,2,\ldots,k$, lokale Maxima an. In jedem der Intervalle (θ_{j+1},θ_j), $j=1,2,\ldots,k-1$, sowie in $(-\infty,\theta_k)$ und (θ_1,∞) (wegen $q(t)\to\infty$ für $t\to\pm\infty$) liegt jeweils wenigstens ein lokales Minimum. q' müßte also mindestens $2k+1$ Nullstellen haben, und wir sind auf einen Widerspruch gestoßen. Somit besitzt p_k genau $k+1$ Extremalpunkte und zwei davon sind $1\pm i\Lambda$, d.h. i)ii) sind gezeigt. iii) folgt aus i), da die Extremalpunkte von $p_k\in\Pi_k$ in konjugiert komplexen Paaren auftreten.

zu e) Ein Punkt $a = \mathrm{Re}\, a + i\,\mathrm{Im}\, a$ aus $\mathbb{C}$ geht bei Spiegelung an der Geraden $\mathrm{Re}\, z = 1$ in $\hat{a} = 2 - \mathrm{Re}\, a + i\,\mathrm{Im}\, a$ über; offensichtlich gilt $|z-a|=|z-\hat{a}|$ für alle z mit $\mathrm{Re}\, z = 1$, sowie $|a|<|\hat{a}|$ falls $\mathrm{Re}\, a < 1$. Wir nehmen an, p_k hätte Nullstellen in $\{z \mid \mathrm{Re}\, z < 1\}$; ersetzt man in der Darstellung

$$p_k(z) \equiv \prod_{j=1}^{k} \frac{z_j - z}{z_j}$$

gerade jene z_j mit $\mathrm{Re}\, z_j < 1$ durch $\hat{z}_j$, so ergibt sich ein $\hat{p}_k\in\Pi_k$ mit

$$\max_{z\in E_\Lambda} |\hat{p}_k(z)| < m_k.$$

zu f)i) Für $k=1$ erhalten wir

$$(m_1)^2 = \min_{\sigma\in\mathbb{R}} \max_{-\Lambda\le t\le\Lambda} |1 + \sigma(1+it)|^2 = \min_{\sigma\in\mathbb{R}} ((1+\sigma)^2 + \sigma^2\Lambda^2) = \frac{\Lambda^2}{1+\Lambda^2},$$

wobei das Minimum genau für $\sigma^* = -\frac{1}{1+\Lambda^2}$ angenommen wird.

ii) Die Polynome aus Π_2 lassen sich in der Form

$$p_{\sigma,\nu}(z) \equiv 1 + (\sigma-\nu)z + \nu z^2, \quad \sigma,\nu\in\mathbb{R},$$

parametrisieren; es gilt

$$|p_{\sigma,\nu}(1+it)|^2 = (1+\sigma-\nu t^2)^2 + t^2(\sigma+\nu)^2, \quad -\Lambda \le t \le \Lambda,$$

und das Maximum der rechten Seite - da eine nach oben geöffnete Parabel in t^2 - wird für $t^2=0$ oder $t^2=\Lambda^2$ angenommen. Aus d) ergibt sich $D_2=\{1, 1+i\Lambda, 1-i\Lambda\}$, und als Optimalpolynome kommen nur die $p_{\sigma,\nu}$ mit

$$(1+\sigma)^2 = |p_{\sigma,\nu}(1)|^2 = |p_{\sigma,\nu}(1\pm i\Lambda)|^2$$
$$= (1+\sigma)^2 + \Lambda^2(\sigma^2 + (1+\Lambda^2)\nu^2 - 2\nu)$$

in Frage. Mit $\Gamma := \frac{1}{1+\Lambda^2}$ (<1) erhält diese Bedingung die Form

$$\sigma^2 = \Gamma - \frac{1}{\Gamma}(\nu - \Gamma)^2,$$

und es resultiert

$$(m_2)^2 = \min_{\substack{\sigma,\nu \in \mathbb{R} \\ \sigma^2 = \Gamma - \frac{1}{\Gamma}(\nu-\Gamma)^2}} (1+\sigma)^2 = (1-\sqrt{\Gamma})^2$$

mit $\sigma^* = -\sqrt{\Gamma}$, $\nu^* = \Gamma$. □

III. NUMERISCHE BEISPIELE

17. Allgemeines. Modellprobleme

Als Beispiele, die das numerische Verhalten der verschiedenen Algorithmen illustrieren sollen, dienen uns symmetrische bzw. unsymmetrische lineare Gleichungssysteme, wie sie bei der Diskretisierung der elliptischen Differentialgleichung

$$(17.1) \qquad -\Delta u - \sigma u = f(x,y) \ , \ (x,y) \in \Omega$$

bzw.

$$(17.2) \qquad -\Delta u + \sigma u_x = f(x,y) \ , \ (x,y) \in \Omega$$

- mit jeweils vorgegebenen Randwerten $u|_{\partial\Omega}=g$ - anfallen. Dabei sei $\sigma \in \mathbb{R}$ eine Konstante und $\Omega=\{(x,y) \mid 0<x,y<1\}$ das Einheitsquadrat des $\mathbb{R}^2$. Nach Wahl einer natürlichen Zahl l überziehen wir $\Omega \cup \partial\Omega$ mit dem Gitter $(x_i,y_j)=(ih,jh)$, $i,j=0,1,\ldots,l+1$, der Maschenweite $h=1/(l+1)$ und versuchen nun Näherungen U_{ij} der Lösungen $u(x_i,y_j)$ von (17.1) bzw. (17.2) an diesen Stellen zu berechnen. Dazu wird in den inneren Gitterpunkten (x_i,y_j) $-\Delta u$ durch die Fünf-Punkte-Formel

$$(4U_{ij} - U_{i-1,j} - U_{i+1,j} - U_{i,j-1} - U_{i,j+1})/h^2$$

bzw. u_x durch

$$(U_{i+1,j} - U_{i-1,j})/2h$$

ersetzt, und anstelle von (17.1) bzw. (17.2) erhält man das lineare Gleichungssystem

$$(17.3) \quad (4 - \sigma h^2)U_{ij} - U_{i-1,j} - U_{i+1,j} - U_{i,j-1} - U_{i,j+1} = h^2 f(x_i,y_j), \qquad i,j = 1,2,\ldots,l,$$

bzw.

$$(17.4) \quad 4U_{ij} - (1 + \tfrac{\sigma h}{2})U_{i-1,j} - (1 - \tfrac{\sigma h}{2})U_{i+1,j} - U_{i,j-1} - U_{i,j+1} = h^2 f(x_i,y_j), \qquad i,j = 1,2,\ldots,l,$$

- wobei jeweils $U_{ij}=g(x_i,y_j)$ für $(x_i,y_j)\in\partial\Omega$ zu setzen ist - für die $n:=l^2$ Unbekannten U_{ij}, $i,j=1,2,\ldots,l$, welche wir in der Anordnung

$$U_{11},U_{21},\ldots,U_{l1},U_{12},\ldots,U_{l2},\ldots,U_{1l},\ldots,U_{ll}$$

zu einem Vektor der Länge n zusammenfassen.
Durch die symmetrische n×n-Matrix

$$A = A_o - \sigma h^2 I, \tag{17.5}$$

dabei entspricht

$$A_o = \begin{bmatrix} T & -I & & \sigma \\ -I & T & \ddots & \\ & \ddots & \ddots & -I \\ \sigma & & -I & T \end{bmatrix} \tag{17.6}$$

mit den l×l-Blöcken

$$T = \begin{bmatrix} 4 & -1 & & \sigma \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ \sigma & & -1 & 4 \end{bmatrix}$$

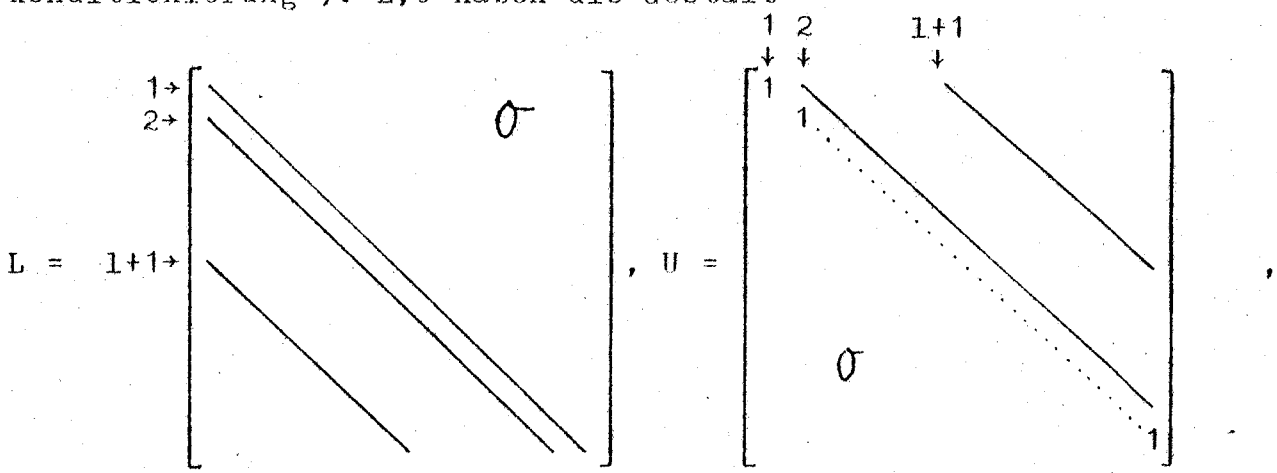
gerade dem Fall σ=0, ist dann die Koeffizientenmatrix des Gleichungssystems (17.3) gegeben, welches wir als <u>symmetrisches Modellproblem</u> bezeichnen und als Testbeispiel für OD-, STOD-Algorithmus und ihre prekonditionierten Varianten verwenden wollen. Die Eigenwerte λ_{ij} (und die zugehörigen Eigenvektoren) der positiv definiten Matrix A_o lassen sich explizit angeben (siehe etwa [34]), nämlich

$$\lambda_{ij} = 2\left(2 - \cos\frac{i\pi}{l+1} - \cos\frac{j\pi}{l+1}\right), i,j = 1,2,\ldots,l; \tag{17.7}$$

A besitzt die Eigenwerte $\lambda_{ij}-\sigma h^2$ und ist also für geeignetes σ>0 indefinit und nichtsingulär. Für unsere Tests wählen wir für l,n,σ Werte, die auch von Chandra [6] benutzt wurden: Zum einen

$l=15$, $n=225$, zum anderen $l=31$, $n=961$ und beidemal $\sigma=30$ und $\sigma=90$. Im Fall $\sigma=30$ ist sowohl für $n=225$, als auch für $n=961$ genau der eine Eigenwert $\lambda_{11}-\sigma h^2$ von A negativ, im Fall $\sigma=90$ liegen jeweils die vier negativen Eigenwerte $\lambda_{ij}-\sigma h^2$, $i,j=1,2$, vor.

Als Prekonditionierungsmatrix QQ^T für das symmetrische Modellproblem verwenden wir, dem Vorschlag Chandras [6] folgend, die positiv definite Matrix LU, welche man bei der näherungsweisen Faktorisierung von $A_o \approx LU$ nach Dupont, Kendall und Rachford [9] erhält ("DKR-Prekonditionierung"). L,U haben die Gestalt



die beiden Subdiagonalen von L stimmen ferner mit denjenigen von A_o überein, bestehen also nur aus -1 bzw. 0, so daß zum Lösen eines Gleichungssystems $LUq=p$ lediglich etwa $3n$ Multiplikationen benötigt werden.

Als Koeffizientenmatrix des Systems (17.4) ergibt sich die $n\times n$-Matrix $A=M-N$, deren schiefsymmetrischer Anteil durch

$$N = \frac{\sigma h}{2}\begin{bmatrix} S & & \sigma \\ & S & \\ \sigma & & \ddots \\ & & & S \end{bmatrix}$$

mit den $l\times l$-Blöcken

$$S = -S^T = \begin{bmatrix} 0 & -1 & & \sigma \\ 1 & 0 & \ddots & \\ & \ddots & \ddots & -1 \\ \sigma & & 1 & 0 \end{bmatrix}$$

gegeben ist, während die Matrix (17.6) gerade den symmetrischen Anteil darstellt: $M=A_o$. A ist somit positiv reell und liefert unser Testbeispiel, das unsymmetrische Modellproblem, für die Verfahren aus den Abschnitten 14 und 15. Für n,σ wählen wir die Werte $n=49$, $n=225$, $n=961$ und jeweils $\sigma=1$, $\sigma=10$, $\sigma=100$; als Prozedur für das Lösen von Gleichungssystemen mit M als Koeffizientenmatrix wird der Algorithmus von Buneman (eine Beschreibung findet man z.B. in [33]) benutzt.

Zu der rechten Seite unserer Testsysteme $Ax=b$ gelangen wir, indem wir zunächst die Lösung $\bar{x}$ aus n Zufallszahlen (aus dem Intervall $[-1,1]$) vorgeben und dann $b:=A\bar{x}$ setzen. In jedem Iterationsschritt werden stets der relative euklidische Fehler

$$E_k := \|x_k - \bar{x}\| / \|x_o - \bar{x}\|$$

und das relative Residuum

$$R_k := \|r_k\| / \|r_o\|$$

berechnet. Für einen Teil der Beispiele wurde der Verlauf von E_k (durchgehende Linie ———) und R_k (unterbrochene Linie – – –) geplottet. Als Startwert wurde jeweils $x_o=0$ gewählt, und gestoppt wurden die Algorithmen, sobald $R_k \leq 10^{-7}$ erreicht war.

18. Symmetrisches Modellproblem ohne Prekonditionierung

Der OD-Algorithmus in der Form (2.2) ist instabil: Die Fehler E_k sind nur bis zu einer Iterationsnummer $\hat{k}$ monoton fallend, für $k>\hat{k}$ wachsen sie wieder an und explodieren schließlich. Die Genauigkeit des besten Näherungswertes $x_{\hat{k}}$ ist sehr bescheiden, wie man folgender Tabelle entnehmen kann, in der die Werte von $\hat{k} \mid E_{\hat{k}}$ aufgeführt sind:

	σ=30	σ=90
n=225	47 | 3.82_{10}-5	58 | 1.47_{10}-4
n=961	82 | 2.54_{10}-3	103 | 4.21_{10}-3

Um den Grund für dieses instabile Verhalten zu finden, wurden einige Modifikationen des OD-Algorithmus(2.2) getestet:
In Version (A) wurden die Residuen in jedem Schritt mittels $r_k=b-Ax_k$ berechnet, in (B) wurde die Berechnung der Skalarprodukte $r_k^T p_{k-1}$, $p_k^T p_k$, $p_k^T A p_k$ in doppelter Genauigkeit ausgeführt, und in (C) wurden orthonormale Suchrichtungen verwendet, die mittels einer von Paige [24] vorgeschlagenen Variante des Lanczos-Algorithmus erzeugt wurden.
Es zeigt sich, daß man auf diese Weise die Instabilität nicht beseitigen kann, es ergeben sich lediglich minimal verbesserte Werte für $E_{\hat{k}}$. So erhält man z.B. im Fall n=225, σ=30 für $\hat{k} \mid E_{\hat{k}}$:

(A)	(B)	(C)
47 | 3.59_{10}-5	47 | 3.22_{10}-5	48 | 2.58_{10}-5

Das Scheitern des OD-Algorithmus wird verständlich, wenn man neben α_k stets $\tilde{\alpha}_k=(\bar{x}-x_k)^T p_k / p_k^T p_k$ und damit $\Delta\alpha_k \| p_k \|$, sowie die Größen $c_k^{(1)}$, $c_k^{(2)}$ und

$$c_k := -c_k^{(1)} \Delta\alpha_{k-1} \| p_{k-1} \| - c_k^{(2)} \Delta\alpha_{k-2} \| p_{k-2} \|$$

aus der Formel (7.6), auf welche uns die Diskussion in Abschnitt 7

geführt hat, berechnet. Z.B. ergibt sich für n=961, σ=30:

k	5	10	20	30	40
$\|\alpha_k\| \|\|p_k\|\|$	2.49	1.02	4.84_{10}-1	1.48_{10}-1	1.23_{10}-1
$\|\Delta\alpha_k\| \|\|p_k\|\|$	3.62_{10}-7	5.46_{10}-6	1.05_{10}-4	5.63_{10}-4	2.55_{10}-3
$\|\Delta\alpha_k \|\|p_k\|\| - c_k\|$	1.99_{10}-9	3.26_{10}-10	4.82_{10}-10	1.65_{10}-10	4.82_{10}-10
$c_k^{(1)}$	2.64	2.23	2.09	2.00	1.98
$c_k^{(2)}$	0.94	1.00	1.03	1.00	0.99

50	60	70	80	82(=$\hat{k}$)	90
1.25_{10}-1	2.30_{10}-2	2.08_{10}-2	2.55_{10}-2	2.11_{10}-2	4.26_{10}-2
7.20_{10}-3	7.74_{10}-3	3.43_{10}-3	7.89_{10}-3	9.84_{10}-3	4.07_{10}-2
5.37_{10}-10	9.87_{10}-11	8.46_{10}-11	4.75_{10}-11	3.80_{10}-11	1.22_{10}-10
1.93	2.19	2.06	1.95	2.15	2.17
0.95	1.03	1.07	1.02	1.01	1.01

Bereits nach wenigen Iterationen wird also - in Einklang mit (7.6) - $\Delta\alpha_k \|\|p_k\|\|$ fast vollständig durch c_k beschrieben. Die Faktoren $c_k^{(1)}$, $c_k^{(2)}$ bewirken, daß die beiden letzten Schrittweitenfehler meist verstärkt werden (zu beachten ist, daß die $\Delta\alpha_k \|\|p_k\|\|$ fast durchweg alternierendes Vorzeichen aufweisen) und $|\Delta\alpha_k| \|\|p_k\|\| \approx |c_k|$ mit wachsendem k ansteigt, bis schließlich in der Gegend von $\hat{k}$ der relative Schrittweitenfehler $|\Delta\alpha_k| / |\alpha_k|$ so groß wird, daß die für $k>\hat{k}$ weiter berechneten Näherungswerte x_k unsinnig werden.

Als einfache Möglichkeit, den OD-Algorithmus(2.2) zu stabilisieren, bietet sich die Verwendung der folgenden Restart-Technik an: Man

speichert den Näherungswert x_{Γ_k} mit dem bislang kleinsten Residuum, d.h. $\|r_{\Gamma_k}\| = \min_{0\leq j\leq k} \|r_j\|$, und startet den Algorithmus mit diesem x_{Γ_k} als Startwert neu, sobald

$$\|r_{k+1}\| > \beta \cdot \min_{0\leq j\leq k} \|r_j\| \tag{18.1}$$

(mit einer geeigneten Schranke $\beta>1$) eintritt. Es ergeben sich dann für

k_o = erster Index mit $R_{k_o} \leq 10^{-7}$,

k_{REST} = Iteration, nach der ein Restart erfolgt,

die Werte:

	σ=30			σ=90		
	k_o	k_{REST}	β	k_o	k_{REST}	β
n=225	78	51	6	114	44,107	12
n=961	178	89	12	228	112	20

Der Verlauf von E_k, R_k für die Fälle n=961, σ=30 bzw. σ=90 ist in Fig. 1a bzw. 2a dargestellt, "□ " kennzeichnet dabei einen Restart.

Wesentlich bessere Resultate liefert der stabile STOD-Algorithmus (7.7), welcher

	σ=30	σ=90
n=225	56 (54)	73 (71)
n=961	114 (108)	146 (141)

Iterationsschritte benötigt, um $R_k \leq 10^{-7}$ ($E_k \leq 10^{-6}$) zu erreichen.

Zeichnungen zu n=961, σ=30 bzw. σ=90 findet man in Fig. 1b bzw. 2b; ein Vergleich mit Fig. 1a,2a zeigt deutlich die Überlegenheit des STOD-Algorithmus gegenüber dem OD-Algorithmus.

Der MCR-Algorithmus(5.4) braucht für die Reduzierung der Fehler auf $R_k \leq 10^{-7}$ ($E_k \leq 10^{-6}$)

	σ=30	σ=90
n=225	50 (49)	66 (66)
n=961	94 (96)	121 (124)

Iterationen, also etwas weniger als der STOD-Algorithmus. Man kann aber wohl sagen, daß der STOD-Algorithmus einem Vergleich mit dem MCR-Algorithmus standhält.

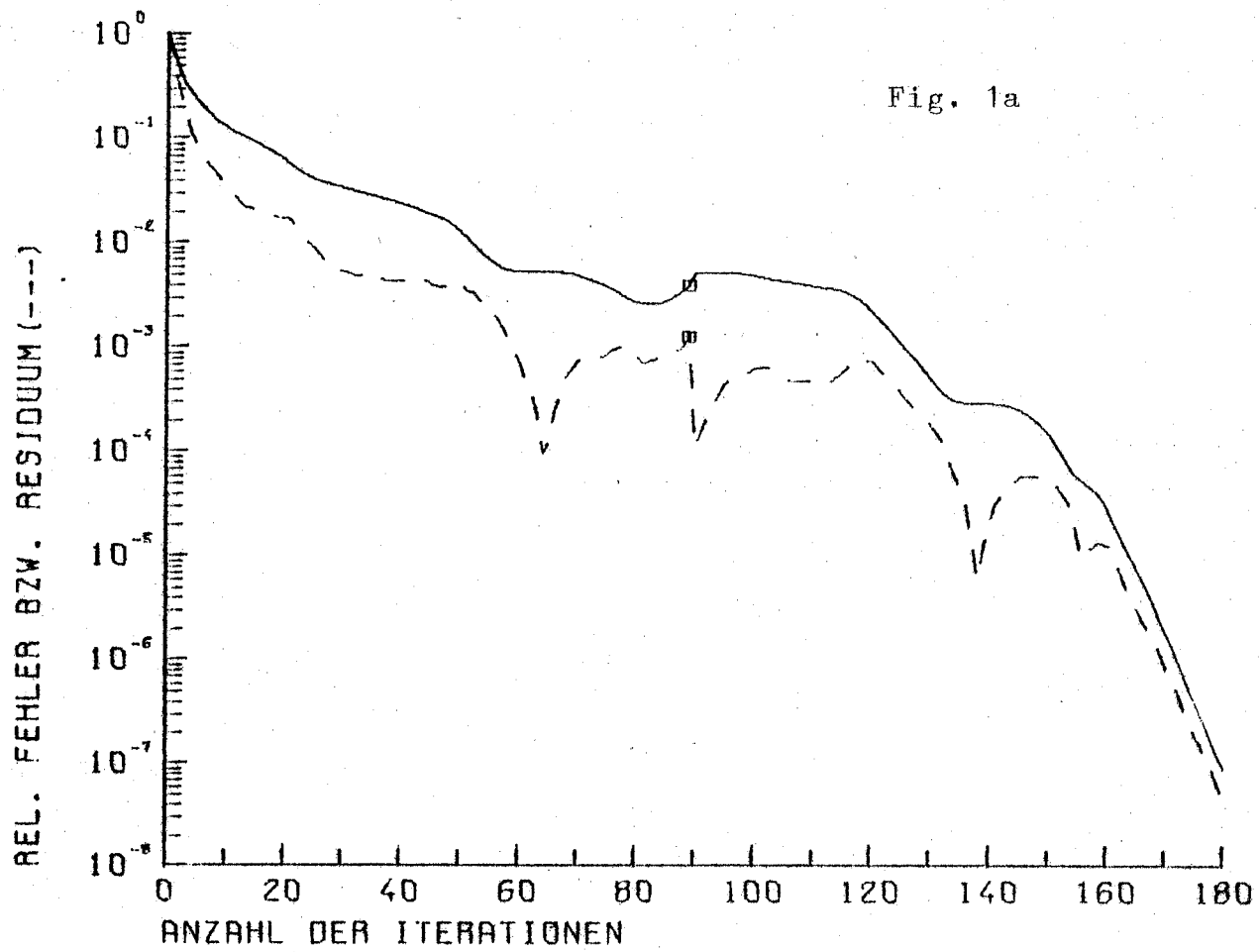

REL. FEHLER BZW. RESIDUUM (---)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
0
20
40
60
80
100
120
140
160
180
ANZAHL DER ITERATIONEN

Fig. 1a

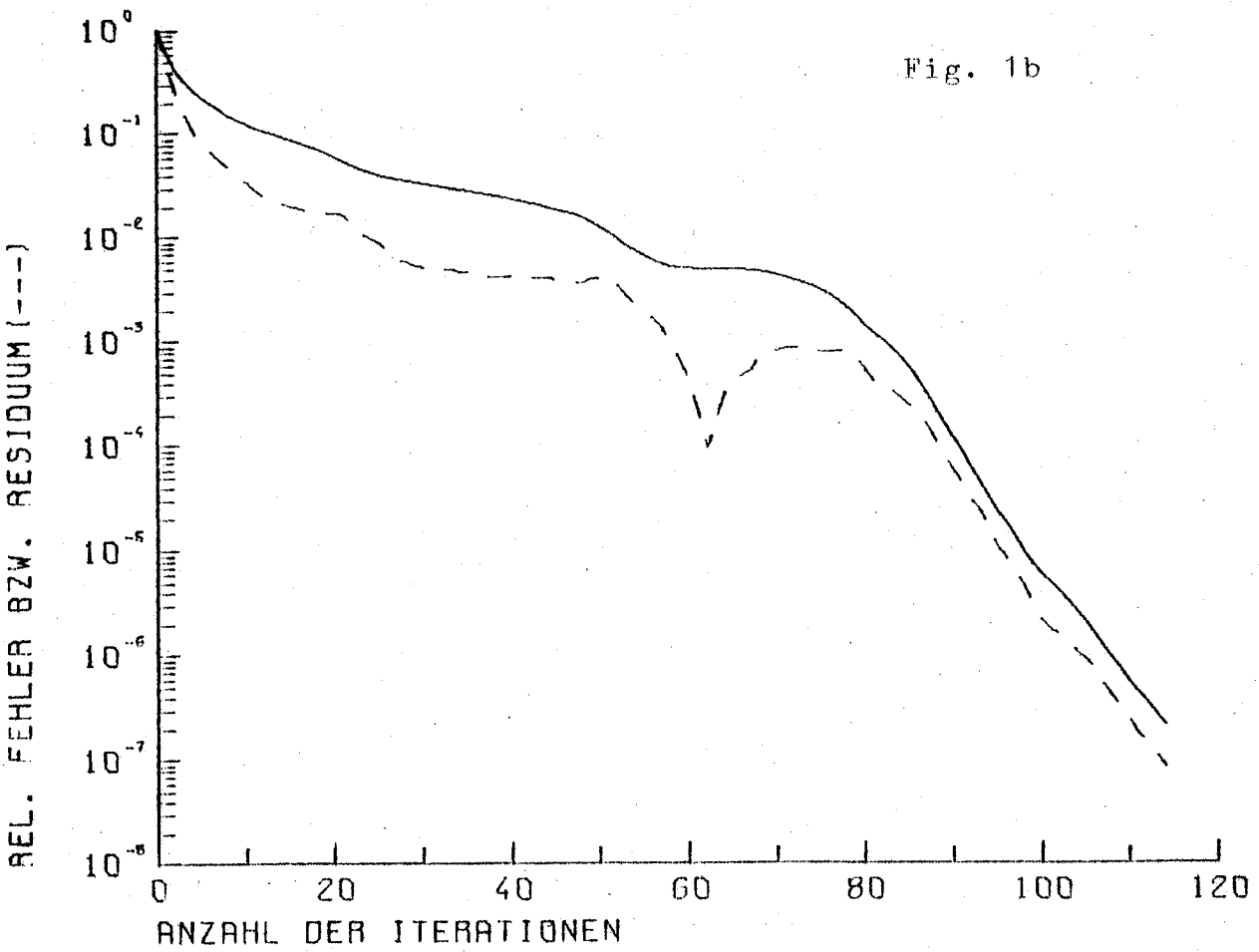

REL. FEHLER BZW. RESIDUUM (---)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
0
20
40
60
80
100
120
ANZAHL DER ITERATIONEN

Fig. 1b

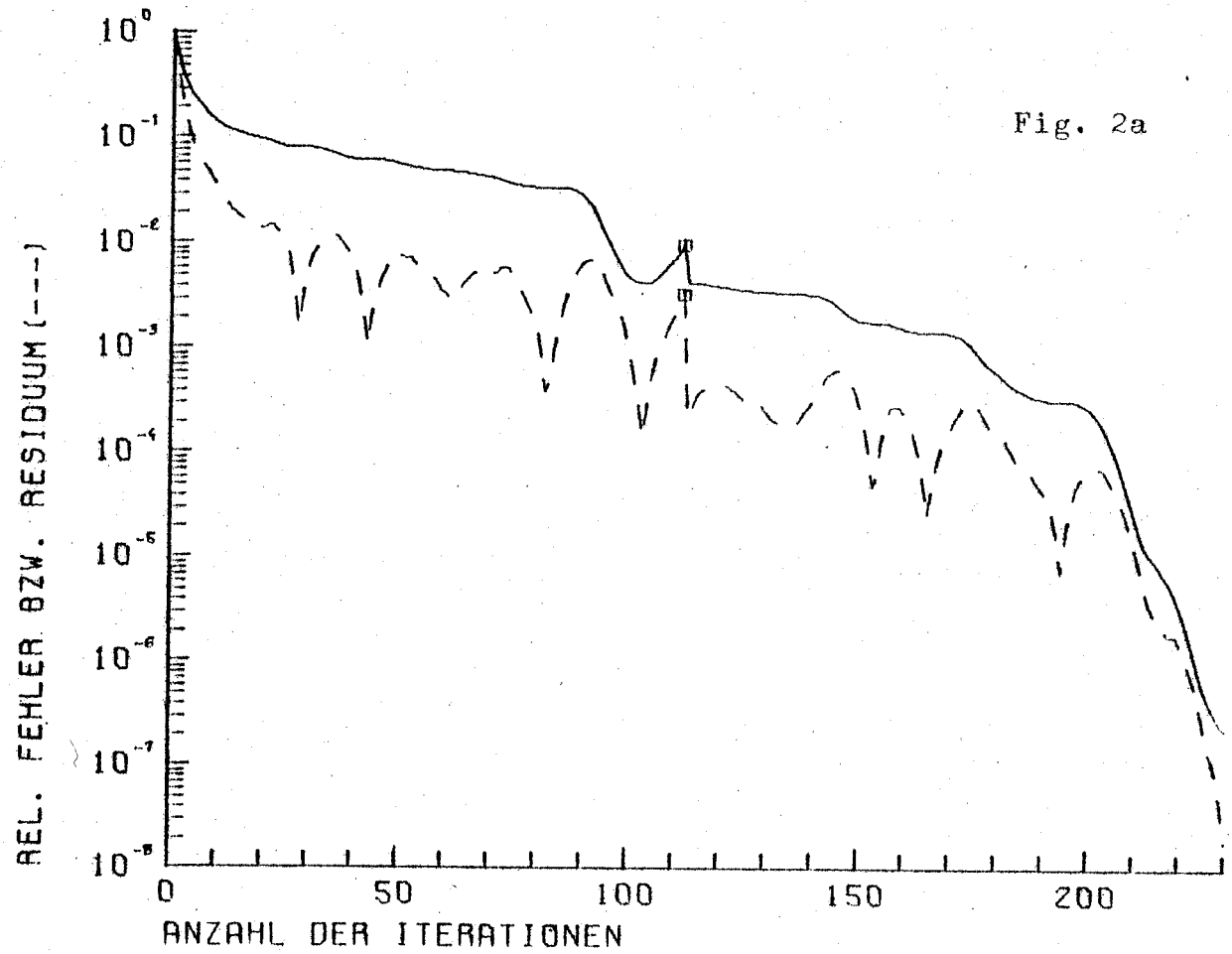

REL. FEHLER BZW. RESIDUUM (---)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
0
50
100
150
200
ANZAHL DER ITERATIONEN

Fig. 2a

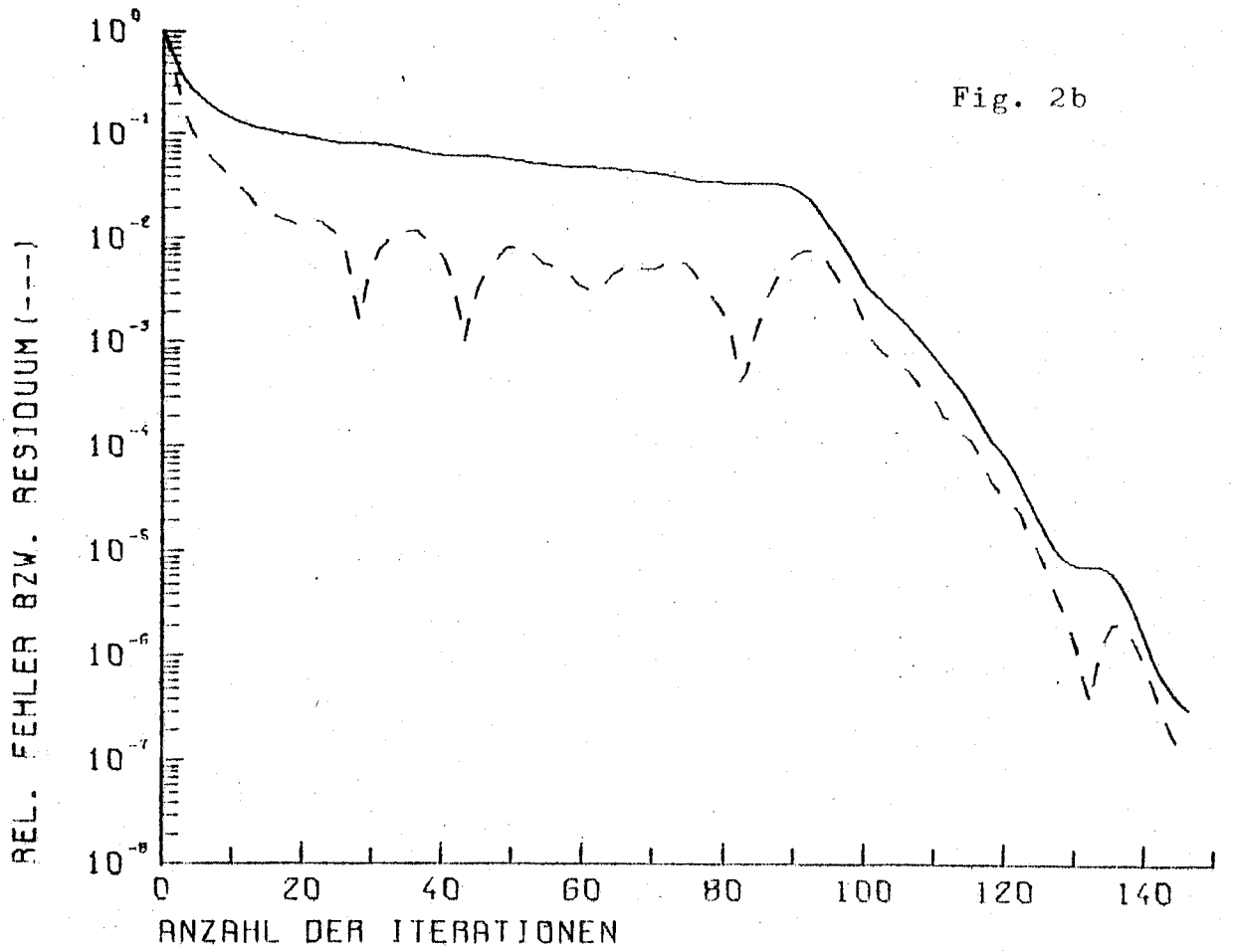

REL. FEHLER BZW. RESIDUUM (---)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
0
20
40
60
80
100
120
140
ANZAHL DER ITERATIONEN

Fig. 2b

19. Symmetrisches Modellproblem mit DKR-Prekonditionierung

Der PCOD-Algorithmus(8.4) ist wiederum instabil, so daß man ihn in Verbindung mit dem Restart-Kriterium (18.1) implementieren muß, um überhaupt zu einem ersten Index k_o mit $R_{k_o} \leq 10^{-7}$ zu gelangen; es ergeben sich dann folgende Werte:

	σ=30			σ=90		
	k_o	k_{REST}	β	k_o	k_{REST}	β
n=225	26	19	5	59	35	8
n=961	37	25	2	83	42	3

Dagegen zeigt der PCSTOD-Algorithmus(8.5) ein stabiles Verhalten, und $R_k \leq 10^{-7}$ ($E_k \leq 10^{-6}$) ist nach

	σ=30	σ=90
n=225	21 (20)	40 (39)
n=961	30 (30)	54 (56)

Iterationsschritten erreicht. Der Verlauf von E_k, R_k ist für die Fälle n=961, σ=30 bzw. σ=90 in Fig. 3a bzw. 4a (PCOD-Algorithmus) und in Fig. 3b bzw. 4b (PCSTOD-Algorithmus) dargestellt.

Für die Zahl der Iterationen, die man mit der prekonditionierten Version des MCR-Algorithmus(5.4) zur Reduktion der Fehler auf $R_k \leq 10^{-7}$ ($E_k \leq 10^{-6}$) benötigt, ergeben sich mit

	σ=30	σ=90
n=225	18 (18)	37 (35)
n=961	25 (26)	49 (50)

ähnliche Werte wie für das PCSTOD-Verfahren.

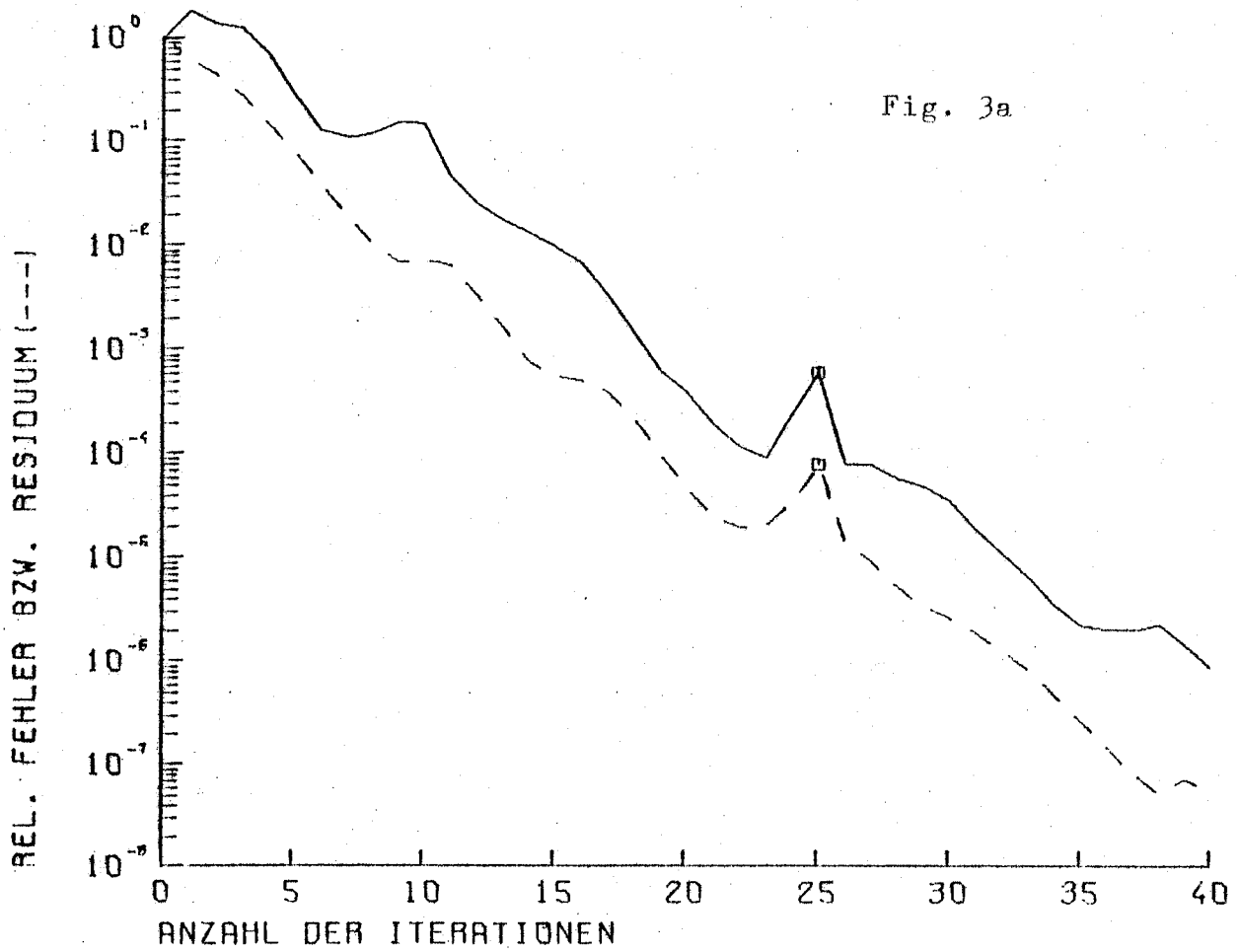

REL. FEHLER BZW. RESIDUUM (---)
ANZAHL DER ITERATIONEN
0
5
10
15
20
25
30
35
40

Fig. 3a

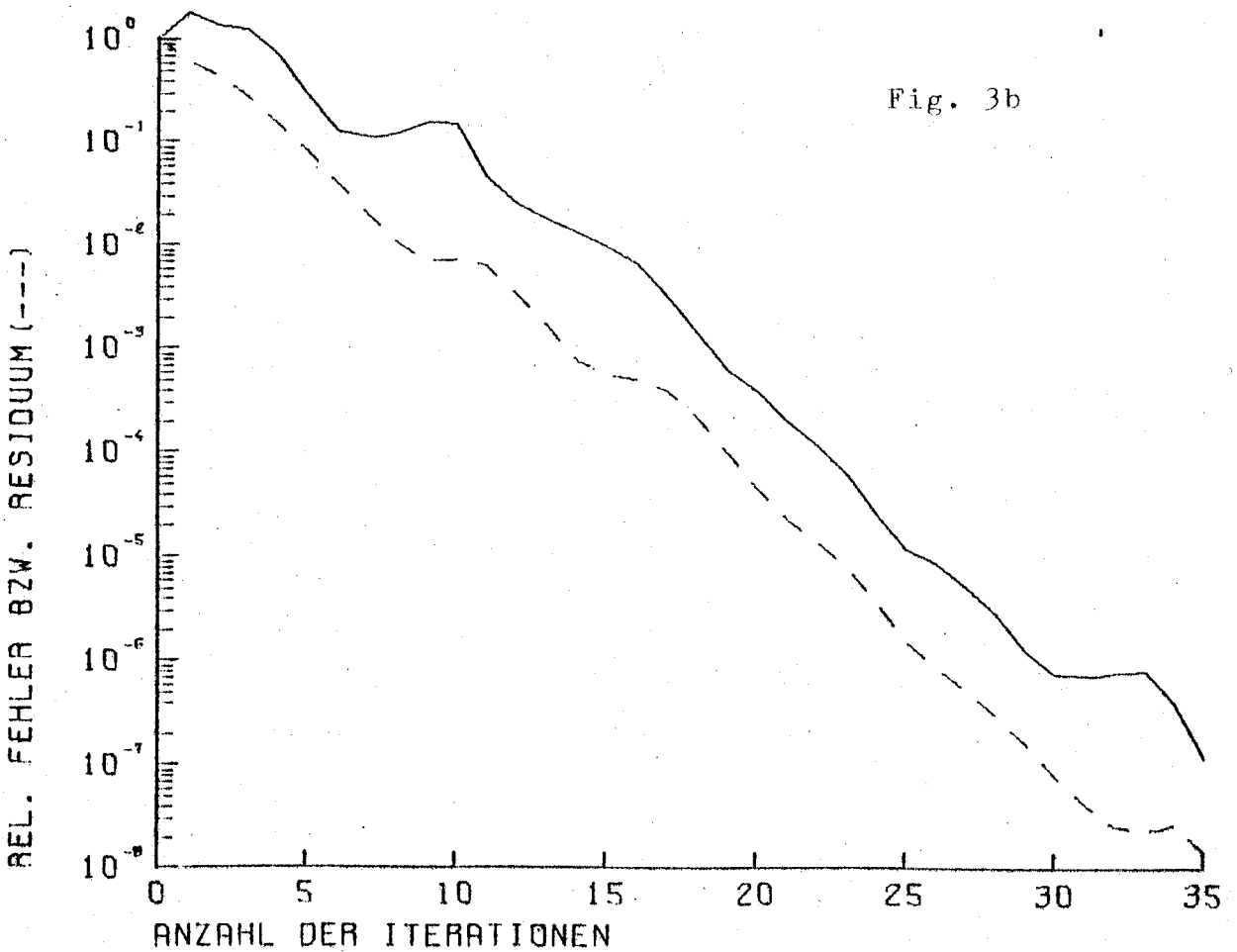

REL. FEHLER BZW. RESIDUUM (---)
ANZAHL DER ITERATIONEN
0
5
10
15
20
25
30
35

Fig. 3b

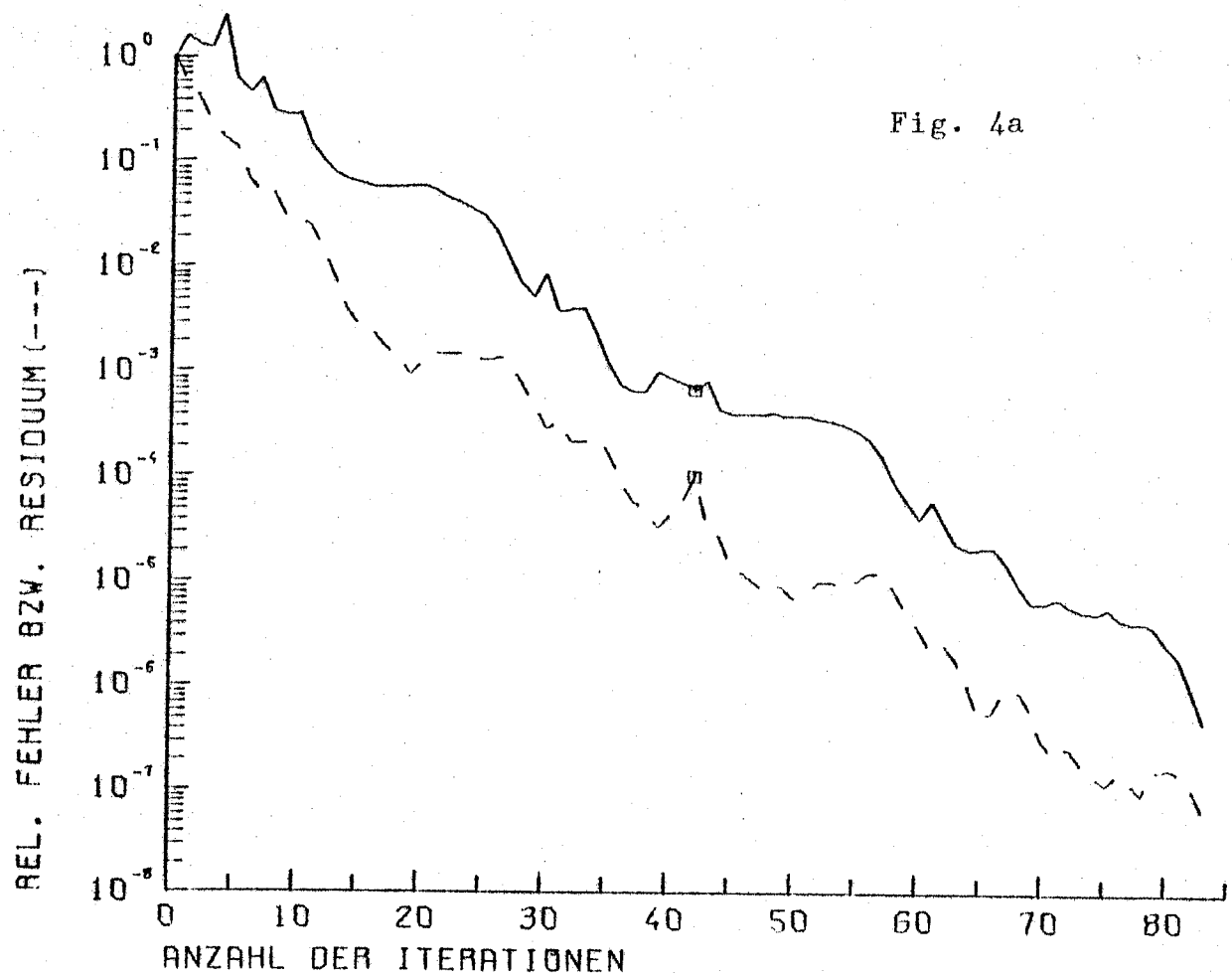

REL. FEHLER BZW. RESIDUUM (---)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
0 10 20 30 40 50 60 70 80
ANZAHL DER ITERATIONEN

Fig. 4a

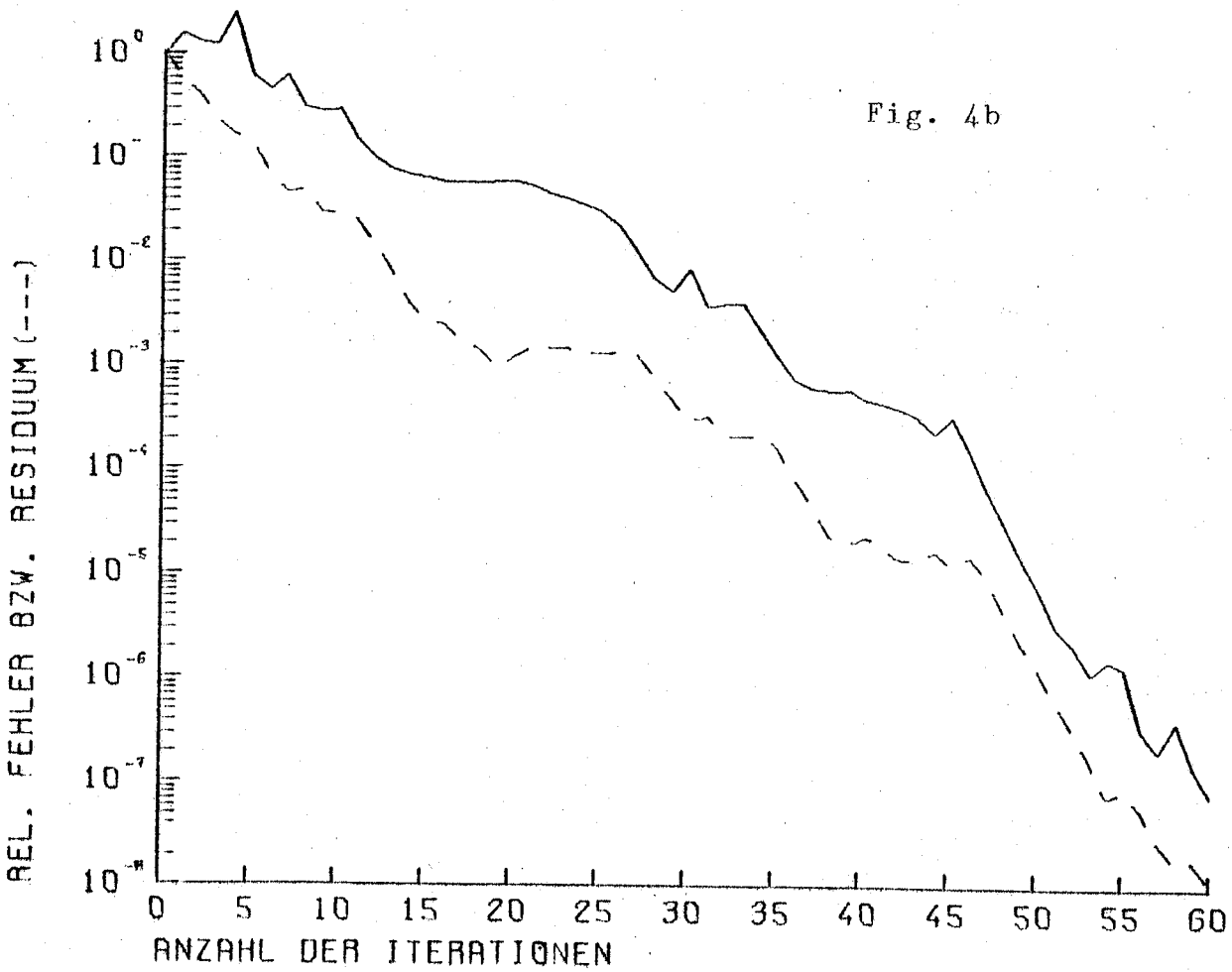

REL. FEHLER BZW. RESIDUUM (---)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
0 5 10 15 20 25 30 35 40 45 50 55 60
ANZAHL DER ITERATIONEN

Fig. 4b

20. Weitere Beispiele symmetrischer Matrizen

a) Singuläres symmetrisches Modellproblem

Die Matrix (17.5) $A=A_o-\sigma h^2 I$ des symmetrischen Modellproblems wird für $\sigma h^2=\lambda_{ij}$ (= einer der Eigenwerte (17.7) von A_o) singulär und liefert uns ein Testbeispiel für das numerische Verhalten des STOD-Algorithmus im singulären Fall. Als Werte für n,σ wählen wir n=225 bzw. n=961, sowie jeweils $\sigma=\lambda_{12}/h^2$ und $\sigma=\lambda_{22}/h^2$; durch $b:=Az$ - mit einem Vektor z aus Zufallszahlen - wird die Konsistenz von $Ax=b$ garantiert. Da nach Satz(10.5)j) die Lösung $\bar{x}=\tilde{x}_o+A^+b$, auf welche das STOD-Verfahren stößt, von x_o abhängt, wurden zwei verschiedene Startwerte getestet.
Es zeigt sich, daß der STOD-Algorithmus(7.7) auch im Fall dieses singulären symmetrischen Modellproblems stabil bleibt und bei Verwendung von

$$x_o = (0,0,\ldots,0)^T$$

	$\sigma = \lambda_{12}/h^2$	$\sigma = \lambda_{22}/h^2$
n=225	54 (51)	63 (56)
n=961	108 (102)	120 (115)

bzw. $x_o = (1,1,\ldots,1)^T$

	$\sigma = \lambda_{12}/h^2$	$\sigma = \lambda_{22}/h^2$
n=225	54 (50)	64 (56)
n=961	108 (101)	128 (121)

Iterationsschritte benötigt, um $R_k \leq 10^{-7}$ ($E_k \leq 10^{-6}$) zu erreichen. Für n=961, $\sigma=\lambda_{22}/h^2$ ist der Verlauf von E_k, R_k in Fig. 5a ($x_o=0$) bzw. 5b ($x_o=(1,\ldots,1)^T$) dargestellt.

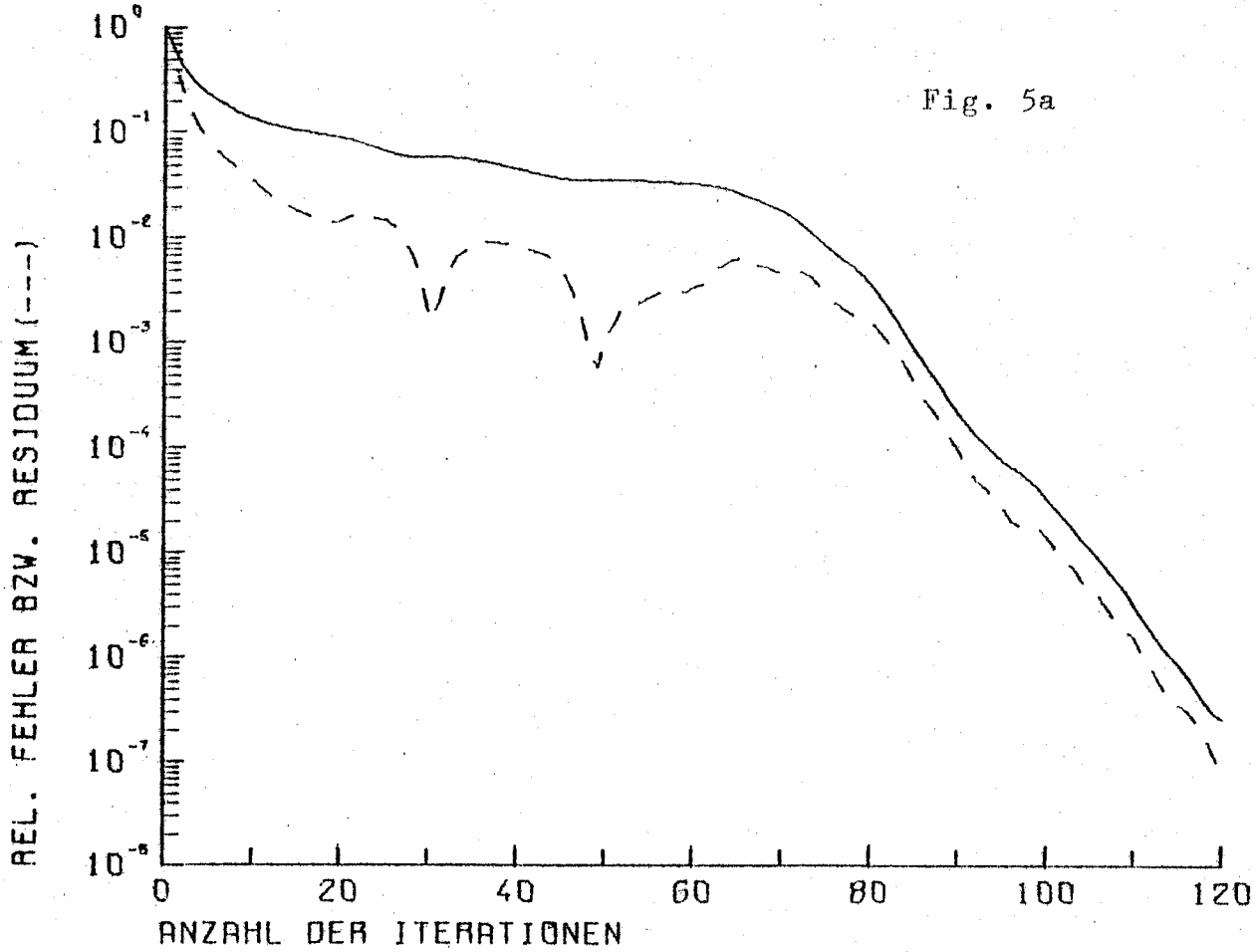

REL. FEHLER BZW. RESIDUUM(---)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
0
20
40
60
80
100
120
ANZAHL DER ITERATIONEN

Fig. 5a

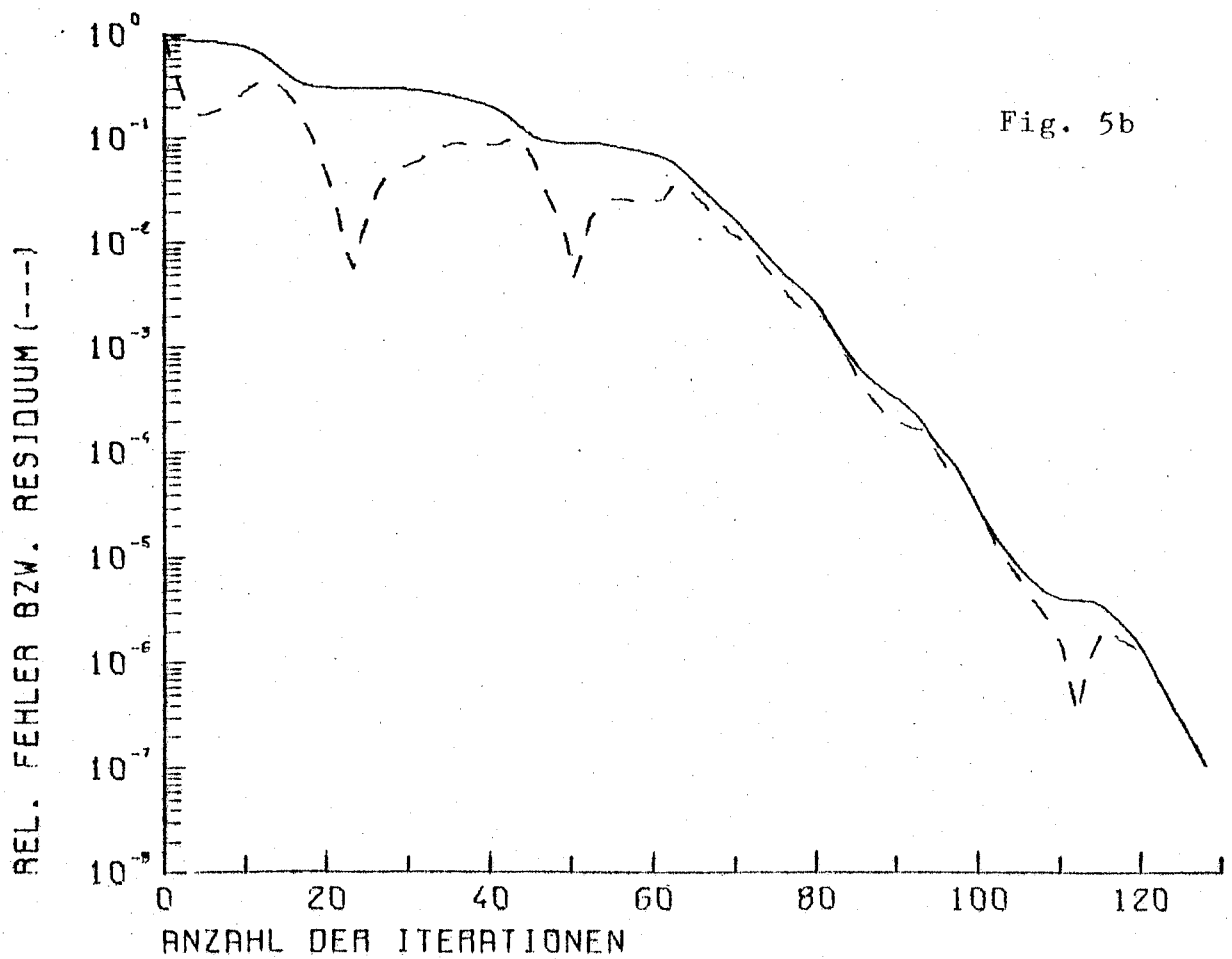

REL. FEHLER BZW. RESIDUUM(---)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
0
20
40
60
80
100
120
ANZAHL DER ITERATIONEN

Fig. 5b

b) Zwei Beispiele zum Entartungsfall

Als Koeffizientenmatrix für unser Testsystem Ax=b dient die 100×100-Matrix

$$A_{1/2} := \mathrm{diag}(1,\sqrt{2},\sqrt{3},\ldots,\sqrt{50},-\sqrt{50},\ldots,-\sqrt{3},-\sqrt{2},-1)$$

bzw.

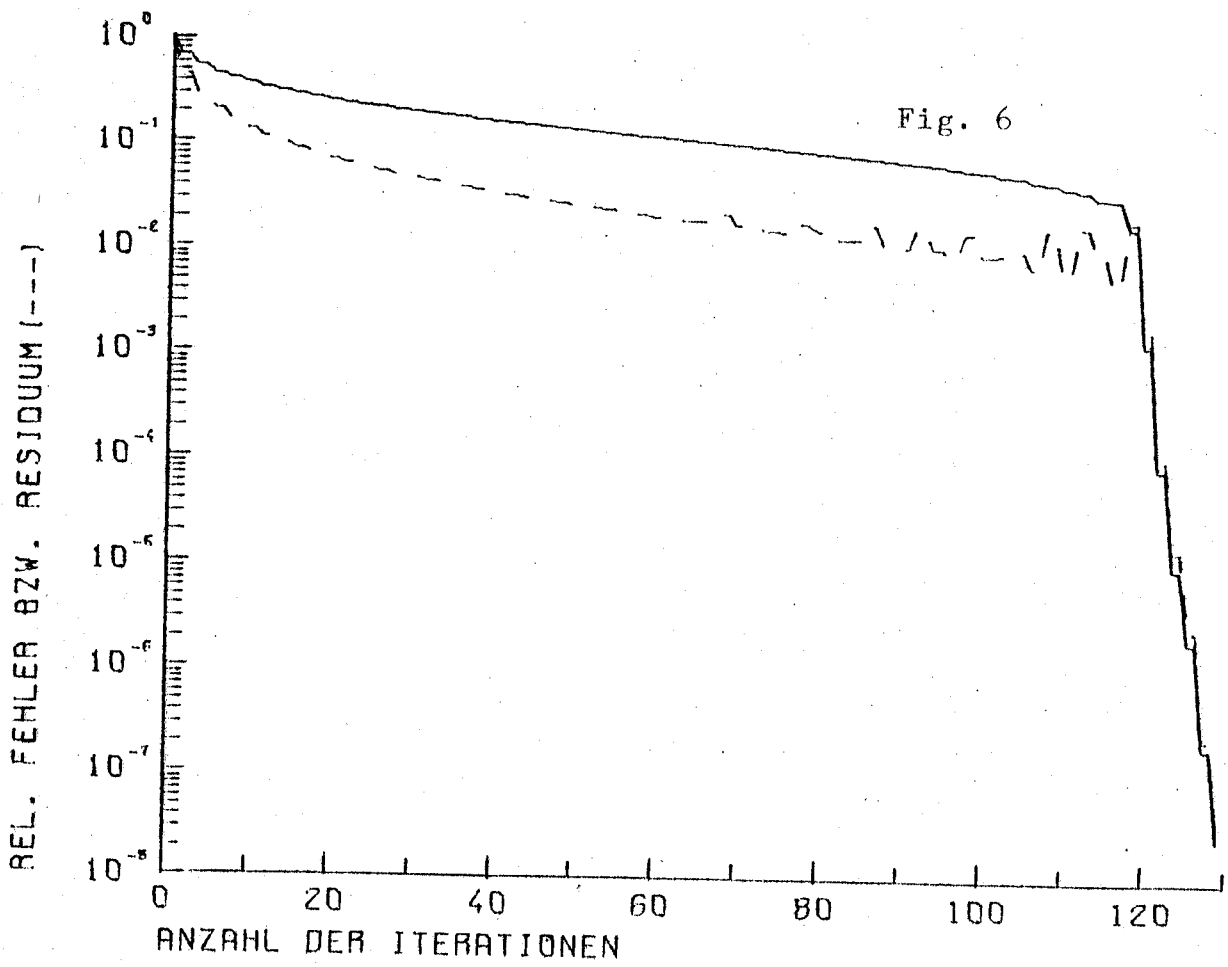
$$A_1 := \mathrm{diag}(1,2,3,\ldots,50,-50,\ldots,-3,-2,-1),$$

und die rechte Seite $b_{1/2}$ bzw. b_1 wird so gewählt, daß die Lösung jeweils durch $\bar{x}=(1,1,\ldots,1)^T$ gegeben ist. Ferner benutzen wir den Startwert $x_o=0$; dann gilt $e_o=\bar{x}-x_o=\bar{x}$, und es liegt gerade der Entartungsfall im Sinne von (6.4) vor.
STOD- und MCR-Algorithmus (7.7) und (5.4) schneiden hier gleich gut ab und benötigen

	$A = A_{1/2}$	$A = A_1$
STOD	69 (65)	129 (127)
MCR	70 (66)	130 (126)

Iterationen zur Reduktion der Fehler auf $R_k \leq 10^{-7}$ ($E_k \leq 10^{-6}$). Fig. 6 zeigt den Verlauf von E_k, R_k für das STOD-Verfahren im Fall $A=A_1$.

Fig. 6



21. Unsymmetrisches Modellproblem

In der folgenden Tabelle sind für das Verfahren(14.7) von Concus, Golub, Widlund, für GSTOD-, SPC-Craig-Algorithmus (14.3) bzw. (14.5), sowie für das GMCR-Verfahren(15.8) und seine Version(15.9) die Werte von k_o (= erster Index mit $R_{k_o} \leq 10^{-7}$) und - jeweils darunter stehend - R_{k_o} zusammengestellt:

n	σ	(14.7)	(14.3)	(14.5)	(15.8)	(15.9)
49	1	6 $7.9_{10}{}^{-9}$	5 $7.9_{10}{}^{-9}$	3 $7.9_{10}{}^{-9}$	6 $7.9_{10}{}^{-9}$	6 $7.9_{10}{}^{-9}$
"	10	14 $4.8_{10}{}^{-8}$	13 $4.8_{10}{}^{-8}$	7 $4.8_{10}{}^{-8}$	14 $4.7_{10}{}^{-8}$	14 $4.7_{10}{}^{-8}$
"	100	51 $8.6_{10}{}^{-8}$	53 $6.5_{10}{}^{-8}$	26 $8.4_{10}{}^{-8}$	51 $9.1_{10}{}^{-8}$	54 $9.3_{10}{}^{-8}$
225	1	5 $6.1_{10}{}^{-8}$	5 $2.8_{10}{}^{-9}$	3 $2.9_{10}{}^{-9}$	5 $6.1_{10}{}^{-8}$	5 $6.1_{10}{}^{-8}$
"	10	15 $2.7_{10}{}^{-8}$	15 $7.6_{10}{}^{-9}$	8 $7.6_{10}{}^{-9}$	15 $2.8_{10}{}^{-8}$	15 $2.8_{10}{}^{-8}$
"	100	83 $5.1_{10}{}^{-8}$	83 $5.3_{10}{}^{-8}$	42 $4.8_{10}{}^{-8}$	82 $9.1_{10}{}^{-8}$	84 $8.9_{10}{}^{-8}$
961	1	5 $1.2_{10}{}^{-8}$	5 $7.5_{10}{}^{-10}$	3 $7.8_{10}{}^{-10}$	5 $1.2_{10}{}^{-8}$	5 $1.2_{10}{}^{-8}$
"	10	13 $5.9_{10}{}^{-8}$	13 $1.7_{10}{}^{-8}$	7 $1.7_{10}{}^{-8}$	13 $6.0_{10}{}^{-8}$	13 $6.0_{10}{}^{-8}$
"	100	87 $9.7_{10}{}^{-8}$	87 $7.4_{10}{}^{-8}$	44 $7.4_{10}{}^{-8}$	87 $8.7_{10}{}^{-8}$	89 $8.7_{10}{}^{-8}$

Diese numerischen Resultate spiegeln sowohl die theoretische Äquivalenz der Algorithmen (14.3) und (14.5) bzw. (15.8) und

(15.9), als auch den Zusammenhang zwischen den beiden Verfahren (14.7) und (14.3) recht gut wieder. Es fällt weiter auf, daß das GMCR-Verfahren, welches ja in jedem Schritt eine neue Näherung liefert, nicht besser abschneidet als der GSTOD-Algorithmus, der in jeder zweiten Iteration auf der Stelle tritt. Berücksichtigt man noch Operationszahlen und Speicherplatzbedarf der fünf Verfahren (siehe Abschnitt 15), so sind also die Algorithmen (14.3),(14.5),(14.7) (und unter diesen wäre wohl der SPC-Craig-Algorithmus zu favorisieren) dem GMCR-Verfahren vorzuziehen.

22. Fehlerkomponenten längs einzelner Eigenvektoren beim CG-Verfahren

Anhand einiger numerischer Beispiele wollen wir aufzeigen, wie sich die in Abschnitt 11 für den CG-Algorithmus angegebenen Schranken im Vergleich zu den tatsächlichen Fehlerkomponenten darstellen. Als Koeffizientenmatrix unseres Testsystems $Ax=b$ dient jeweils die $n\times n$-Diagonalmatrix

$$A := \mathrm{diag}(\lambda,\lambda_2,\lambda_3,\ldots,\lambda_n),$$

$$\text{wobei } \lambda > 0 \text{ und } \lambda_j := (j+8)/10 \text{ für } j=2,3,\ldots,n,$$

als Startwert wird $x_o=0$ benutzt und die rechte Seite b nach Wahl eines Lösungsvektors

$$\bar{x} = (\rho,\rho_2,\rho_3,\ldots,\rho_n)^T \quad (= e_o)$$

durch $b:=A\bar{x}$ definiert. Wir interessieren uns für die Fehlerkomponente z^Te_k längs des zu λ gehörenden Eigenvektors $z=(1,0,\ldots,0)^T$ und zwar für den Fall, daß λ isoliert vom restlichen Spektrum liegt, d.h. $0<\lambda \ll \lambda_2$ oder $\lambda \gg \lambda_n$. Man beachte, daß dann für

$$\tilde{\varepsilon}_k = \sin\sphericalangle(z,\tilde{S}_k) = \left(\frac{\tan^2\sphericalangle(z,\tilde{S}_k)}{1+\tan^2\sphericalangle(z,\tilde{S}_k)}\right)^{1/2},$$

$$\varepsilon_k = \sin\sphericalangle(z,S_k) = \left(\frac{\tan^2\sphericalangle(z,S_k)}{1+\tan^2\sphericalangle(z,S_k)}\right)^{1/2}$$

nach Lemma(11.5)a) die Abschätzungen

$$(22.1)\quad \tilde{\varepsilon}_k \le \left(1 + T_{k-1}^2\left(\frac{\lambda_n+\lambda_2-2\lambda}{\lambda_n-\lambda_2}\right) / \tan^2\sphericalangle(z,A^{1/2}r_o)\right)^{-1/2} =: \tilde{\omega}_k,$$

$$(22.2)\quad \varepsilon_k \le \left(1 + T_{k-1}^2\left(\frac{\lambda_n+\lambda_2-2\lambda}{\lambda_n-\lambda_2}\right) / \tan^2\sphericalangle(z,r_o)\right)^{-1/2} =: \omega_k$$

gelten, und wir dürfen in den oberen Schranken für $|z^Te_k|$ $\tilde{\varepsilon}_k$ bzw. ε_k durch $\tilde{\omega}_k$ bzw. ω_k ersetzen.

Es wurde jeweils der Verlauf der folgenden fünf Größen geplottet: Kurve 1 zeigt die tatsächliche Fehlerkomponente

$$\sqrt{\lambda}\ |v^T e_k| / \|e_o\|_A$$

des CG-Algorithmus (der - um Rundungsfehlereinflüsse zurückzudrängen - in doppelt genauer Arithmetik implementiert wurde) und Kurve 2 die zugehörige Schranke

$$\tilde{\omega}_k \|e_k\|_A / \|e_o\|_A , \tag{22.3}$$

wie sie Korollar(11.3)a) zusammen mit (22.1) liefert. Kombiniert man die üblichen Abschätzungen für $\|e_k\|_A/\|e_o\|_A$ (siehe [1]) mit (11.3)b), so führt (22.3) zu der "a priori-Schranke" (Kurve 3)

$$\tilde{\omega}_k \cdot \min\left\{ \frac{1}{T_k(\frac{\kappa+1}{\kappa-1})},\ \frac{\max\{|1-\frac{\lambda_2}{\lambda}|,|1-\frac{\lambda_n}{\lambda}|\}}{T_{k-1}(\frac{\lambda_n+\lambda_2}{\lambda_n-\lambda_2})},\ \frac{|\rho|\sqrt{\lambda}\,\tilde{\omega}_k + \tilde{\tau}}{\|e_o\|_A} \right\},$$

wobei $\kappa := \kappa(A)$ und $\tilde{\tau} := (\sum_{j=2}^{n} \rho_j^2\, \lambda_j)^{1/2}$. Schließlich wurde die Schranke (11.11) (mit ω_k aus (22.2)) gezeichnet:

$$2\,\frac{\sqrt{\lambda}}{\|e_o\|_A}\,\omega_k(|\rho|\omega_k+\tau)(1+\sqrt{\chi_k})(1+\frac{\|A\|}{\lambda-2\,\omega_k^2\|A\|(1+2\,\chi_k)}) ,$$

dabei benutzten wir einmal den illusorischen Wert $\chi_k=0$ (Kurve 4) und einmal die realistische Größe $\chi_k=\lambda_n/\lambda_2$ falls $\lambda<\lambda_2$ bzw. $\chi_k=\kappa(A)$ falls $\lambda>\lambda_n$ (Kurve 5).

Dem Beispiel, das in Fig. 7 dargestellt ist, liegt die Wahl $n=592$, $\lambda=500$ ($\gg 60=\lambda_{592}$) zugrunde; Fig. 8 gehört zu $n=192$, $\lambda=10^{-3}$ ($\ll 1=\lambda_2$). In beiden Fällen dienten als Komponenten $\rho,\rho_2,\rho_3,\ldots,\rho_n$ von e_o Zufallszahlen aus $[-1,1]$. Ebenso wurden die Zahlen $\rho_2,\ldots,\rho_n$ für die Beispiele (beide mit $n=62$, $\lambda=10^{-4}$) in Fig. 9 und 10 erzeugt, dagegen die Komponenten $\rho=z^T e_o$ ausgezeichnet gewählt, nämlich $\rho=10^4$ (Fig. 9) und $\rho=10^{-2}$ (Fig. 10).

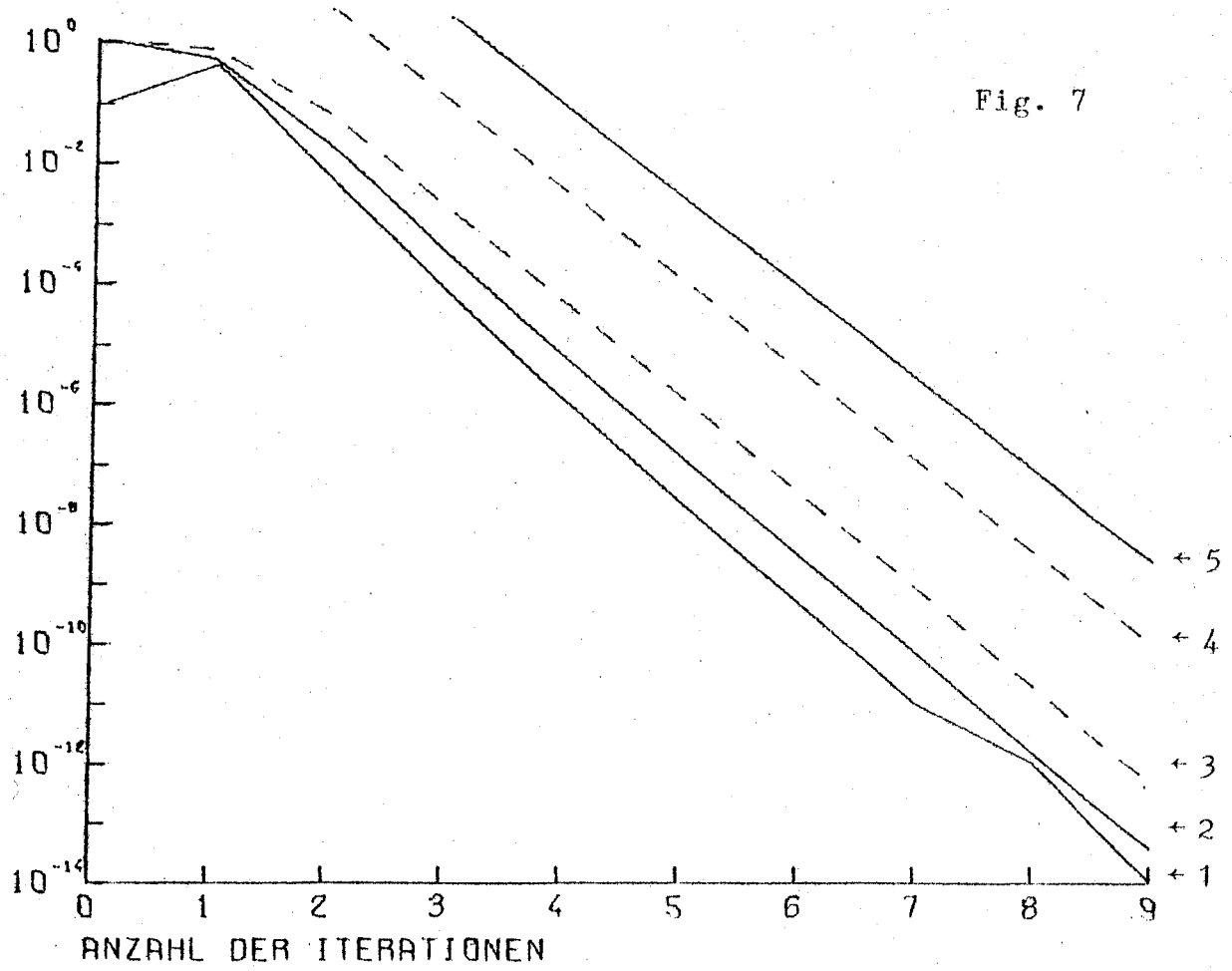

10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
10^-14
0 1 2 3 4 5 6 7 8 9
ANZAHL DER ITERATIONEN
← 5
← 4
← 3
← 2
← 1

Fig. 7

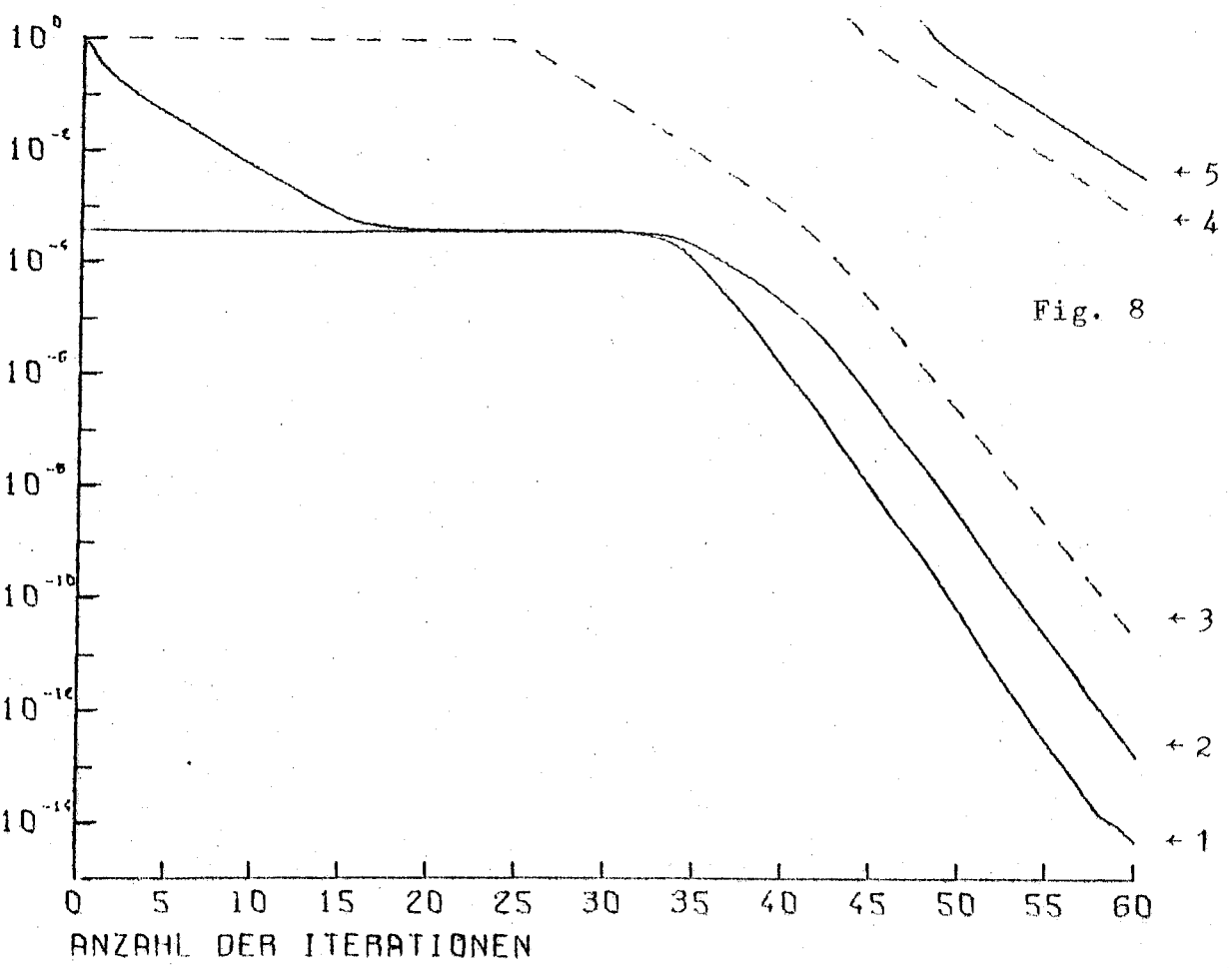

10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
10^-14
0 5 10 15 20 25 30 35 40 45 50 55 60
ANZAHL DER ITERATIONEN
← 5
← 4
← 3
← 2
← 1

Fig. 8

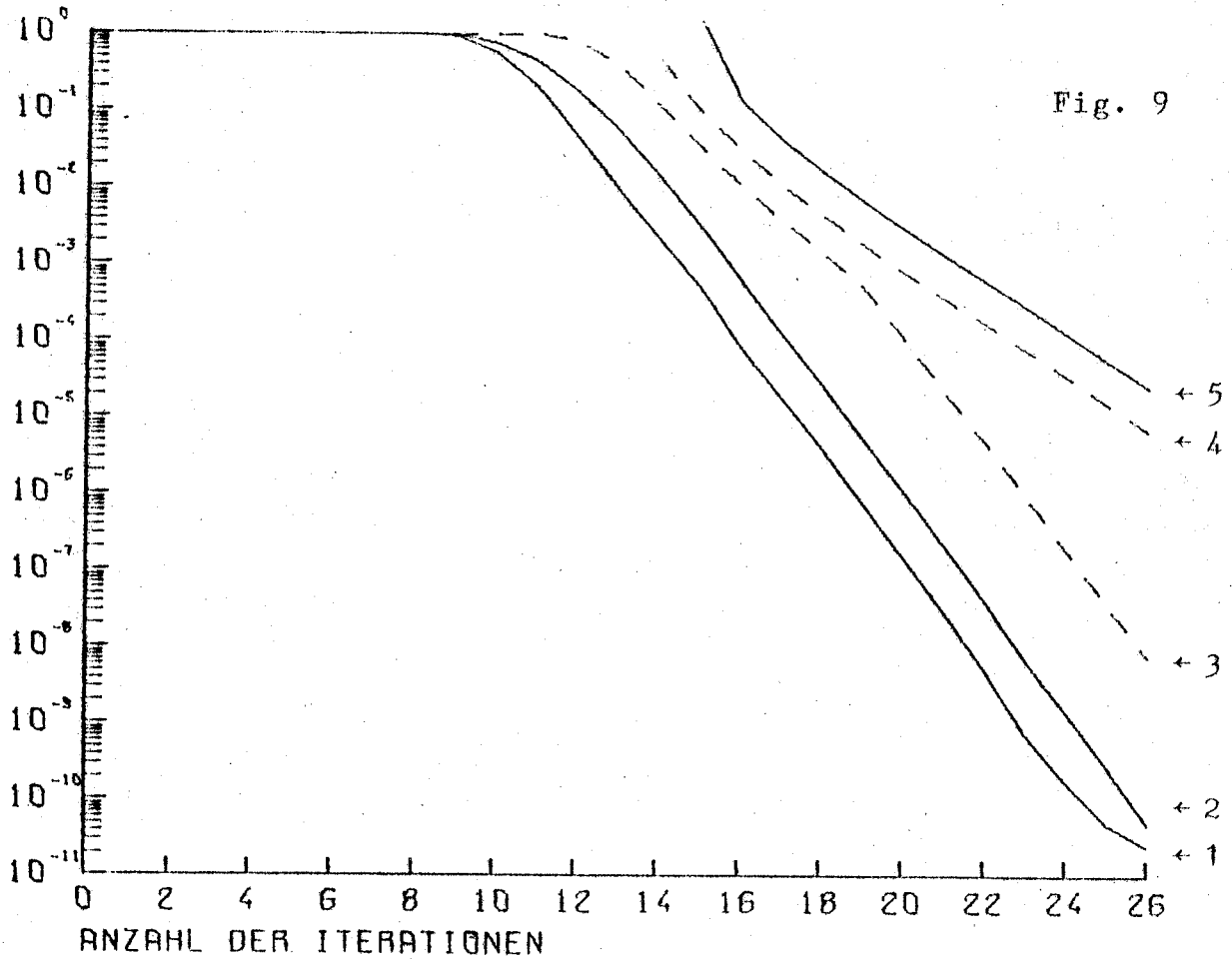

10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-9
10^-10
10^-11
0 2 4 6 8 10 12 14 16 18 20 22 24 26
ANZAHL DER ITERATIONEN
← 5
← 4
← 3
← 2
← 1

Fig. 9

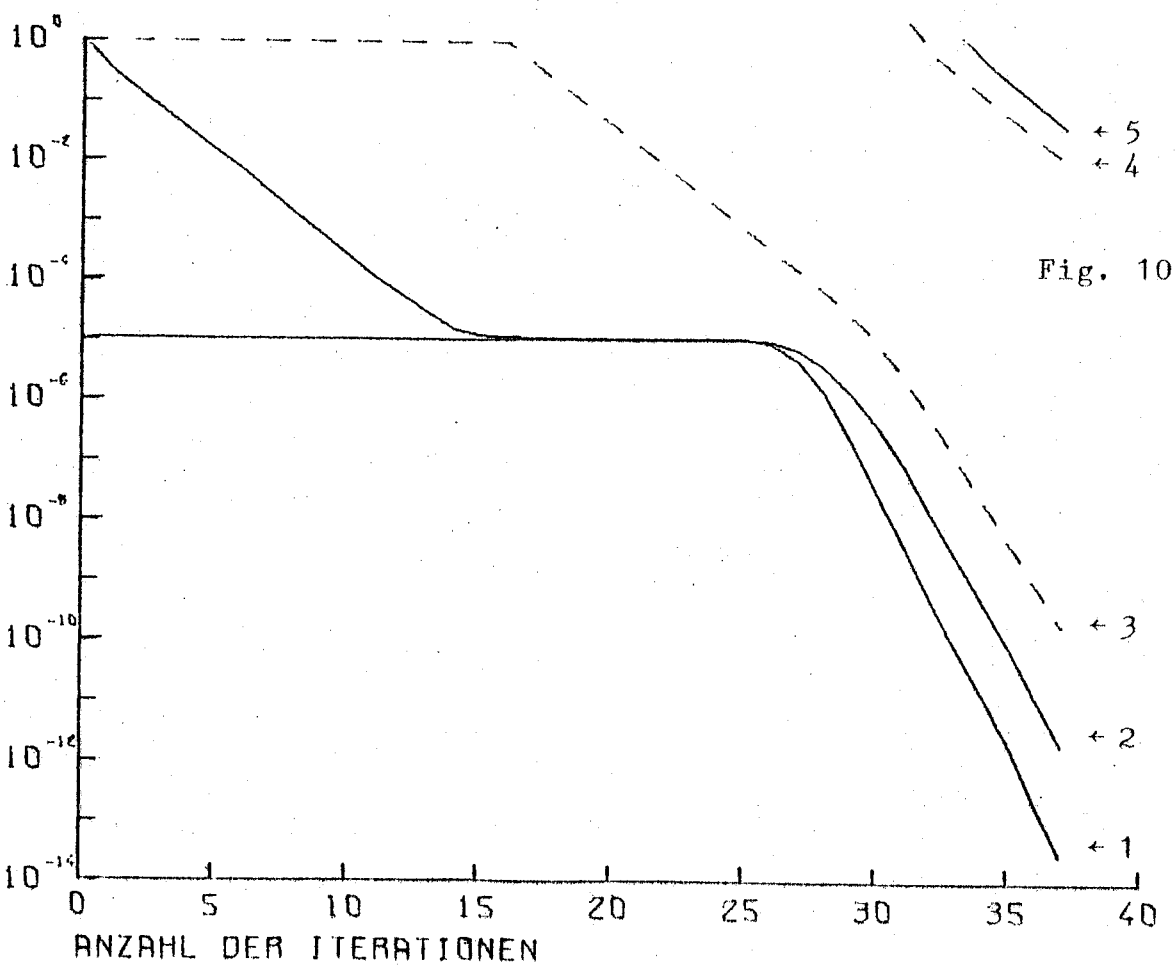

10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
10^-14
0 5 10 15 20 25 30 35 40
ANZAHL DER ITERATIONEN
← 5
← 4
← 3
← 2
← 1

Fig. 10

Literatur

[1] Axelsson,O.: Solution of linear systems of equations: iterative methods. In: Sparse Matrix Techniques (V.A.Barker, ed.). Lecture Notes in Mathematics 572. Berlin- Heidelberg-New York: Springer 1977.

[2] Axelsson,O.: Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations. Lin. Alg. Appl. 29 (1980) 1-16.

[3] Berman,A., Plemmons,R.J.: Nonnegative matrices in the mathematical sciences. Computer Science and Applied Mathematics. New York-San Francisco-London: Academic Press 1979.

[4] Bunch,J.R., Kaufman,L., Parlett,B.N.: Decomposition of a symmetric matrix. Numer. Math. 27 (1976) 95-109.

[5] Bunch,J.R., Parlett,B.N.: Direct methods for solving symmetric indefinite systems of linear equations. SIAM J. Numer. Anal. 8 (1971) 639-655.

[6] Chandra,R.: Conjugate gradient methods for partial differential equations. Ph. D. Thesis, Research report # 129, Yale University, 1978.

[7] Concus,P., Golub,G.H.: A generalized conjugate gradient method for nonsymmetric systems of linear equations. In: Computing Methods in Applied Sciences and Engineering (R.Glowinski and J.L.Lions, eds.). Lecture Notes in Economics and Mathematical Systems 134. Berlin- Heidelberg-New York: Springer 1976.

[8] Craig,E.J.: The N-step iteration procedures. J. Math. Phys. 34 (1955) 64-73.

[9] Dupont,T., Kendall,R.P., Rachford,H.H.: An approximate factorization procedure for solving self-adjoint elliptic difference equations. SIAM J. Numer. Anal. 5 (1968) 559-573.

[10] Elfving,T.: On the conjugate gradient method for solving linear least squares problems. Report LiTH-MAT-R-1978-3, Universität Linköping, Schweden, 1978.

[11] Engeli,M., Ginsburg,T., Rutishauser,H., Stiefel,E.: Refined iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems. Mitteilungen aus dem Institut für angewandte Mathematik an der ETH Zürich 8 (1959), Basel: Birkhäuser.

[12] Faddejew,D.K., Faddejewa,W.N.: Numerische Methoden der linearen Algebra. München-Wien: R.Oldenbourg 1964.

[13] Fletcher,R.: Conjugate gradient methods for indefinite systems. In: Numerical Analysis Dundee 1975 (G.A.Watson, ed.). Lecture Notes in Mathematics 506. Berlin-Heidelberg-New York: Springer 1976.

[14] Fridman,V.M.: The method of minimum iterations with minimum errors for a system of linear algebraic equations with a symmetrical matrix. USSR Computational Math. and Math. Phys. 2 (1963) 362-363.

[15] Hageman,L.A., Luk,F.T., Young,D.M.: On the equivalence of certain iterative acceleration methods. SIAM J. Numer. Anal. 17 (1980) 852-873.

[16] Hestenes,M.R., Stiefel,E.: Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. of Standards 49 (1952) 409-436.

[17] Kershaw,D.S.: The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations. J. Comp. Physics 26 (1978) 43-65.

[18] Lanczos,C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J. Res. Nat. Bur. of Standards 45 (1950) 255-282.

[19] Lebedev,V.I.: Iterative methods for solving operator equations

with a spectrum contained in several intervals. USSR Computational Math. and Math. Phys. 9 (1969) 17-24.

[20] Manteuffel,T.A.: The Tchebychev iteration for nonsymmetric linear systems. Numer. Math. 28 (1977) 307-327.

[21] Manteuffel,T.A.: An incomplete factorization technique for positive definite linear systems. Math. of Comp. 34 (1980) 473-497.

[22] Meijerink,J.A., van der Vorst,H.A.: An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Math. of Comp. 31 (1977) 148-162.

[23] Meinardus,G.: Approximation von Funktionen und ihre numerische Behandlung. Berlin-Göttingen-Heidelberg-New York: Springer 1964.

[24] Paige,C.C.: Computational variants of the Lanczos method for the eigenproblem. J. Inst. Math. Appl. 10 (1972) 373-381.

[25] Paige,C.C.: Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. J. Inst. Math. Appl. 18 (1976) 341-349.

[26] Paige,C.C., Saunders,M.A.: Solution of sparse indefinite systems of linear equations. SIAM J. Numer. Anal. 12 (1975) 617-629.

[27] Parlett,B.N.: The symmetric eigenvalue problem. Series in Computational Mathematics. Englewood Cliffs: Prentice Hall 1980.

[28] Reid,J.K.: On the method of conjugate gradients for the solution of large sparse systems of linear equations. In: Large Sparse Sets of Linear Equations (J.K.Reid, ed.). London-New York: Academic Press 1971.

[29] Saad,Y.: On the rates of convergence of the Lanczos and the block-Lanczos methods. SIAM J. Numer. Anal. 17 (1980) 687-706.

[30] Saad,Y.: Krylov subspace methods for solving large unsymmetric linear systems. Math. of Comp. 37 (1981) 105-126.

[31] Stewart,G.W.: The convergence of the method of conjugate gradients at isolated extreme points of the spectrum. Numer. Math. 24 (1975) 85-93.

[32] Stoer,J.: Konjugierte-Gradienten-Verfahren in der Numerischen Mathematik. Preprint Nr. 59 der Mathematischen Institute der Julius-Maximilians-Universität Würzburg, 1979.

[33] Stoer,J., Bulirsch,R.: Einführung in die Numerische Mathematik II. Berlin-Heidelberg-New York: Springer 1978.

[34] Varga,R.S.: Matrix iterative analysis. Series in Automatic Computation. Englewood Cliffs: Prentice Hall 1962.

[35] Varga,R.S., Saff,E.B., Mehrmann,V.: Incomplete factorizations of matrices and connections with H-matrices. SIAM J. Numer. Anal. 17 (1980) 787-793.

[36] Widlund,O.: A Lanczos method for a class of nonsymmetric systems of linear equations. SIAM J. Numer. Anal. 15 (1978) 801-812.

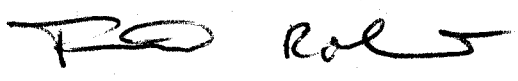
[37] Young,D.M.: Iterative solution of large linear systems. Computer Science and Applied Mathematics. New York-San Francisco-London: Academic Press 1971.

[38] Young,D.M., Jea,K.C.: Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods. In: Large Scale Matrix Problems (A.Björck, R.J.Plemmons and H.Schneider, eds.). New York-Oxford: North Holland 1981.

Lebenslauf

1. August	1955	geboren in Schweinfurt als zweites Kind des Bauschlossers Wilhelm Freund und seiner Ehefrau Frieda, geb. Meder
seit	1979	verheiratet mit Susanne Freund, geb. Rühl
September	1961	Evangelische Volksschule Gochsheim
September	1966	Alexander-von-Humboldt-Gymnasium Schweinfurt
Juni	1975	Abitur
November	1975	Studium der Mathematik und Physik an der Universität Würzburg
Frühjahr	1978	Vorexamen in Mathematik und Physik
Oktober	1979	Vordiplom in Mathematik
Herbst	1980	Staatsexamen in Mathematik und Physik
April	1982	Hauptdiplom in Mathematik
seit Juli	1982	wissenschaftlicher Mitarbeiter am Institut für Angewandte Mathematik und Statistik der Universität Würzburg

Würzburg, im Mai 1983



Ich erkläre, daß ich die vorliegende Arbeit selbständig angefertigt und dabei keine anderen als die angegebenen Hilfsmittel verwendet habe.