Institut für Angewandte Analysis
und Stochastik

im Forschungsverbund Berlin e.V.

Iterative Verfahren für lineare Gleichungssysteme mit schwach besetzten Koeffizientenmatrizen

R. Schlundt

submitted: 17th December 1992

Institut für Angewandte Analysis
und Stochastik
Hausvogteiplatz 5-7
D – O 1086 Berlin
Germany

Preprint No. 31
Berlin 1992

1991 Mathematics Subject Classification. 65 F 10.
Key words and phrases. große linare Systeme, iterative Verfahren, Krylov–Unterraum–Methoden, GMRES–Algorithmus, QMR–Algorithmus.

Herausgegeben vom
Institut für Angewandte Analysis und Stochastik
Hausvogteiplatz 5-7
D – O 1086 Berlin

Fax: + 49 30 2004975
e-Mail (X.400): c=de;a=dbp;p=iaas-berlin;s=preprint
e-Mail (Internet): preprint@iaas-berlin.dbp.de

ITERATIVE VERFAHREN FÜR LINEARE GLEICHUNGSSYSTEME MIT SCHWACH BESETZTEN KOEFFIZIENTENMATRIZEN

Rainer Schlundt

Abstract. Für die Lösung großer linearer Gleichungssysteme mit schwach besetzten Koeffizientenmatrizen werden das GMRES-Verfahren und die QMR-Methode vorgestellt. Beide iterativen Verfahren basieren auf *Krylov*-Unterraum-Methoden. Es werden sowohl die *Gram-Schmidt*- als auch die *Householder*-Orthogonalisierung für GMRES betrachtet. Das QMR-Verfahren wird mit dem *look-ahead Lanczos*-Algorithmus kombiniert. Ein einfacher Vergleich zwischen GMRES und QMR wird angegeben.

Key words. große lineare Systeme, iterative Verfahren, Krylov-Unterraum-Methoden, GMRES-Algorithmus, QMR-Algorithmus

AMS subject classifications. 65F10

Inhaltsverzeichnis

1 Einführung

1.1 Ein Überblick über Krylov-Unterräume

Es werden iterative Methoden für die Lösung von linearen Gleichungssystemen

$$A * x = b \tag{1}$$

betrachtet, wobei $A \in \mathrm{R}^{n \times n}$ nicht symmetrisch und schwach besetzt ist. Von besonderem Interesse sind hierbei die *Krylov*-Unterraum-Methoden. Ausgehend von einem Startwert x_0 für das lineare Gleichungssystem (1) wird mit der Projektionsmethode eine approximative Lösung x_m aus dem affinen Unterraum $x_0 + K_m$ der Dimension m gesucht, wobei die *Galerkin*-Bedingung

$$b - A * x_m \perp L_m \tag{2}$$

erfüllt sein muß.
L_m ist hierbei ein anderer Unterraum der Dimension m. Die *Krylov*-Unterraum-Methode ist eine Methode, in der der Unterraum K_m der *Krylov*-Unterraum

$$K_m(A, r_0) = span\{r_0, Ar_0, A^2r_0, \ldots, A^{m-1}r_0\} \tag{3}$$

ist, wobei

$$r_0 = b - A * x_0 \tag{4}$$

das Anfangsresiduum darstellt. Die verschiedenen Versionen von *Krylov*-Unterraum- Methoden hängen von der Wahl der Unterräume K_m und L_m und von der Art und Weise der Vorkonditionierung ab. Die häufigste Wahl von K_m und L_m soll die folgende Aufstellung verdeutlichen:

1. $L_m = K_m = K_m(A, r_0)$
   Dies ist die Methode der orthogonalen Projektion. Ein Vertreter ist die Methode von *Arnoldi* [15]. Falls A symmetrisch und positiv definit ist, erhält man das CG-Verfahren (conjugate gradient). Das CG-Verfahren und die Methode von *Arnoldi* sind dann äquivalent. Ist A symmetrisch und indefinit, ist die Methode von *Arnoldi* dem SYMMLQ-Algorithmus [14] äquivalent.

2. $L_m = AK_m; K_m = K_m(A, r_0)$
   Mit dieser Wahl von L_m minimiert die approximative Lösung x_m die Residualnorm $\|b - Ax\|_2$ über alle Vektoren $x_m \in x_0 + K_m$. Der bekannteste Vertreter ist das GMRES-Verfahren (generalized minimum residual) [16]. Falls A symmetrisch und positiv definit ist, sind das CR-Verfahren (conjugate residual) und GMRES äquivalent. Ist A symmetrisch und indefinit, sind GMRES und der MINRES-Algorithmus [14] äquivalent.

3. $L_m = K_m(A^T, r_0); K_m = K_m(A, r_0)$
   Für den symmetrischen Fall reduzieren sich die Methoden dieser Klasse auf die Methoden der ersten Klasse. Für den nichtsymmetrischen Fall ist das BCG-Verfahren (biconjugate gradient) [11], [6] ein guter Vertreter dieser Klasse. Eine effektive Variante dieser Methode ist das CGS-Verfahren (conjugate gradient squared) [17]. Ein anderer interessanter Vertreter dieser Klasse ist das QMR-Verfahren (quasi-minimal residual) mit dem *look-ahead Lanczos* -Algorithmus [8].

4. $L_m = K_m = K_m(A^TA, A^Tr_0)$
   Das CG-Verfahren auf die Normalgleichung

$$A^TA * x = A^T * b \tag{5}$$

angewandt, liefert das CGNR-Verfahren [3]. Die Buchstaben „NR“ stehen hier für die Minimierung der Residualnorm (residual norm) über K_m. Die Konvergenz des CG-Verfahrens hängt vom Spektrum der Koeffizientenmatrix A ab. Für das CGNR-Verfahren hängt damit die Kovergenzumgebung von

$$\lambda(A^T A) = \{\sigma^2 | \sigma \in \sigma(A)\} \tag{6}$$

ab, d.h. vom Quadrat der Singulärwerte von A, wobei $\lambda(A^T A)$ die Menge aller Eigenwerte von $A^T A$ darstellt. Im einzelnen wird die Konvergenzumgebung des CGNR-Verfahrens von der Konditionszahl

$$\kappa_2(A^T A) = (\kappa_2(A))^2 \tag{7}$$

beherrscht. Da für viele Prozesse die Konditionszahl (7) zu groß ist , erweist sich der Weg über die Normalgleichungen (5) als nicht sinnvoll. In dieser Klasse sind auch Methoden vorhanden, die mit dem CG-Verfahren die Gleichung

$$A^T A * y = b \qquad \text{mit} \qquad x = A^T * y \tag{8}$$

behandeln. Sie werden CGNE-Verfahren genannt. Die Buchstaben „NE“ stehen für die Minimierung der Fehlernorm (error norm) über K_m. Wenn man die *Galerkin*-Bedingung (2) für die Variable y betrachtet, so ist klar, daß $K_m = K_m(AA^T, r_0)$ und $L_m = K_m$ gilt. Verwendet man die Beziehung $x = A^T * y$ zwischen den Variablen x und y, so gilt $K_m = K_m(A^T A, A^T r_0)$ und $L_m = A^{-T} K_m$.

Für Spezialfälle ist die Lösung der Normalgleichungen (5) bzw. (8) sehr effektiv. So liefern z.B. die beiden Verfahren CGNR und CGNE für unitäre Matrizen A die exakte Lösung schon nach einem Schritt, während die *Krylov*-Unterraum-Methoden sehr langsam konvergieren [13].

1.2 Vorkonditionierung

Ein wichtiger Faktor für die erfolgreiche Anwendung CG-ähnlicher Methoden ist die Präkonditionierungstechnik. Ein typisches Beispiel ist die Ersetzung des Originalsystems (1) durch das äquivalente System

$$W^{-1} A * x = W^{-1} * b. \tag{9}$$

Im klassischen Fall ist die Matrix W ein unvollständiger LU-Vorkonditionierer (ILU - incomplete LU preconditioning) der Form $W = LU$, wobei L eine untere Dreiecksmatrix und U eine obere Dreiecksmatrix ist. Die Dreiecksmatrizen L und U haben dieselbe Struktur wie der untere bzw. obere Dreiecksteil der Matrix A. Ist die Matrix A schwach besetzt,

erhält man die unvollständige Faktorisierung aus der Standard-LU-Faktorisierung von A, wobei die während des Generierungsprozesses anfallenden fill-in-Elemente vernachlässigt werden. Es existieren viele andere Wege, um die unvollständige Faktorisierung einer gegebenen Matrix zu definieren. In den meisten Fällen wird irgendeine Form der Diagonaldominanz ausgenutzt.

1.3 Das Matrix-Vektor-Produkt

Die Matrix-Vektor-Multiplikation $y = A * x$ ist auf den meisten Computern relativ leicht und effizient zu implementieren. Einige Speicherungstechniken sollen hier vorgestellt werden.

1. Zeilenweise Speicherung
   Die Datenstruktur besteht aus den drei eindimensionalen Feldern AR, JA und IA. In einem *REAL*-Feld AR(1:NNZ) werden die Nichtnullelemente von A zeilenweise abgespeichert, d.h. die Elemente einer gegebenen Zeile sind kontinuierlich gespeichert. In einem *INTEGER*-Feld JA(1:NNZ) sind die Spaltennummern der Nichtnullelemente von A gespeichert. Schließlich wird noch ein *INTEGER*-Feld IA(1:N+1) als Pointerfeld benötigt. In IA sind die Startpositionen der Nichtnullelemente für jede Zeile von A enthalten, wobei IA(N+1)=NNZ+1. Dieses Speicherschema erlaubt es, die Komponenten des Vektors y als unabhängige Skalarprodukte zu berechnen.

```
      DO 110 I = 1,N
         K1 = IA(I)
         K2 = IA(I+1)
         Y(I) = DOTPRODUCT(AR(K1:K2) , X(JA(K1:K2)))
110   CONTINUE
```

   Ausgehend vom Standpunkt der Implementation kann die äußere Schleife parallel abgearbeitet werden. Die indirekte Adressierung im zweiten Vektor des Skalarprodukts muß durch spezielle Hardware-Instruktionen behandelt werden.

2. Spaltenweise Speicherung mit Diagonalbehandlung
   Die Datenstruktur besteht aus zwei zweidimensionalen Feldern AC und KA. Die Nichtnullelemente von A werden in dem *REAL*-Feld AC(1:N,1:NZ) abgespeichert, wobei NZ die maximale Anzahl der Nichtnullelemente in einer Zeile von A angibt. In der ersten Spalte von AC steht die Hauptdiagonale von A. Die Nichtdiagonalelemente von A werden zeilenweise und lückenlos in AC gespeichert. In dem *INTEGER*-Feld KA(1:N,1:NZ) werden die Spaltennummern der Elemente von A, die in AC gespeichert sind, angegeben. Ein Nachteil dieser Speicherung ist das Vorhandensein von Nullelementen in AC. Die entsprechenden Indizes von KA werden mit $x, 1 \leq x \leq n$, belegt. Eine sehr effektive Implementation für Vektorrechner ist die folgende Variante:

```
      DO 120 I = 1,N
         T = X(I)*AC(I,1)
         DO 110 J = 2,NZ
            T = T+U(KA(I,J))*AC(I,J)
110      CONTINUE
         Y(I) = T
120   CONTINUE
```

3. Zeilenweise Speicherung mit Vorsortierung und Diagonalbehandlung
   Diese Speicherungstechnik stellt eine Modifikation der zweiten dar. Ausgangspunkt ist die spaltenweise Speicherung. In dem Feld AC werden die Zeilen aufsteigend nach der Anzahl der Nichtnullelemente pro Zeile sortiert, und man erhält das Feld AC'. AC' hat eine Blockstruktur. Analog wird das Feld KA behandelt. Das Ergebnis ist in dem Feld KA' abgespeichert. Hieraus wird eine Datenstruktur aus vier eindimensionalen Feldern ACP, KAP, IAP und JAP aufgebaut. Die Nichtnullelemente in AC' sind spaltenweise pro Block in dem *REAL*-Feld ACP(1:NNZ) abgespeichert. Analog wird KA' in das *INTEGER*-Feld KAP(1:NNZ) transformiert. Das *INTEGER*- Feld IAP(1:NB+1) enhält die Startpositionen der einzelnen Blöcke, wobei NB die Anzahl der Blöcke angibt und IAP(NB+1)=NNZ+1. Schließlich sind in dem *INTEGER*-Feld JAP(1:NB) die Anzahl der Spalten für jeden Block enthalten.

2 Die Methode von Arnoldi und GMRES

In diesem Abschnitt wird ein Überblick über die Methode von *Arnoldi* und den GMRES-Algorithmus gegeben. Es werden sowohl die *Gram-Schmidt*- als auch die *Householder*-Orthogonalisierung betrachtet. Ausgangspunkt ist das lineare Gleichungssystem (1) mit eventueller Vorkonditionierung (9). Ausgehend vom Startwert x_0 wird eine approximative Lösung x_m für (1) gesucht. Mit $x = x_0 + z$ geht (1) in das äquivalente System

$$A * z = r_0 \tag{10}$$

über, wobei r_0 das Anfangsresiduum (4) darstellt. Sei K_m der *Krylov*-Unterraum (3), dann erhält man

$$x_m = x_0 + z_m \qquad \text{mit} \qquad z_m \in K_m, \tag{11}$$

so daß $(b - Ax_m) \perp K_m$ oder äquivalent $(r_0 - Az_m) \perp K_m$ für die Methode von *Arnoldi* gilt. Für das GMRES-Verfahren erhält man

$$\|b - Ax_m\|_2 = \min_{x \in x_0 + K_m} \|b - Ax\|_2 = \min_{z \in K_m} \|r_0 - Az\|_2 . \tag{12}$$

Gesucht wird eine l_2-Orthonormalbasis $V_m = \{v_1, \ldots, v_m\}$ von K_m. Für den *Arnoldi*-Prozeß wird V_m durch *Gram-Schmidt*-Orthogonalisierung erzeugt.

2.1 Die Methode von Arnoldi

Der Algorithmus zur Bestimmung der l_2-Orthonormalbasis und zur Erzeugung der approximativen Lösung lautet folgendermaßen:

1. Bestimmung des Anfangsresiduums r_0 aus x_0; $r_0 = b - Ax_0$
2. Bestimmung des ersten Basisvektors; $v_1 = r_0/\|r_0\|_2$
3. Iterationsprozeß für $j = 1, 2, \ldots$
   (a) Bildung von Av_j
   (b) Bildung von $h_{ij} = (Av_j, v_i)$ für $i = 1, \ldots, j$
   (c) Konstruktion des nächsten Basisvektors:
   $$\bar{v}_{j+1} = Av_j - \sum_{i=1}^{j} h_{ij} v_i$$
   (d) $h_{j+1j} = \|\bar{v}_{j+1}\|_2$
   (e) $v_{j+1} = \bar{v}_{j+1}/h_{j+1j}$
   (f) $p_j = \|b - Ax_j\|_2$
   (g) Entscheidung über die Akzeptanz der Näherungslösung:
   falls $p_j \leq \varepsilon$, dann $m = j$ und *goto* (4)
4. Bestimmung der Näherungslösung x_m^A unter Benutzung der $m \times m$-*Hessenberg*-Matrix $H_m = (h_{ij})$ für $1 \leq i \leq min(j+1, m)$ und $1 \leq j \leq m$:
   $x_m^A = x_0 + z_m^A$ mit $z_m^A = \|r_0\|_2 V_m H_m^{-1} e_1$ und $e_1 = [1, 0, \ldots, 0]^T \in \mathrm{R}^m$

Bei der Konstruktion des Basisvektors $\bar{v}_{j+1}$ nach (3c) wird überprüft, ob der neuentstandene Vektor $\bar{v}_{j+1}$ orthogonal zu allen vorhergehenden Vektoren $v_i, i = 1, \ldots, j$, ist. Falls dies nicht der Fall ist, wird eine Nachorthogonalisierung durchgeführt. Für jedes i, für das das Skalarprodukt $\eta_i = (\bar{v}_{j+1}, v_i)$ die Ungleichung $|\eta_i| > eps * |h_{ij}|$ erfüllt, werden die Korrekturen $\bar{v}_{j+1} := \bar{v}_{j+1} - \eta_i v_i$ und $h_{ij} := h_{ij} - \eta_i$ vorgenommen. Die Größe *eps* ist hierbei eine Genauigkeitsschranke.
Einige Bemerkungen zur Bestimmung von $p_j = \|b - Ax_j\|_2$:
Aus der Konstruktion der Basisvektoren (3c) folgt

$$AV_m = V_m H_m + \bar{v}_{m+1} e_m^T, \tag{13}$$

wobei $e_m = [0, \ldots, 0, 1]^T \in \mathrm{R}^m$. Diese Relation ist gleichbedeutend mit

$$AV_m = V_{m+1} \bar{H}_m, \tag{14}$$

wobei $\bar{H}_m$ eine $(m+1) \times m$-*Hessenberg*-Matrix mit $H_m = [I_m, 0]\bar{H}_m$ (Streichung der letzten Zeile) ist. Für das Verfahren von *Arnoldi* ist die Bedingung $(b - Ax_m) \perp K_m$ äquivalent zu

$$V_m^T (b - Ax_m^A) = 0. \tag{15}$$

Da die Lösung x_m^A in dem affinen Raum $x_0 + K_m$ liegt, gilt $x_m^A = x_0 + V_m y$ für beliebiges $y \in \mathrm{R}^m$. Das System (15) ist äquivalent zu

$$V_m^T A V_m y = V_m^T r_0. \tag{16}$$

Mit Hilfe von (13) erhält man $H_m = V_m^T A V_m$. Unter Verwendung von $r_0 = \|r_0\|_2 v_1$ geht (16) über in

$$H_m y = \|r_0\|_2 e_1. \tag{17}$$

Sei y_m^A die Lösung des linearen Gleichungssystems (17) der Ordnung m. Damit ergibt sich für die Lösung x_m^A der Ausdruck $x_m^A = x_0 + V_m y_m^A$.
Zur Bestimmung des Residuumvektors r_m betrachtet man die folgend Abschätzung

$$\begin{aligned}
r_m^A &= b - A x_m^A \\
&= b - A(x_0 + V_m y_m^A) \\
&= r_0 - A V_m y_m^A \\
&= r_0 - (V_m H_m + \bar{v}_{m+1} e_m^T) y_m^A \\
&= r_0 - V_m H_m y_m^A - \bar{v}_{m+1} e_m^T y_m^A \\
&= \|r_0\|_2 v_1 - V_m H_m y_m^A - \bar{v}_{m+1} e_m^T y_m^A \\
&= V_m(\|r_0\|_2 e_1 - H_m y_m^A) - \bar{v}_{m+1} e_m^T y_m^A \\
&= -\bar{v}_{m+1} e_m^T y_m^A.
\end{aligned}$$

Hieraus ergibt sich die Abschätzung

$$\|r_m^A\|_2 = \|b - A x_m^A\|_2 = h_{m+1m} |e_m^T y_m^A|. \tag{18}$$

Seien L_m und U_m die Faktoren der LU-Zerlegung der Matrix H_m, d.h. $H_m = L_m U_m$. Die Matrix L_m hat neben der Hauptdiagonale nur noch eine untere Nebendiagonale mit den Elementen $l_{m_{j+1j}}, j = 1, \ldots, m-1$. Sei $u_{m_{mm}}$ das letzte Element der Hauptdiagonale der Matrix U_m. Mit diesen Elementen erhält man dann eine Abschätzung für $|e_m^T y_m^A|$:

$$|e_m^T y_m^A| = \|r_0\|_2 |(\prod_{j=1}^{m-1} l_{m_{j+1j}})/u_{m_{mm}}|. \tag{19}$$

Mit (19) geht (18) über in

$$\|r_m^A\|_2 = \|b - A x_m^A\|_2 = \|r_0\|_2 h_{m+1m} |(\prod_{j=1}^{m-1} l_{m_{j+1j}})/u_{m_{mm}}|. \tag{20}$$

Aus (20) ist ersichtlich, daß die LU-Zerlegung der *Hessenberg*-Matrix H_m sowohl zur Bestimmung des Vektors y_m^A als auch zur Normabschätzung von $b - A x_m^A$ verwendet werden kann.

2.2 Der GMRES-Algorithmus

Der Iterationsprozeß zur Bestimmung der Orthonormalbasis $V_m = \{v_1, \ldots, v_m\}$ ist identisch mit dem Verfahren von *Arnoldi*. Die Bestimmung der Näherungslösung x_m^A im Punkt

4. ändert sich.
Zur Bestimmung der Näherungslösung x_m^G wird die $(m+1) \times m$-*Hessenberg*-Matrix $\bar{H}_m$ mit $\bar{H}_m = (h_{ij})$ für $1 \leq i \leq j+1$ und $1 \leq j \leq m$ benötigt. Für die Normabschätzung des Residuums gilt:

$$\begin{aligned} \|b - Ax_m\|_2 &= \|b - A(x_0 + z_m)\|_2 \\ &= \|r_0 - Az_m\|_2 \\ &= \|\|r_0\|_2 v_1 - AV_m y_m\|_2 \\ &= \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m y_m)\|_2 \\ &= \|\|r_0\|_2 e_1 - \bar{H}_m y_m\|_2. \end{aligned}$$

Sei y_m^G die Lösung des Minimumproblems

$$\begin{array}{l} min \\ y \in \mathrm{R}^m \end{array} \quad \|\|r_0\|_2 e_1 - \bar{H}_m y_m\|_2 \quad . \tag{21}$$

Mit y_m^G erhält man $z_m^G = V_m y_m^G$ und damit $x_m^G = x_0 + z_m^G$. Zur Bestimmung der Lösung des Minimumproblems (21) wird die *Hessenberg*-Matrix $\bar{H}_m$ durch QR-Faktorisierung zerlegt. Die Zerlegung wird mittels *Givens*-Rotationen durchgeführt. Sei $G_{jk} \in \mathrm{R}^{(k+1)\times(k+1)}$ eine Drehungsmatrix, dann ist $G_{jj} * G_{j-1j} * \cdots * G_{1j} * \bar{H}_j = R_j \in \mathrm{R}^{(j+1)\times j}$ eine obere Dreiecksmatrix, deren letzte Zeile nur Nullelemente enthält. Hieraus ergibt sich

$$Q_m^T \bar{H}_m = R_m \tag{22}$$

mit $Q_m^T = G_{mm} * G_{m-1m} * \cdots * G_{1m} \in \mathrm{R}^{(m+1)\times(m+1)}$ und $R_m \in \mathrm{R}^{(m+1)\times m}$.
Es gilt dann

$$\begin{aligned} \|\|r_0\|_2 e_1 - \bar{H}_m y_m\|_2 &= \|\|r_0\|_2 e_1 - Q_m R_m y_m\|_2 \\ &= \|\|r_0\|_2 Q_m^T e_1 - R_m y_m\|_2. \end{aligned}$$

Mit

$$R_m = \begin{pmatrix} \hat{R}_m \\ 0 \ldots 0 \end{pmatrix}$$

und

$$g_m = \|r_0\|_2 Q_m^T e_1 = \begin{pmatrix} \hat{g}_m \\ g \end{pmatrix},$$

wobei $\hat{g}_m \in \mathrm{R}^m$ und $g \in \mathrm{R}$, ergibt sich y_m^G zu

$$y_m^G = (\hat{R}_m)^{-1} \hat{g}_m. \tag{23}$$

Für die Abschätzung der Norm des Residuums erhält man:

$$\begin{aligned} \|r_m^G\|_2 &= \|b - Ax_m^G\|_2 \\ &= \|\|r_0\|_2 Q_m^T e_1 - R_m y_m^G\|_2 \\ &= \|\|r_0\|_2 e_{m+1}^T Q_m^T e_1\|_2 \\ &= |g|. \end{aligned}$$

Mit den Nichtdiagonalelementen $s_i, i = 1, \ldots, m$, in den Drehungsmatrizen G_{im} erhält man für das Residuum die Abschätzung

$$\|r_m^G\|_2 = |g| = \|r_0\|_2 | \prod_{i=1}^{m} s_i|. \tag{24}$$

2.3 Vergleich der Residualnormen von Arnoldi mit GMRES

Ausgangspunkt ist die QR-Zerlegung der *Hessenberg*-Matrix H_m des *Arnoldi*-Verfahrens. Zwischen den *Hessenberg*-Matrizen H_m und $\bar{H}_{m-1}$ besteht die Beziehung

$$H_m = [\bar{H}_{m-1}, h_m] \tag{25}$$

mit $h_m = (h_{1m}, \ldots, h_{mm})^T$. Mit der QR-Zerlegung von $\bar{H}_{m-1}$ in $\bar{H}_{m-1} = Q_{m-1}R_{m-1}$, wobei $Q_{m-1} \in \mathrm{R}^{m \times m}$ und $R_{m-1} \in \mathrm{R}^{m \times (m-1)}$, erhält man für (25)

$$\begin{aligned} H_m &= [Q_{m-1}R_{m-1}, h_m] \\ &= Q_{m-1}[R_{m-1}, Q_{m-1}^T h_m] \\ &= Q_{m-1}\hat{R}_m, \end{aligned}$$

wobei $\hat{R}_m = [R_{m-1}, Q_{m-1}^T h_m] \in \mathrm{R}^{m \times m}$. Aus (17) folgt $Q_{m-1}\hat{R}_m y = \|r_0\|_2 e_1$ und damit

$$\hat{R}_m y = \|r_0\|_2 Q_{m-1}^T e_1. \tag{26}$$

Betrachtet man nun die letzte Komponente in (26), so ergibt sich

$$\begin{aligned} \hat{r}_{m_{mm}} &= \|r_0\|_2 e_m^T Q_{m-1}^T e_1 \\ &= \|r_0\|_2 | \prod_{i=1}^{m-1} s_i|, \end{aligned} \tag{27}$$

wobei $\hat{r}_{m_{mm}}$ das letzte Element der Hauptdiagonale von $\hat{R}_m$ ist. Aus (18), (24) und (27) erhält man

$$\|r_m^A\|_2 = \|r_{m-1}^G\|_2 h_{m+1m} / |\hat{r}_{m_{mm}}|. \tag{28}$$

Unter Zuhilfenahme der Definition der *Givens*-Rotation (siehe [10])

$$|s_m| = h_{m+1m} \sqrt{\hat{r}_{m_{mm}}^2 + h_{m+1m}^2}$$

erhält man nach [2] das folgende Resultat.

SATZ 1:

Nach m erfolgreichen Iterationsschritten im Arnoldi-Prozeß und bei nichtsingulärer Matrix H_m gilt

$$\|r_m^A\|_2 = \|r_m^G\|_2 * \sqrt{1 + (h_{m+1m}/\hat{r}_{m_{mm}})^2} \tag{29}$$

für die Residualnormabschätzung beider Verfahren.

Durch algebraische Umformungen erhält man unter Verwendung von (28) und (29) die Abschätzung

$$\|r_m^A\|_2^2 = \|r_m^G\|_2^2 (1 - s_m^2)^{-1}. \tag{30}$$

Dieses Resultat zeigt, daß für $s_m \longrightarrow 1$ die Residualnorm des *Arnoldi*-Verfahrens unbeschränkt wächst. Für den GMRES-Algorithmus kann man nach (24) eine „Stagnation" ableiten.

2.4 GMRES mit Householder-Orthogonalisierung

Die Bildung von orthogonalen Vektoren auf der Basis des *Gram-Schmidt* -Prozesses kann zu fehlerhaften Ergebnissen führen, wenn die zu orthogonalisierenden Vektoren nicht genügend unabhängig sind. Ein Ergebnis von [1] soll dies veranschaulichen. Ausgehend von einer $n \times m$-Matrix $S = (s_1, \ldots, s_m)$, deren Spalten orthogonalisiert werden sollen, erhält man nach dem *Gram-Schmidt*-Prozeß als Resultat die Matrix $Q = (q_1, \ldots, q_m)$. In Abhängigkeit vom Rundungsfehler ε und der Konditionszahl $\kappa_2(S)$ gilt die Abschätzung

$$Q^T Q = I + E \qquad \text{mit} \qquad \|E\|_2 \approx \varepsilon \kappa_2(S),$$

wobei $\kappa_2(S)$ das Verhältnis vom größten zum kleinsten Singulärwert von S widerspiegelt. Für den *Gram-Schmidt*-Prozesses im Verfahren von *Arnoldi* bedeutet dies, wenn $\kappa_2((v_1, \ldots, v_m, Av_m))$ groß ist, liegt Av_m „fast" im Unterraum, der von $\{v_1, \ldots, v_m\}$ aufgespannt wird. Eine Alternative hierzu ist die *Householder*-Orthogonalisierung, welche auch zuverlässig ist, wenn die zu orthogonalisierenden Vektoren nicht genügend unabhängig sind. Eine *Householder*-Transformation ist von der Form

$$P = I - 2uu^T \qquad \text{mit} \qquad \|u\|_2 = 1, \tag{31}$$

wobei I die Einheitsmatrix ist. Es gilt $P = P^T = P^{-1}$. Jedes Vielfache $w = \mu u$ von u mit $\mu \in \mathrm{R}$ und $\mu \neq 0$ liefert gemäß

$$P = T - \frac{ww^T}{\gamma} \qquad \text{mit} \qquad \gamma = \frac{w^T w}{2} \tag{32}$$

dieselbe *Householder*-Transformation [10]. Für die numerische Praxis ist die Darstellung (32) wegen der größeren Freiheit in der Wahl des Vektors w jedoch günstiger als (31). Um die Spalten der Matrix S zu orthogonalisieren, werden die *Householder*-Transformationen $P_1, \ldots, P_m$ so bestimmt, daß $P_m \ldots P_1 S = R$ eine obere Dreiecksmatrix ist. Wegen $S = P_1 \ldots P_m R$ liefert die Matrix Q, die aus den ersten m Spalten von $P_1 \ldots P_m$ besteht, die gewünschte Orthogonalisierung. In Abhängigkeit vom Rundungsfehler ε gilt nach [1] die Abschätzung

$$Q^T Q = I + E \qquad \text{mit} \qquad \|E\|_2 \approx \varepsilon.$$

Zunächst soll der Algorithmus angegeben werden, der unter Verwendung von *Householder*-Transformationen eine Orthonormalbasis des *Krylov*-Unterraumes K_m erzeugt.

1. ein beliebiger Vektor v_1 mit $\|v_1\|_2$ sei gegeben
2. Konstruktion von P_1, so daß $P_1 v_1 = e_1$
3. Iterationsprozeß für $j = 1, 2, \ldots$
   (a) Bildung von $v_j = P_1 \ldots P_j e_j$
   (b) Konstruktion von P_{j+1}, so daß $P_{j+1} \ldots P_1 (v_1, Av_1, \ldots, Av_j)$ eine obere Dreiecksmatrix ist

Nach [18] gilt der folgende Satz:

SATZ 2:

Die nach dem obigen Algorithmus gebildete Menge $V_j^H = \{v_1, \ldots, v_j\}$ *ist eine Orthonormalbasis von* $K_j(A, v_1)$.

Die *Gram-Schmidt*-Orthogonalisierung wird nun durch die *Householder*-Orthogonalisierung ersetzt. Dazu wird $v_1 = \pm r_0/\|r_0\|_2$ verwendet, wobei aufgrund numerischer Aspekte das Vorzeichen so gewählt wird, daß die erste Komponente von v_1 positiv ist.
Mit der oberen *Hessenberg*-Matrix $\bar{H}_m$,

$$P_{m+1} \ldots P_1 (Av_1, \ldots, Av_m) = \bar{H}_m,$$

ist das folgende Minimumproblem zu lösen.

$$\begin{array}{lll} \underset{z \in K_m(A, r_0)}{min} \|b - A(x_0 + z)\|_2 & = & \underset{y \in \mathrm{R}^m}{min} \|r_0 - AV_m^H y\|_2 \\ & = & \underset{y \in \mathrm{R}^m}{min} \| \pm \|r_0\|_2 e_1 - \bar{H}_m y\|_2 \\ & = & \underset{y \in \mathrm{R}^m}{min} \| \pm \|r_0\|_2 Q_m^T e_1 - R_m y\|_2 \end{array} \tag{33}$$

Der vollständige Algorithmus nach [18] lautet folgendermaßen:

1. Bestimmung des Anfangsresiduums r_0 aus x_0; $r_0 = b - Ax_0$
2. Konstruktion von P_1, so daß $P_1 r_0 = \pm\|r_0\|_2 e_1 \equiv w$
3. Iterationsprozeß für $j = 1, 2, \ldots$
   (a) Bildung von $v_j = P_j \ldots P_1 A P_1 \ldots P_j e_j$
   (b) Konstruktion von P_{j+1}:
   die ersten j Komponenten des zugehörigen *Householder*-Vektors sind 0 und die Komponenten ab Position $j+2$ von $P_{j+1}v_j$ sind 0
   (c) $\bar{h}_j = P_{j+1} v_j$, $\bar{h}_j = (h_{1j}, \ldots, h_{jj}, h_{j+1j})^T$
   (d) für $j > 1$:
   $G_{j-1j} \ldots G_{1j} \bar{h}_j = \tilde{r}_j$, $G_{lj} \in \mathrm{R}^{(j+1)\times(j+1)}$,
   $\tilde{r}_j = (\tilde{r}_{1j}, \ldots, \tilde{r}_{j+1j})^T$
   (e) Bestimmung von G_{jj}:
   $G_{jj}\tilde{r}_j = G_{jj} G_{j-1j} \ldots G_{1j} \bar{h}_j = r_j$
   r_j - j-te Spalte der Matrix R_m (bzw. R_j)
   $r_j = (\hat{r}_j, 0)^T$
   $\hat{r}_j$ - j - te Spalte der Matrix $\hat{R}_m$ (bzw. $\hat{R}_j$)
   $w := G_{jj} w = G_{jj} G_{j-1j-1} \ldots G_{11} | \prod_{i=1}^{j} s_i |$
   (f) $p_j = \|b - Ax_j\|_2 = |w_{j+1}| = \|r_0\|_2 | \prod_{i=1}^{j} s_i |$

(g) Entscheidung über die Akzeptanz der Näherungslösung:
falls $p_j \leq \varepsilon$, dann $m = j$ und *goto* (4)

4. y_m^H sei die Lösung des Minimumproblrems (33):
$z_m^H = V_m^H y_m^H = (P_1 e_1, P_1 P_2 e_2, \ldots, P_1 \ldots P_m e_m)(y_{m_1}^H, \ldots, y_{m_m}^H)^T$ und $x_m^H = x_0 + z_m^H$

In diesem Algorithmus werden anstelle der Orthonormalbasisvektoren $v_1, \ldots, v_m$ die die *Householder*-Matrizen $P_1, \ldots, P_m$ erzeugenden *Householder*-Vektoren gespeichert. Jeder Basisvektor $v_i = P_1 \ldots P_i e_i, i = 1, \ldots, m$, wird zu dem Zeitpunkt generiert, wo er benutzt wird.
Es wird jetzt noch eine Variation des vorhergehenden Algorithmus angegeben. Diese Variation beinhaltet eine gewisse Parallelität des Verfahrens und basiert auf dem folgenden Sachverhalt (siehe [18]).
Für die Vektoren $v, s_1, t_1, \ldots, s_m, t_m \in \mathrm{R}^m$ kann man die Beziehung

$$(I + s_m t_m^T) \ldots (I + s_1 t_1^T) v = v + \sum_{i=1}^{m} d_i s_i$$

aufstellen, wobei der Vektor $d = (d_1, \ldots, d_m)^T \in \mathrm{R}^m$ bestimmt ist durch

$$\begin{aligned} d_1 &= s_1^T v \\ d_k &= s_k^T v + \sum_{i=1}^{k-1} d_i s_k^T t_k \quad , \quad k = 2, \ldots, m. \end{aligned} \tag{34}$$

(34) ist äquivalent zur Lösung des Gleichungssystems $Ld = S^T v$, wobei $S = (s_1, \ldots, s_m)$ und

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -s_2^T t_1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -s_m^T t_1 & \cdots & -s_m^T t_{m-1} & 1 \end{pmatrix}.$$

Mit $T = (t_1, \ldots, t_m)$ kann man dann

$$\begin{aligned} (I + s_m t_m^T) \ldots (I + s_1 t_1^T) v &= v + Td \\ &= v + TL^{-1} S^T v \\ &= (I + TL^{-1} S^T) v \end{aligned} \tag{35}$$

schreiben.
Mit Hilfe der *Householder*-Transformation (32) und den Ersetzungen $t_k = w_k$ und $s_k = -\frac{w_k}{\gamma_k} = -\frac{w_k}{|w_{kk}|}$ erhält man nach (35)

$$\begin{aligned} P_m \ldots P_1 v &= (I - \tfrac{w_m w_m^T}{|w_{mm}|}) \ldots (I - \tfrac{w_1 w_1^T}{|w_{11}|}) v \\ &= (I - (w_1, \ldots, w_m) L_m^{-1} (\tfrac{w_1}{|w_{11}|}, \ldots, \tfrac{w_m}{|w_{mm}|})^T) v, \end{aligned} \tag{36}$$

wobei

$$L_m = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \frac{w_2^T w_1}{|w_{22}|} & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \frac{w_m^T w_1}{|w_{mm}|} & \cdots & \frac{w_m^T w_{m-1}}{|w_{mm}|} & 1 \end{pmatrix}.$$

Mit $W_m = (w_1, \ldots, w_m)$ und $\bar{W}_m = (\frac{w_1}{w_{11}|}, \ldots, \frac{w_m}{|w_{mm}|})$ ergibt sich (36) zu
$P_m \ldots P_1 v = (I - W_m L_m^{-1} \bar{W}_m^T)v$. Weiterhin gilt

$$\begin{aligned} P_1 \ldots P_m v &= (P_m \ldots P_1)^T v \\ &= (I - W_m L_m^{-1} \bar{W}_m^T)^T v \\ &= (I - \bar{W}_m L_m^{-T} W_m^T)v. \end{aligned}$$

Nach Kostruktion sind die Matrizen L_j in den Matrizen L_{j+1} für $j = 1, \ldots, m-1$ enthalten. Der Punkt 4. des vorhergehenden Algorithmus ändert sich dann wie folgt:

$$\begin{aligned} (P_1 e_1, \ldots, P_1 \ldots P_m e_m) &= ((I - \bar{W}_1 L_1^{-T} W_1^T)e_1, \ldots, (I - \bar{W}_m L_m^{-T} W_m^T)e_m) \\ &= ((I - \bar{W}_m L_m^{-T} W_m^T)e_1, \ldots, (I - \bar{W}_m L_m^{-T} W_m^T)e_m) \\ &= (I - \bar{W}_m L_m^{-T} W_m^Y)(e_1, \ldots, e_m). \end{aligned}$$

Hiermit erhält man

$$z_m^H = (I - \bar{W}_m L_m^{-T} W_m^T)(e_1, \ldots, e_m) y_m^H.$$

3 Das QMR-Verfahren

Es wird das QMR-Verfahren auf der Basis des *look-ahead Lanczos*-Prozesses nach [8] beschrieben.

3.1 Der klassische Lanczos-Algorithmus

Die *Lanczos*-Methode im klassischen Sinne bedeutet die Reduktion einer nichtsymmetrischen allgemeinen Matrix $A \in \mathrm{R}^{n \times n}$ auf Tridiagonalform. Der Algorithmus startet mit Vektoren $v_1 \in \mathrm{R}^n$ und $w_1 \in \mathrm{R}^n$, wobei $v_1 \neq 0$ und $w_1 \neq 0$. Dann werden Basen $\{v_i\}$ und $\{w_i\}$ für die *Krylov*-Unterräume $K_n(A, v_1)$ bzw. $K_n(A^T, w_1)$ mit der Biorthogonalitätsbedingung

$$(w_j, v_k) = \begin{cases} 1 & \textit{für} \quad k = j \\ 0 & \textit{sonst} \end{cases} \tag{37}$$

konstruiert.
Die beiden Basen werden mit einer 3-Term-Rekursion gebildet. Es werden pro Schritt wenig Operationen durchgeführt und minimaler Speicherplatz benutzt. Der klassische *Lanczos*-Algorithmus lautet folgendermaßen (siehe [8]):

1. wähle: $\bar{v}_1, \bar{w}_1 \in \mathrm{R}^n$ mit $\bar{v}_1, \bar{w}_1 \neq 0$
   setze: $v_0 = w_0 = 0$

2. Iterationprozeß für $j = 1, 2, \ldots$

   (a) berechne: $\delta_j = (\bar{w}_j, \bar{v}_j)$
   falls $\delta_j = 0$, dann $m = j - 1$ und stop

(b) wähle: $\beta_j, \gamma_j \in \mathrm{R}$ mit $\beta_j \gamma_j = \delta_j$
setze: $v_j = \bar{v}_j / \gamma_j$ und $w_j = \bar{w}_j / \beta_j$

(c) $\alpha_j = (w_j, Av_j)$
$\bar{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$
$\bar{w}_{j+1} = A^T w_j - \alpha_j w_j - \gamma_j w_{j-1}$
falls $\bar{v}_{j+1} = 0$ oder $\bar{w}_{j+1} = 0$, dann $m = j$ und stop

Die spezielle Wahl von α_j, β_j und γ_j sichert die Biorthogonalität (37). Die Basen $\{v_j\}$ und $\{w_j\}$ werden zusammengefaßt zu $V_m = \{v_1, \ldots, v_m\}$ bzw. $W_m = \{w_1, \ldots, w_m\}$. Mit der *Hessenberg*-Matrix $\bar{H}_m \in \mathrm{R}^{(m+1)\times m}$,

$$\bar{H}_m = \begin{pmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \gamma_2 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_n \\ \vdots & & \ddots & \gamma_n & \alpha_n \\ 0 & \cdots & \cdots & 0 & \gamma_{n+1} \end{pmatrix},$$

und $H_m = [I_m, 0]\bar{H}_m$ (Streichung der letzten Zeile) erhält man

$$\begin{aligned} AV_m &= V_m H_m + [0, \ldots, 0, \bar{v}_{m+1}] \\ A^T W_m &= W_m H_m^T + [0, \ldots, 0, \bar{w}_{m+1}]. \end{aligned}$$

Die Biorthogonalitätsbedingung (37) kann als

$$W_m^T V_m = I_m$$

geschrieben werden. Für symmetrische Matrizen $A = A^T$ kann man $\bar{w}_1 = \bar{v}_1$ mit $\beta_j = \gamma_j$ wählen.
Das *Lanczos*-Verfahren kann verschiedenartig abbrechen. Ein regulärer Abbruch liegt vor, wenn $v_{m+1} = 0$ oder $w_{m+1} = 0$. In diesem Fall spannen die rechten *Lanczos*-Vektoren $v_1, \ldots, v_m$ einen A-invarianten Unterraum auf bzw. die linken *Lanczos*-Vektoren $w_1, \ldots, w_m$ einen A^T-invarianten Unterraum. Ein möglicher irregulärer Abbruch der *Lanczos*-Methode ist $\delta_m = (\bar{w}_m, \bar{v}_m) = 0$, wobei weder $\bar{v}_m = 0$ noch $\bar{w}_m = 0$. Das bedeutet, daß die entsprechenden Basisvektoren $A^{m-1}v_1$ bzw. $(A^T)^{m-1}w_1$ nicht konstruierbar sind. Dies ist kein Problem der Skalierung, sondern die Biorthogonalitätsbedingung (37) kann nicht erfüllt werden. Es kann sein, daß (37) für eine höhere Potenz ($> m$) von A und A^T wieder erfüllt ist. Ein Algorithmus, der irgendwie auf ein solches Paar von *Lanczos*-Vektoren vorausschaut, wird *look-ahead Lanczos*-Algorithmus genannt.

3.2 Der look-ahead Lanczos-Algorithmus

Die Grundidee des *look-ahead Lanczos*-Algorithmus ist die Abschwächung der Biorthogonalitätsbedingung (37). Für jedes $m = 1, 2, \ldots$ werden die Vektoren $v_1, \ldots, v_m$ und

$w_1, \ldots, w_m$ durch den *look-ahead* Prozeß in $l = l(m)$ Blöcke unterteilt. Für $k = 1, \ldots, l-1$ seien die k-ten Blöcke der *Lanczos*-Vektoren durch

$$V^{(k)} = [v_{m_k}, v_{m_k+1}, \ldots, v_{m_{k+1}-1}] \quad , \quad W^{(k)} = [w_{m_k}, w_{m_k+1}, \ldots, w_{m_{k+1}-1}]$$

und der l-te (aktuelle) Block durch

$$V^{(l)} = [v_{m_l}, v_{m_l+1}, \ldots, v_{m_{l+1}-1}] \quad , \quad W^{(l)} = [w_{m_l}, w_{m_l+1}, \ldots, w_{m_{l+1}-1}]$$

gegeben, wobei $1 = m_1 < m_2 < \ldots < m_k < \ldots < m_l \leq m < m_{l+1}$. Die so konstruierten Blöcke schwächen die Biorthogonalitätsbedingung (37) zu

$$(W^{(j)})^T V^{(k)} = \begin{cases} D^{(k)} & \text{für} \quad j = k \\ 0 & \text{sonst} \end{cases} \quad , \quad j, k = 1, \ldots, l, \tag{38}$$

ab, wobei die Matrizen $D^{(k)}$ nichtsingulär sind für $k = 1, \ldots, l-1$. Die Matrix $D^{(l)}$ ist nichtsingulär, wenn der l-te Block vollständig ist, d.h. $m = m_{l+1} - 1$. Die ersten Vektoren v_{m_k} und w_{m_k} eines jeden Blockes werden reguläre Vektoren genannt, die anderen innere Vektoren.
Es wird jetzt die Basisstruktur des *look-ahead Lanczos*-Algorithmus nach [9] angegeben.

1. wähle: $v_1, w_1 \in \mathrm{R}^n$ mit $\|v_1\|_2 = 1$ und $\|w_1\|_2 = 1$
   setze: $V^{(1)} = v_1, W^{(1)} = w_1, D^{(1)} = (W^{(1)})^T V^{(1)}$,
   setze: $m_1 = 1, k = 1, v_0 = 0, w_0 = 0, V^{(0)} = 0, W^{(0)} = 0, \varrho_1 = 1, \xi_1 = 1$

2. Iterationsprozeß für $m = 1, 2, \ldots$

   (a) Entscheidung über die Konstruktion von v_{m+1} und w_{m+1} als reguläre oder innere Vektoren und *goto* (b) bzw. (c)

   (b) Regulärer Schritt:

$$\begin{aligned} \bar{v}_{m+1} &= Av_m &&- V^{(k)}(D^{(k)})^{-1}(W^{(k)})^T Av_m \\ &&&- V^{(k-1)}(D^{(k-1)})^{-1}(W^{(k-1)})^T Av_m \\ \bar{w}_{m+1} &= A^T w_m &&- W^{(k)}(D^{(k)})^{-T}(V^{(k)})^T A^T w_m \\ &&&- W^{(k-1)}(D^{(k-1)})^{-T}(V^{(k-1)})^T A^T w_m \end{aligned}$$

   setze: $m_{k+1} = m+1, k = k+1, V^{(k)} = 0, W^{(k)} = 0$, *goto* (d)

   (c) Innerer Schritt:

$$\begin{aligned} \bar{v}_{m+1} &= Av_m &&- \zeta_m v_m - \tfrac{\eta_m}{\varrho_m} v_{m-1} \\ &&&- V^{(k-1)}(D^{(k-1)})^{-1}(W^{(k-1)})^T Av_m \\ \bar{w}_{m+1} &= A^T w_m &&- \zeta_m w_m - \tfrac{\eta_m}{\xi_m} w_{m-1} \\ &&&- W^{(k-1)}(D^{(k-1)})^{-T}(V^{(k-1)})^T A^T w_m \end{aligned}$$

   Die Wahl der Rekursionskoeffizienten ζ_m und η_m ist nicht sehr wichtig, da der Algorithmus im allgemeinen sehr kleine Blöcke bildet. Wie numerische Beispiele zeigen (siehe [7], [12]), kann man $\zeta_m = 1$ und $\eta_m = 1$ wählen.

(d) $\varrho_{m+1} = \|\bar{v}_{m+1}\|_2 \quad , \quad \xi_{m+1} = \|\bar{w}_{m+1}\|_2$
falls $\varrho_{m+1} = 0$ oder $\xi_{m+1} = 0$, dann stop
$v_{m+1} = \bar{v}_{m+1}/\varrho_{m+1} \quad , \quad w_{m+1} = \bar{w}_{m+1}/\xi_{m+1}$
$V^{(k)} = [V^{(k)}, v_{m+1}]$, $W^{(k)} = [W^{(k)}, w_{m+1}]$, $D^{(k)} = (W^{(k)})^T V^{(k)}$

Schritt (2c) ist eine Blockversion des klassischen *Lanczos*-Algorithmus. Die inneren Vektoren überspannen den „Graben" zwischen zwei regulären Vektoren. Die obere *Hessenberg*-Matrix $\bar{H}_m$ ist hierbei von block-tridiagonaler Gestalt mit Blöcken der Größe $(m_k - m_{k-1}) \times (m_k - m_{k-1})$ auf der Diagonalen. Mit $V_m = [V^{(1)}, \ldots, V^{(l)}]$ und $W_m = [W^{(1)}, \ldots W^{(l)}]$ geht (38) in

$$W_m^T V_m = D_m = diag(D^{(1)}, \ldots, D^{(l)})$$

über. D_m ist nichtsingulär, wenn $m = m_{l+1} - 1$. Werden nur reguläre Schritte durchgeführt, ist dieser *look-ahead Lanczos*-Algorithmus mit dem klassischen *Lanczos*-Prozeß identisch. Für die Entscheidung, ob ein regulärer oder ein innerer Schritt durchgeführt wird, ist es notwendig, daß die Matrix $D^{(k)}$ nichtsingulär ist. Hierzu kann man den kleinsten Singulärwert $\sigma_{min}(D^{(k)})$ von $D^{(k)}$ überprüfen. Es muß auch noch gesichert sein, daß sowohl die rechten *Lanczos*-Vektoren $v_1, \ldots, v_m$ als auch die linken *Lanczos*-Vektoren $w_1, \ldots, w_m$ linear unabhängig sind.

3.3 Das QMR-Verfahren

Die grundlegende Idee des QMR-Verfahrens ist die Erzeugung der Matrix V_m mittels kurzer Rekursionen mit Hilfe des *look-ahead Lanczos*-Prozesses, wobei dann aber die Minimierungsbedingung (12) ein wenig abgeschwächt wird. Auch dieses Verfahren liefert nicht das „ideale *Krylov*-Verfahren". Es müßte die beiden folgenden Kriterien gleichzeitig erfüllen:

1. Die Iterierten erfüllen eine Minimumbedingung (z.B. (12)), und
2. der Arbeits- und Speicheraufwand pro Iteration bleibt annähernd konstant.

Nach [4], [5] gibt es für allgemeine Matrizen A kein CG-ähnliches Verfahren, daß die beiden Bedingungen gleichzeitig erfüllt. Nur für Matrizen der Form

$$A = e^{i\theta}(B + \sigma I) \ , \quad \text{mit} \quad B = B^T \quad \text{und} \quad \theta, \sigma \in \mathrm{R}$$

kann man solche Verfahren konstruieren.
Der QMR-Algorithmus lautet folgendermaßen:

1. Bestimmung des Anfangsresiduums r_0 aus x_0; $r_0 = b - Ax_0$
2. Bestimmung der ersten Basisvektoren; $v_1 = r_0/\|r_0\|_2$, $w_1 \equiv v_1$
3. Iterationsprozeß für $m = 1, 2, \ldots$

(a) Bestimmung der Matrizen V_m, V_{m+1} und $\bar{H}_m$ mit dem *look-ahead Lanczos*-Algorithmus

(b) Minimierung des Residuums mittels *Givens*-Rotationen (siehe 2.2)
Aus $x_m = x_0 + V_m y_m$ folgt $r_m = r_0 - AV_m y_m$. Mit (14) erhält man $r_m = v_1\|r_0\|_2 - V_{m+1}\bar{H}_m y_m$ und hieraus

$$r_m = V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m y_m). \tag{39}$$

Da V_{m+1} nicht unitär ist, wird der Koeffizientenvektor y_m minimiert. Die Größe y_m^L sei die Lösung des Minimumproblems

$$\|\|r_0\|_2 e_1 - \bar{H}_m y_m^L\|_2 \;=\; \min_{y_m \in \mathrm{R}^m} \|\|r_0\|_2 e_1 - \bar{H}_m y_m\|_2. \tag{40}$$

Der Minimierungsaufwand für (39) würde sonst für die Anzahl der Operationen $O(n \cdot m^2)$ und für den Speicherbedarf $O(n \cdot m)$ betragen.
Daß (40) gerechtfertigt ist, zeigt folgende Überlegung:
Es ist

$$r_m = b - Ax_m = V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m y_m).$$

Aus $W_{m+1}^T V_{m+1} = D_{m+1}$ folgt $D_{m+1}^{-1} W_{m+1}^T V_{m+1} = I_{m+1}$ und damit

$$D_{m+1}^{-1} W_{m+1}^T V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m y_m) = D_{m+1}^{-1} W_{m+1}^T (b - Ax_m).$$

Somit erhält man

$$\|r_0\|_2 e_1 - \bar{H}_m y_m = D_{m+1}^{-1} W_{m+1}^T (b - Ax_m). \tag{41}$$

(41) zeigt, daß das Minimumproblem (40) äquivalent zur Minimierung des Residuums in wechselnden Unterräumen ist , d.h.

$$\min_{x_m \in x_0 + K_m(A, r_0)} \|D_{m+1}^{-1} W_{m+1}^T (b - Ax_m)\|_2.$$

(c) Bildung von x_m^L mit Konvergenzüberprüfung (siehe 2.2)

$$x_m^L = x_0 + V_m y_m^L$$

Eine Vereinfachung erhält man dadurch, wenn man eine Matrix P konstruiert, so daß

$$A^T P = PA \tag{42}$$

gilt. P ist eine symmetrische und nichtsinguläre Matrix. Unter Benutzung des Startvektors $w_1 = \frac{Pv_1}{\|Pv_1\|_2}$ gilt dann $w_m = \frac{Pv_m}{\|Pv_m\|_2}$ für alle m.

$$\begin{aligned}
\bar{w}_{m+1} &= A^T w_m && - W^{(k)}(D^{(k)})^{-T}(V^{(k)})^T A^T w_m \\
&&& - W^{(k-1)}(D^{(k-1)})^{-T}(V^{(k-1)})^T A^T w_m \\
&= \tfrac{1}{\|Pv_m\|_2}\{A^T P v_m && - W^{(k)}(D^{(k)})^{-T}(V^{(k)})^T A^T P v_m \\
&&& - W^{(k-1)}(D^{(k-1)})^{-T}(V^{(k-1)})^T A^T P v_m\} \\
&= \tfrac{1}{\|Pv_m\|_2}\{PAv_m && - W^{(k)}(D^{(k)})^{-T}(V^{(k)})^T PAv_m \\
&&& - W^{(k-1)}(D^{(k-1)})^{-T}(V^{(k-1)})^T PAv_m\} \\
&= \tfrac{1}{\|Pv_m\|_2}\{PAv_m && - PW^{(k)}(D^{(k)})^{-T}(V^{(k)})^T Av_m \\
&&& - PW^{(k-1)}(D^{(k-1)})^{-T}(V^{(k-1)})^T Av_m\} \\
&= \tfrac{P\bar{v}_{m+1}}{\|Pv_m\|_2}
\end{aligned}$$

Hieraus folgt

$$w_{m+1} = \frac{\bar{w}_{m+1}}{\|\bar{w}_{m+1}\|_2} = \frac{P\bar{v}_{m+1}}{\|P\bar{v}_{m+1}\|_2} = \frac{Pv_{m+1}}{\|Pv_{m+1}\|_2}.$$

4 Einige Bemerkungen zu GMRES und QMR

Mit wachsendem m nimmt der Arbeits- und Speicheraufwand des GMRES-Algorithmus zu. Deshalb ist es sinnvoll, eine maximale Dimension m_{max} des *Krylov*-Unterraumes $K_m(A, r_0)$ vorzugeben. Ist die Konvergenz nach m_{max} Iterationsschritten nicht erreicht, wird die Iterierte x_m als neuer Startwert genommen, d.h. $x_0 := x_m$. Diese Variante mit „*restart*" erlaubt es, den Speicheraufwand in Grenzen zu halten. Andererseits kann sich dadurch die Konvergenz verlangsamen.
Das QMR-Verfahren löst nach [8] dasselbe Approximationsproblem wie GMRES und hat auch dasselbe quantitative Konvergenzverhalten, wenn man von der Optimalität(GMRES) zur quasi-optimalen Approximation (QMR)übergeht.

4.1 Ein einfacher Vergleich

Jetzt soll ein einfacher Vergleich zwischen beiden Algorithmen bzgl.des Rechenaufwandes gemacht werden. Dazu wird das QMR-Verfahren mit „*restart*" betrachtet. Der Vergleich soll sich nur auf die Anzahl der durchgeführten Matrix $*$ Vektor-Operationen und auf die Anzahl der zu berechnenden Skalarprodukte für die Erzeugung der Matrix $\bar{H}_m$ beziehen. Außerdem wird noch vorausgesetzt, daß der *look-ahead Lanczos*-Algorithmus nur reguläre Vektoren erzeugt.
Einige Bezeichnungen:

$c_G(MV)$	–	Anzahl der Matrix $*$ Vektor-Operationen für GMRES
$c_G(SP)$	–	Anzahl der zu berechnenden Skalarprodukte für GMRES
$c_Q(MV)$	–	Anzahl der Matrix $*$ Vektor-Operationen für QMR
$c_Q(M^TV)$	–	Anzahl der (Matrix)$^T*$ Vektor-Operationen für QMR
$c_Q(SP)$	–	Anzahl der zu berechnenden Skalarprodukte für QMR

Betrachtet man für die Matrix A eine spaltenweise Speicherung mit Diagonalbehandlung (siehe 1.3), so kann eine Matrix $*$ Vektor- bzw. (Matrix)$^T*$ Vektor-Operation durch nz Skalarprodukte abgeschätzt werden.
Hiermit erhält man folgende Abschätzungen:

GMRES: $c_G(MV) = m_{max} + 1$
$c_G(SP) = \frac{m_{max}(m_{max}+1)}{2} + \theta$
θ - Anzahl der Skalarprodukte für die Nachorthogonalisierung,
$0 \leq \theta \leq \frac{(m_{max}-1)m_{max}}{2}$

QMR: $c_Q(MV) = m_{max} + 1$
$c_Q(M^TV) = m_{max}$
$c_Q(SP) = 4m_{max} - 3$

$c_Q(SP)$ kann man nach [9] noch auf $2m_{max}$ Skalarprodukte mit gewissem Rekursionsaufwand für die anderen Skalarprodukte reduzieren. Betrachtet man die Konstruktion regulärer Vektoren nach dem *look-ahead Lanczos*-Algorithmus, so ergeben sich nach Punkt 3.2 die folgenden Berechnungsvorschriften.

$$\begin{aligned}
\bar{v}_{m+1} &= Av_m - v_m d_m^{-1} w_m^T A v_m - v_{m-1} d_{m-1}^{-1} w_{m-1}^T A v_m \\
\bar{w}_{m+1} &= A^T w_m - w_m d_m^{-1} v_m^T A^T w_m - w_{m-1} d_{m-1}^{-1} v_{m-1}^T A^T w_m \\
d_m &= w_m^T v_m
\end{aligned}$$

Es ist

$$w_m^T A v_m = (w_m, A v_m) = v_m^T A^T w_m.$$

Weiterhin folgt nach der obigen Berechnungsvorschrift und nach der Biorthogonalitätsbedingung (38)

$$\begin{aligned}
w_{m-1}^T A v_m &= (A^T w_{m-1})^T v_m \\
&= (\bar{w}_m + w_{m-1}(d_{m-1}^{-1} v_{m-1}^T A^T w_{m-1}) + w_{m-2}(d_{m-2}^{-1} v_{m-2}^T A^T w_{m-1}))^T v_m \\
&= (\bar{w}_m^T + (d_{m-1}^{-1} v_{m-1}^T A^T w_{m-1})^T w_{m-1}^T + (d_{m-2}^{-1} v_{m-2}^T A^T w_{m-1})^T w_{m-2}^T) v_m \\
&= \bar{w}_m^T v_m
\end{aligned}$$

und

$$\begin{aligned}
v_{m-1}^T A^T w_m &= (A v_{m-1})^T w_m \\
&= (\bar{v}_m + v_{m-1}(d_{m-1}^{-1} w_{m-1}^T A v_{m-1}) + v_{m-2}(d_{m-2}^{-1} w_{m-2}^T A v_{m-1}))^T w_m \\
&= (\bar{v}_m^T + (d_{m-1}^{-1} w_{m-1}^T A v_{m-1})^T v_{m-1}^T + (d_{m-2}^{-1} w_{m-2}^T A v_{m-1})^T v_{m-2}^T) w_m \\
&= \bar{v}_m^T w_m.
\end{aligned}$$

Hieraus folgt:

$$w_{m-1}^T A v_m = \bar{w}_m^T v_m = \|\bar{w}_m\|_2 w_m^T v_m = \|\bar{w}_m\|_2 d_m$$

und

$$v_{m-1}^T A^T w_m = \bar{v}_m^T w_m = \|\bar{v}_m\|_2 v_m^T w_m = \|\bar{v}_m\|_2 d_m.$$

Nun kann man eine einfache Beziehung aufstellen zwischen der maximalen Dimension des *Krylov*-Unterraumes und den Kosten für GMRES und QMR.

$$0 \approx c_G(MV) + c_G(SP) - (c_Q(MV) + c_Q(M^T V) + c_Q(SP))$$

Dies ist gleichbedeutend mit

$$\frac{m_{max}(m_{max}+1)}{2} + \theta - (nz+2)m_{max} = 0. \tag{43}$$

Hieraus erhält man

$$\underline{m}_{max} \leq m_{max} \leq \overline{m}_{max} \tag{44}$$

mit

$$\underline{m}_{max} = nz + 2 \tag{45}$$

und

$$\overline{m}_{max} = 2nz + 3. \tag{46}$$

Aus den Abschätzungen (44) - (46) kann man schlußfolgern, daß für großes nz der Rechenaufwand des QMR-Algorithmus gegenüber GMRES erheblich ansteigt, weil die $(\text{Matrix})^T*$ Vektor-Operation nicht vernachlässigbar ist.
Andererseits bietet der QMR-Algorithmus die Möglichkeit, ohne großen zusätzlichen Rechenaufwand auf Normalgleichungssysteme der Gestalt (5) umzuschalten, so daß man hier verschiedene *Krylov*-Unterraum-Methoden kombinieren kann.
Die Entscheidung, ob GMRES oder QMR anzuwenden ist, hängt von der konkreten Problemstellung ab.

Literatur

[1] A.Bjorck, Solving linear least-squares problems by Gram-Schmidt orthogonalization, BIT, 7(1967), pp. 1-21.

[2] P.N.Brown, A theoretical comparison of the Arnoldi and GMRES algorithms, SIAM J. Sci. Statist. Comput. 12(1)(1991), pp. 58-78.

[3] H.C.Elman,Iterative Methods for Large Sparse Nonsymmetric Systems of Linear Equations, PhD thesis, Computer Science Dept., Yale University, New Haven, CT, 1982.

[4] V.Faber and T.Manteuffel, Necessary and sufficient conditions for the existence of a conjugate gradient method, SIAM J. Numer. Anal. 21(1984), pp. 352-362.

[5] V.Faber and T.Manteuffel, Orthogonal error methods, SIAM J. Numer. Anal. 24(1987), pp. 170-187.

[6] R.Fletcher, Conjugate gradient methods for indefinite systems, In G.A.Watson, editor, Proceedings of the Dundee Biennial Conference on Numerical Analysis 1974, pages 73-89, University of Dundee, Scotland, Springer-Verlag, New-York, 1975.

[7] R.W.Freund, Krylov subspace methods for complex non-Hermitian linear systems, Habilitationsschrift, Universität Würzburg, 1991.

[8] R.W.Freund, G.H.Golub and N.M.Nachtigal, Iterative Solution of Linear Systems, Acta Numerica (1991), pp. 1-44.

[9] R.W.Freund, M.H.Gutknecht and N.M.Nachtigal, An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, Technical Report 91.09, RIACS, NASA Ames Research Center, Moffet Field.

[10] A.Kielbasinski, H.Schwetlick, Numerische lineare Algebra, VEB Deutscher Verlag der Wissenschaften, Berlin 1988.

[11] C.Lanczos, Solution of systems of linear equations by minimized iterations, J. Res. Nat. Bur. Standards, 49(1952), pp. 33-53.

[12] N.M.Nachtigal, A look-ahead variant of the Lanczos algorithm and its application to the quasi-minimal residual method for non-Hermitian linear systems, PhD thesis, Massachusetts Institute of Technology, Cambridge,MA, August 1991.

[13] N.M.Nachtigal, S.C.Reddy and L.N.Trefethen, How fast are nonsymmetric matrix iterations?, SIAM J. Matrix Anal. Appl.,to appear, 1990.

[14] C.C.Paige and M.A.Saunders, Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal., 12(1975), pp. 617-629.

[15] Y.Saad, Krylov subspace methods for solving unsymmetric linear systems, Math. Comp., 37(1981), pp. 105-126.

[16] Y.Saad and M.H.Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statst. Comput., 7(1986), pp. 856-869.

[17] P.Sonnfeld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Statist. Comput., 10(1)(1989), pp. 36-52.

[18] H.F.Walker, Implementation of the GMRES method using Householder transformations, SIAM J. Sci. Statist. Comput., 9(1988) , pp. 152-163.

Veröffentlichungen des Instituts für Angewandte Analysis und Stochastik

Preprints 1992

1. D.A. Dawson and J. Gärtner: Multilevel large deviations.
2. H. Gajewski: On uniqueness of solutions to the drift–diffusion–model of semiconductor devices.
3. J. Fuhrmann: On the convergence of algebraically defined multigrid methods.
4. A. Bovier and J.-M. Ghez: Spectral properties of one–dimensional Schrödinger operators with potentials generated by substitutions.
5. D.A. Dawson and K. Fleischmann: A super–Brownian motion with a single point catalyst.
6. A. Bovier, V. Gayrard: The thermodynamics of the Curie–Weiss model with random couplings.
7. W. Dahmen, S. Prößdorf, R. Schneider: Wavelet approximation methods for pseudodifferential equations I: stability and convergence.
8. A. Rathsfeld: Piecewise polynomial collocation for the double layer potential equation over polyhedral boundaries. Part I: The wedge, Part II: The cube.
9. G. Schmidt: Boundary element discretization of Poincaré–Steklov operators.
10. K. Fleischmann, F. I. Kaj: Large deviation probability for some rescaled superprocesses.
11. P. Mathé: Random approximation of finite sums.
12. C.J. van Duijn, P. Knabner: Flow and reactive transport in porous media induced by well injection: similarity solution.
13. G.B. Di Masi, E. Platen, W.J. Runggaldier: Hedging of options under discrete observation on assets with stochastic volatility.
14. J. Schmeling, R. Siegmund–Schultze: The singularity spectrum of self-affine fractals with a Bernoulli measure.
15. A. Koshelev: About some coercive inequalities for elementary elliptic and parabolic operators.
16. P.E. Kloeden, E. Platen, H. Schurz: Higher order approximate Markov chain filters.

17. H.M. Dietz, Y. Kutoyants: A minimum–distance estimator for diffusion processes with ergodic properties.

18. I. Schmelzer: Quantization and measurability in gauge theory and gravity.

19. A. Bovier, V. Gayrard: Rigorous results on the thermodynamics of the dilute Hopfield model.

20. K. Gröger: Free energy estimates and asymptotic behaviour of reaction–diffusion processes.

21. E. Platen (ed.): Proceedings of the 1^{st} workshop on stochastic numerics.

22. S. Prößdorf (ed.): International Symposium "Operator Equations and Numerical Analysis" September 28 – October 2, 1992 Gosen (nearby Berlin).

23. K. Fleischmann, A. Greven: Diffusive clustering in an infinite system of hierarchically interacting diffusions.

24. P. Knabner, I. Kögel–Knabner, K.U. Totsche: The modeling of reactive solute transport with sorption to mobile and immobile sorbents.

25. S. Seifarth: The discrete spectrum of the Dirac operators on certain symmetric spaces.

26. J. Schmeling: Hölder continuity of the holonomy maps for hyperbolic basic sets II.

27. P. Mathé: On optimal random nets.

28. W. Wagner: Stochastic systems of particles with weights and approximation of the Boltzmann equation. The Markov process in the spatially homogeneous case.

29. A. Glitzky, K. Gröger, R. Hünlich: Existence and uniqueness results for equations modelling transport of dopants in semiconductors.

30. J. Elschner: The *h-p*–version of spline approximation methods for Mellin convolution equations.