
GMRES und verwandte Verfahren

Dieses Kapitel beschäftigt sich mit dem so genannten GMRES-Verfahren und einigen Varianten zur Lösung von linearen Gleichungssystemen mit einer nur noch regulären Koeffizientenmatrix. Für solche Gleichungssysteme nimmt das GMRES-Verfahren eine exponierte Stellung ein. Zum einen ist es stets wohldefiniert (im Gegensatz beispielsweise zu den noch zu besprechenden Verfahren im Kapitel 7), zum anderen findet es die exakte Lösung theoretisch nach endlich vielen (höchstens n) Schritten. Insofern handelt es sich bei dem GMRES-Verfahren um ein direktes Verfahren, das allerdings (wie bereits das CG-Verfahren) üblicherweise als ein iteratives Verfahren betrachtet wird, da gute Näherungen an die exakte Lösung oft schon nach sehr viel weniger als n Schritten erreicht werden. Der große Nachteil des GMRES-Verfahrens besteht in dem viel zu hohen Rechenaufwand und insbesondere Speicherplatzbedarf, weshalb wir uns in diesem und dem folgenden Kapitel noch ausgiebig mit geeigneten Varianten des GMRES-Verfahrens beschäftigen werden.

Inhaltlich beginnt das Kapitel mit dem Abschnitt 6.1, der sich mit einigen Eigenschaften der Krylov-Räume auseinandersetzt. Diese Krylov-Räume werden von nun an eine große Bedeutung erlangen. Der Abschnitt 6.2 führt dann das eigentliche GMRES-Verfahren ein. Eine erste Variante dieses Verfahrens ist die GMRES(m)-Methode aus dem Abschnitt 6.3, bei der es sich im Prinzip um das GMRES-Verfahren handelt, das allerdings nach jeweils m Iterationen neu gestartet wird, um den Rechen- und Speicherplatzaufwand des eigentlichen GMRES-Verfahrens in Grenzen halten zu können. Eine zweite Variante des GMRES-Verfahrens ist die im Abschnitt 6.4 besprochene MINRES-Methode zur Lösung eines linearen Gleichungssystemes mit einer symmetrischen (aber nicht notwendig positiv definiten) Koeffizientenmatrix. Auf das Problem der Prädiktionierung der hier beschriebenen Methoden gehen wir im Abschnitt 6.5 ein, und ein geeigneter Prädiktionierer wird mit der unvollständigen LR-Zerlegung im Abschnitt 6.6 besprochen.

6.1 Krylov-Räume

Als Vorbereitung zu den Verfahren in diesem und dem nachfolgenden Kapitel beschäftigen wir uns in diesem Abschnitt etwas ausführlicher mit dem Krylov-Raum, der bereits im Zusammenhang mit dem CG-Verfahren im Abschnitt 5.2 auftrat. Dieser Krylov-Raum wird bei der Konstruktion aller noch zu beschreibenden Verfahren benutzt werden.

Zunächst erinnern wir noch einmal an seine Definition: Der k -te Krylov-Raum zu einer gegebenen Matrix $A \in \mathbb{R}^{n \times n}$ und einem Vektor $b \in \mathbb{R}^n$ ist gegeben durch

$$\mathcal{K}_k(b, A) := \text{span}\{b, Ab, A^2b, \dots, A^{k-1}b\}.$$

Da $\mathcal{K}_k(b, A)$ von k Vektoren aufgespannt wird, ist die Dimension dieses Raumes höchstens gleich k . Der Fall $\dim(\mathcal{K}_k(b, A)) < k$ wird in dem folgenden Resultat betrachtet.

Satz 6.1. *Seien $A \in \mathbb{R}^{n \times n}$ eine reguläre Matrix, $b \in \mathbb{R}^n$ sowie $x^* := A^{-1}b$ die exakte Lösung des linearen Gleichungssystems $Ax = b$. Dann sind die folgenden Aussagen äquivalent:*

- (a) Die Vektoren $b, Ab, \dots, A^k b$ sind linear abhängig.
- (b) Es ist $\mathcal{K}_k(b, A) = \mathcal{K}_{k+1}(b, A)$.
- (c) Es gilt $A\mathcal{K}_k(b, A) \subseteq \mathcal{K}_k(b, A)$, d.h., $\mathcal{K}_k(b, A)$ ist ein A -invarianter Unterraum.
- (d) Es ist $x^* \in \mathcal{K}_k(b, A)$.

Beweis. (a) \implies (b): Nach Voraussetzung existiert ein $0 \leq \ell \leq k$, so dass $A^\ell b$ linear abhängig von $b, Ab, \dots, A^{\ell-1}b$ ist, also $A^\ell b = \sum_{i=0}^{\ell-1} \gamma_i A^i b$ für gewisse Koeffizienten $\gamma_i \in \mathbb{R}$ gilt. Multiplikation von links mit $A^{k-\ell}$ liefert

$$A^k b = \sum_{i=0}^{\ell-1} \gamma_i A^{k-\ell+i} b \in \mathcal{K}_k(b, A).$$

Folglich ist $\mathcal{K}_{k+1}(b, A) \subseteq \mathcal{K}_k(b, A)$ und daher natürlich $\mathcal{K}_{k+1}(b, A) = \mathcal{K}_k(b, A)$.

(b) \implies (c): Offenbar ist

$$\begin{aligned} A\mathcal{K}_k(b, A) &= A \text{span}\{b, Ab, \dots, A^{k-1}b\} \\ &= \text{span}\{Ab, A^2b, \dots, A^k b\} \\ &\subseteq \text{span}\{b, Ab, A^2b, \dots, A^k b\} = \mathcal{K}_{k+1}(b, A). \end{aligned}$$

Mit (b) folgt daher die Behauptung (c).

(c) \implies (d): Betrachte die durch $T(x) := Ax$ definierte Abbildung $T : \mathcal{K}_k(b, A) \rightarrow \mathbb{R}^n$. Nach Voraussetzung (c) ist $T(x) = Ax \in A\mathcal{K}_k(b, A) \subseteq$

$\mathcal{K}_k(b, A)$ für alle $x \in \mathcal{K}_k(b, A)$. Daher ist $T : \mathcal{K}_k(b, A) \rightarrow \mathcal{K}_k(b, A)$ eine Abbildung des Krylov-Raumes $\mathcal{K}_k(b, A)$ in sich. Die lineare Abbildung $T : \mathcal{K}_k(b, A) \rightarrow \mathcal{K}_k(b, A)$ ist außerdem injektiv, denn aus $T(x) = T(y)$ für $x, y \in \mathcal{K}_k(b, A)$ folgt $A(x - y) = 0$ und daher $x = y$ aufgrund der Regularität der Matrix A . Als injektive lineare Abbildung auf dem endlich-dimensionalen Vektorraum $\mathcal{K}_k(b, A)$ ist T dann bereits bijektiv, vergleiche hierzu die Dimensionsformel aus dem Satz 1.3. Da wir trivialerweise $b \in \mathcal{K}_k(b, A)$ haben, existiert deshalb genau ein $x \in \mathcal{K}_k(b, A)$ mit $Ax = T(x) = b$. Die Regularität von A impliziert dann $x = A^{-1}b = x^*$ und daher $x^* \in \mathcal{K}_k(b, A)$.

(d) \implies (a): Nach Voraussetzung (d) gilt $x^* \in \text{span}\{b, Ab, \dots, A^{k-1}b\}$. Also ist $b = Ax^* \in \text{span}\{Ab, A^2b, \dots, A^kb\}$. Folglich sind die Vektoren b, Ab, A^2b, \dots, A^kb linear abhängig. \square

Wir wollen jetzt noch andeuten, wie die Krylov-Räume zur Lösung eines linearen Gleichungssystems $Ax = b$ eingesetzt werden können. Dabei geben wir hier zunächst nur die grobe Idee an, während die Details den folgenden Abschnitten überlassen bleiben. Prinzipiell beschreiben wir zwei Ansätze zur Lösung eines linearen Gleichungssystems mittels der Krylov-Räume: Zum einen ist dies die Klasse der Minimum-Residuen-Verfahren, und zum anderen ist das die Klasse der Galerkin-Verfahren.

Wir beginnen zunächst mit den Minimum-Residuen-Verfahren. Dazu bezeichne $\|\cdot\|$ weiterhin stets die euklidische Vektornorm im \mathbb{R}^n . Ferner sei $B \in \mathbb{R}^{n \times n}$ eine reguläre Matrix, deren genaue Wahl an dieser Stelle nicht weiter spezifiziert werden soll. Gemäß Lemma 1.17 wird durch

$$\|x\|_B := \|Bx\| \quad \text{für } x \in \mathbb{R}^n$$

dann eine Vektornorm im \mathbb{R}^n definiert. Die Lösung des linearen Gleichungssystems $Ax = b$ ist hiermit offenbar äquivalent zur Minimierung des Residuums

$$\text{minimiere } \frac{1}{2} \|b - Ax\|_B^2, \quad x \in \mathbb{R}^n.$$

Ersetzt man dieses n -dimensionale Problem im k -ten Iterationsschritt durch das nur noch k -dimensionale Problem

$$\text{minimiere } \frac{1}{2} \|b - Ax\|_B^2, \quad x \in x^0 + \mathcal{K}_k(r^0, A),$$

wobei $x^0 \in \mathbb{R}^n$ ein beliebiger Startvektor ist,

$$r^0 := b - Ax^0$$

das zugehörige Anfangsresiduum bezeichnet und

$$\mathcal{K}_k(r^0, A) := \text{span}\{r^0, Ar^0, \dots, A^{k-1}r^0\}$$

gilt, so gelangt man zu der folgenden Klasse von Verfahren.

Algorithmus 6.2. (*Allgemeines Minimum-Residuum-Verfahren*)

Wähle $x^0 \in \mathbb{R}^n$, $B \in \mathbb{R}^{n \times n}$ regulär, und setze $r^0 := b - Ax^0$.

FOR $k = 1, 2, \dots$

Bestimme $x^k \in \mathbb{R}^n$ als Lösung von

$$\text{minimiere } \frac{1}{2} \|b - Ax\|_B^2 \quad \text{für } x \in x^0 + \mathcal{K}_k(r^0, A). \quad (6.1)$$

END

Der Algorithmus 6.2 besitzt in der Wahl der regulären Matrix B natürlich gewisse Freiheiten, auf die wir noch zurückkommen werden. Weiterhin wird zu klären sein, wie man die hier auftretenden k -dimensionalen Teilprobleme (6.1) möglichst geschickt lösen kann. Ansonsten lässt sich für die obige Klasse von Verfahren bereits ein allgemeiner Konvergenzsatz beweisen. Zu diesem Zweck notieren wir zunächst ein Hilfsresultat, welches man als unmittelbare Konsequenz des Satzes 6.1 erhält.

Lemma 6.3. *Seien $A \in \mathbb{R}^{n \times n}$ eine reguläre Matrix und $b \in \mathbb{R}^n$ gegeben. Seien ferner $\mathcal{K}_k := \mathcal{K}_k(r^0, A)$ für $r^0 := b - Ax^0$ und ein beliebiges $x^0 \in \mathbb{R}^n$. Ist $k \in \mathbb{N}$ der kleinste Index mit*

$$\mathcal{K}_0 \subsetneq \mathcal{K}_1 \subsetneq \dots \subsetneq \mathcal{K}_k = \mathcal{K}_{k+1} \quad (\text{hierbei ist } \mathcal{K}_0 := \{0\}),$$

so liegt die Lösung x^* des linearen Gleichungssystems $Ax = b$ bereits in dem affinen Raum $x^0 + \mathcal{K}_k$.

Beweis. Wegen Satz 6.1 und der Voraussetzung $\mathcal{K}_k = \mathcal{K}_{k+1}$ gehört der Lösungsvektor y^* des linearen Gleichungssystems $Ay = r^0$ zum Krylov-Raum \mathcal{K}_k . Wegen $r^0 = b - Ax^0$ ist dies äquivalent zu $A(x^0 + y) = b$. Also ist $x^* := x^0 + y^*$ die Lösung von $Ax = b$, und wegen $y^* \in \mathcal{K}_k$ folgt hieraus $x^* \in x^0 + \mathcal{K}_k$. \square

Da nach spätestens n Schritten $\mathcal{K}_n = \mathcal{K}_{n+1}$ gelten muss (denn im \mathbb{R}^n kann es keine $n + 1$ linear unabhängigen Vektoren geben), erhalten wir den folgenden Satz als direkte Folgerung des Lemmas 6.3.

Satz 6.4. *Sei $A \in \mathbb{R}^{n \times n}$ eine reguläre Matrix. Dann findet der Algorithmus 6.2 nach spätestens n Schritten die Lösung x^* des linearen Gleichungssystems $Ax = b$.*

Der Satz 6.4 garantiert insbesondere, dass der Algorithmus 6.2 nach endlich vielen Schritten abbricht. Die Zahl n dient hierbei als obere Schranke und ist für viele praktische Probleme zu pessimistisch, zumal die iterativen Verfahren gerade bei linearen Gleichungssystemen von sehr hoher Dimension (also für sehr großes n) eingesetzt werden. Tatsächlich werden die in den folgenden Abschnitten zu besprechenden Verfahren oft schon nach sehr viel weniger

Iterationen eine recht brauchbare Näherung für die gesuchte Lösung x^* des linearen Gleichungssystems $Ax = b$ finden.

Neben dem allgemeinen Minimum-Residuum-Verfahren aus dem Algorithmus 6.2 gibt es noch eine weitere wichtige Klasse von Verfahren zur Lösung eines linearen Gleichungssystems $Ax = b$, nämlich die so genannten *Galerkin-Verfahren*, manchmal auch *Projektionsmethoden* genannt. Diese bestimmen die k -te Iterierte ebenfalls aus dem Raum $x^0 + \mathcal{K}_k(r^0, A)$, wobei $x^0 \in \mathbb{R}^n$ wieder ein beliebiger Startvektor ist und $r^0 := b - Ax^0$ das zugehörige Anfangsresiduum bezeichnet. Allerdings wird die k -te Iterierte x^k jetzt nicht mehr als das Minimum des Residuums $b - Ax$ (bzgl. einer geeigneten Norm) berechnet, sondern dieses Residuum soll nun orthogonal zu einem Raum \mathcal{L}_k mit $\dim(\mathcal{L}_k) = k$ sein. Ein Galerkin-Verfahren ist also bestimmt durch die beiden Forderungen

$$x^k \in x^0 + \mathcal{K}_k(r^0, A) \quad \text{und} \quad b - Ax^k \perp \mathcal{L}_k.$$

Speziell für $\mathcal{L}_k = \mathcal{K}_k$ spricht man von einem *Ritz-Galerkin-* oder *Bubnov-Galerkin-Verfahren*, für $\mathcal{L}_k \neq \mathcal{K}_k$ hingegen von einem *Petrov-Galerkin-Verfahren*. Formal sieht ein Galerkin-Verfahren also wie folgt aus.

Algorithmus 6.5. (*Allgemeines Galerkin-Verfahren*)

Wähle $x^0 \in \mathbb{R}^n$, und setze $r^0 := b - Ax^0$.

FOR $k = 1, 2, \dots$

Bestimme $x^k \in \mathbb{R}^n$ mit

$$x^k \in x^0 + \mathcal{K}_k(r^0, A) \quad \text{und} \quad b - Ax^k \perp \mathcal{L}_k \quad (6.2)$$

für einen Vektorraum \mathcal{L}_k mit $\dim(\mathcal{L}_k) = k$.

END

Der Algorithmus 6.5 bricht ebenfalls nach spätestens n Schritten mit der Lösung des linearen Gleichungssystems $Ax = b$ ab, denn der Nullvektor ist offenbar der einzige Vektor, der zu einem n -dimensionalen Vektorraum \mathcal{L}_n orthogonal ist. Auf die Frage, wie sich ein x^k mit den Eigenschaften (6.2) berechnen lässt (und ob ein solches x^k überhaupt existiert) werden wir erst später eingehen, d.h., wir kümmern uns an dieser Stelle nicht weiter um die Realisierung des Algorithmus 6.5.

Stattdessen zeigen wir in unserem folgenden Beispiel, dass das bereits bekannte CG-Verfahren sowohl zur Klasse der allgemeinen Minimum-Residuen-Verfahren als auch zur Klasse der Galerkin-Verfahren gehört.

Beispiel 6.6. (a) Wegen Lemma 5.7 ist die Iterierte x^k des CG-Verfahrens eine Lösung des Optimierungsproblems

$$\text{minimiere } \|b - Ax\|_{A^{-1/2}} \quad \text{für } x \in x^0 + \mathcal{K}_k(r^0, A).$$

Also ist das CG-Verfahren ein Spezialfall des allgemeinen Minimum-Residuen-Verfahrens aus dem Algorithmus 6.2 mit der Matrix $B := A^{-1/2}$.

(b) Wegen $r^k = -g^k$ beim CG-Verfahren gilt aufgrund des Satzes 5.4 die Orthogonalitätsrelation

$$(r^k)^T d^j = 0 \quad \forall j = 0, 1, \dots, k-1.$$

Nach Lemma 5.6 ist aber $\mathcal{K}_k = \text{span}\{d^0, d^1, \dots, d^{k-1}\}$, woraus sich

$$b - Ax^k = r^k \perp \mathcal{K}_k$$

ergibt. Das CG-Verfahren ist demzufolge auch ein Ritz-Galerkin-Verfahren.

6.2 GMRES für reguläre Systeme

Seien $A \in \mathbb{R}^{n \times n}$ und $b \in \mathbb{R}^n$ gegeben. Gesucht ist eine Lösung des linearen Gleichungssystems

$$Ax = b, \tag{6.3}$$

wobei wir außer der Regularität keine weiteren Voraussetzungen an die Matrix A stellen wollen; insbesondere braucht A weder symmetrisch noch positiv definit zu sein. Das GMRES-Verfahren (GMRES=Generalized Minimal Residual) von Saad und Schultz [54] gehört dabei in die allgemeine Klasse von Verfahren aus dem Algorithmus 6.2, wobei wir hier mangels besserer Ideen einfach $B := I$ setzen. Die Grundversion des GMRES-Verfahrens lautet damit wie folgt:

Wähle $x^0 \in \mathbb{R}^n$ und setze $r^0 := b - Ax^0$.

FOR $k = 1, 2, \dots$

Bestimme $x^k \in \mathbb{R}^n$ als Lösung von

$$\text{minimiere } \frac{1}{2} \|b - Ax\|^2 \text{ für } x \in x^0 + \mathcal{K}_k.$$

END

Hierbei bezeichnet

$$\mathcal{K}_k := \mathcal{K}_k(r^0, A) := \text{span}\{r^0, Ar^0, A^2r^0, \dots, A^{k-1}r^0\}$$

natürlich wieder den zur Matrix A und zum Anfangsresiduum r^0 zugehörigen k -ten Krylov-Raum.

Damit ist das GMRES-Verfahren im Prinzip vollständig beschrieben. Praktisch stellt sich jedoch die Frage, wie man das in jeder Iteration auftretende Optimierungsproblem möglichst einfach lösen kann. Wir werden uns im Folgenden mit dieser Frage näher auseinandersetzen.

Zunächst einmal wird es sich als hilfreich erweisen, eine Orthonormalbasis des Krylov-Raums \mathcal{K}_k zu konstruieren. Dies kann etwa durch Anwendung des Gram-Schmidt-Verfahrens geschehen. Die genaue Vorgehensweise ist dabei wie folgt: Sind $\{v^1, \dots, v^j\}$ bereits Orthonormalbasen der Krylov-Räume \mathcal{K}_j

für $j < k$, so spannen die Vektoren $\{v^1, \dots, v^j, Av^j\}$ offenbar den Krylov-Raum \mathcal{K}_{j+1} auf. Wenden wir nun das Gram-Schmidt-Verfahren aus dem Algorithmus 3.8 an, um den neuen Vektor Av^j gegenüber den schon vorher berechneten Vektoren v^1, \dots, v^j zu orthonormalisieren, so ergibt sich gerade das nachstehende Verfahren, vergleiche Aufgabe 6.1.

Algorithmus 6.7. (*Arnoldi-Verfahren*)

Wähle $x^0 \in \mathbb{R}^n$.

Setze $r^0 := b - Ax^0$, $v^1 := \frac{r^0}{\|r^0\|}$.

FOR $j = 1 : k$

$w^j := Av^j$;

FOR $i = 1 : j$

$h_{ij} := (v^i)^T(Av^j)$;

$w^j := w^j - h_{ij}v^i$;

END

$h_{j+1,j} := \|w^j\|$;

$v^{j+1} := w^j/h_{j+1,j}$;

END

Numerisch ist es natürlich wieder besser, wenn man das modifizierte Gram-Schmidt-Verfahren aus dem Algorithmus 3.10 zur Orthonormalisierung des neuen Vektors Av^j anwendet. Dies ergibt das folgende Verfahren, welches aufgrund unserer Ausführungen im Abschnitt 3.3 mathematisch völlig äquivalent zum Algorithmus 6.7 ist.

Algorithmus 6.8. (*Modifiziertes Arnoldi-Verfahren*)

Wähle $x^0 \in \mathbb{R}^n$.

Setze $r^0 := b - Ax^0$, $v^1 := \frac{r^0}{\|r^0\|}$.

FOR $j = 1 : k$

$w^j := Av^j$;

FOR $i = 1 : j$

$h_{ij} := (v^i)^T w^j$;

$w^j := w^j - h_{ij}v^i$;

END

$h_{j+1,j} := \|w^j\|$;

$v^{j+1} := w^j/h_{j+1,j}$;

END

Die Erörterungen aus dem Abschnitt 3.3 zeigen, dass die beiden Arnoldi-Verfahren eine Orthonormalbasis von \mathcal{K}_k konstruieren, sofern die Vektoren $r^0, Ar^0, \dots, A^{k-1}r^0$ linear unabhängig sind. Sollte dies nicht der Fall sein, so wissen wir aufgrund des Lemmas 6.3, dass wir bereits eine Lösung des linearen Gleichungssystems $Ax = b$ gefunden haben, was spätestens nach n Schritten der Fall sein wird. Insbesondere sind die Arnoldi-Verfahren daher wohldefiniert, da stets $h_{j+1,j} \neq 0$ gilt, sofern man noch keine Lösung des zu Grunde liegenden linearen Gleichungssystems gefunden hat. Zusammenfassend haben wir also das folgende Resultat.

Lemma 6.9. *Sind die den Raum \mathcal{K}_k erzeugenden Vektoren $r^0, Ar^0, \dots, A^{k-1}r^0$ linear unabhängig, so bilden die durch den Arnoldi-Algorithmus 6.7 bzw. den hierzu äquivalenten modifizierten Arnoldi-Algorithmus 6.8 konstruierten Vektoren v^1, \dots, v^k eine Orthonormalbasis von \mathcal{K}_k .*

Während wir für numerische Zwecke meist auf das modifizierte Arnoldi-Verfahren aus dem Algorithmus 6.8 zurückgreifen werden, benutzen wir für unsere nachfolgenden theoretischen Betrachtungen den eigentlichen Arnoldi-Algorithmus 6.7. Wir gehen dabei davon aus, dass der Arnoldi-Algorithmus nicht vorzeitig abbricht, also stets $h_{j+1,j} \neq 0$ für alle betrachteten j gilt (sonst, wie gesagt, sind wir schon fertig). Weiterhin bezeichnen wir für unsere nachstehenden Untersuchungen mit $V_k \in \mathbb{R}^{n \times k}$ die durch

$$V_k := \begin{pmatrix} | & & | \\ v^1 & \cdots & v^k \\ | & & | \end{pmatrix} \quad (6.4)$$

definierte Matrix, wobei die v^j natürlich die im Arnoldi-Verfahren 6.7 berechneten Vektoren seien. Außerdem wollen wir an dieser Stelle explizit festhalten, dass die Definition des Vektors w^j im Arnoldi-Verfahren 6.7 offenbar die Gleichheit

$$w^j = Av^j - \sum_{i=1}^j h_{ij}v^i \quad (6.5)$$

impliziert, die in den nachstehenden Untersuchungen einige Male benutzt wird.

Lemma 6.10. *Die durch*

$$H_k := V_k^T AV_k \in \mathbb{R}^{k \times k} \quad (6.6)$$

definierte Matrix ist eine obere Hessenberg-Matrix mit Einträgen

$$(H_k)_{ij} = \begin{cases} h_{ij} & \text{für } i \leq j+1, \\ 0 & \text{für } i > j+1, \end{cases}$$

wobei die h_{ij} die im Arnoldi-Verfahren 6.7 berechneten Elemente sind.

Beweis. Die Elemente von H_k mögen vorübergehend mit \tilde{h}_{ij} bezeichnet werden. Dann folgt aus der Definition (6.6) von H_k und den Vorschriften im Algorithmus 6.7 unmittelbar

$$\tilde{h}_{ij} = (v^i)^T Av^j = h_{ij} \text{ für alle } i \leq j,$$

womit die Aussage des Lemmas zumindest schon mal für das obere Dreieck von H_k nachgewiesen ist. Wir betrachten jetzt die unterhalb der Diagonalen stehenden Einträge von H_k . Für $j = 1, 2, \dots$ und $l = 1, 2, \dots$ ergibt aus (6.6) und dem Algorithmus 6.7 weiterhin

$$\begin{aligned}
\tilde{h}_{j+l,j} &= (v^{j+l})^T A v^j \\
&\stackrel{(6.5)}{=} (v^{j+l})^T \left(w^j + \sum_{i=1}^j h_{ij} v^i \right) \\
&= (v^{j+l})^T w^j + \sum_{i=1}^j h_{ij} \underbrace{(v^{j+l})^T v^i}_{=0} \\
&= h_{j+1,j} (v^{j+l})^T v^{j+1} \\
&= \begin{cases} h_{j+1,j} & \text{für } l = 1, \\ 0 & \text{für } l > 1 \end{cases}
\end{aligned}$$

aufgrund der Orthogonalität der Vektoren v^1, v^2, \dots, v^k aus dem Arnoldi-Verfahren 6.7, vergleiche Lemma 6.9. \square

Eine kompakte Schreibweise des Arnoldi-Verfahrens 6.7 wird durch das folgende Resultat geliefert.

Lemma 6.11. *Die durch*

$$\overline{H}_k := \begin{pmatrix} H_k \\ 0 \dots 0 h_{k+1,k} \end{pmatrix} \in \mathbb{R}^{(k+1) \times k} \quad (6.7)$$

definierte Matrix (mit H_k aus (6.6)) genügt der Gleichung

$$A V_k = V_{k+1} \overline{H}_k \quad (6.8)$$

(mit V_k bzw. V_{k+1} gemäß (6.4)).

Beweis. Die Behauptung folgt sofort aus den Vorschriften des Algorithmus 6.7, wonach

$$A v^j \stackrel{(6.5)}{=} w^j + \sum_{i=1}^j h_{ij} v^i = h_{j+1,j} v^{j+1} + \sum_{i=1}^j h_{ij} v^i = \sum_{i=1}^{j+1} h_{ij} v^i$$

für alle $j = 1, \dots, k$ gilt. \square

Wir kehren nun wieder zu dem vom GMRES-Verfahren zu lösenden Optimierungsproblem

$$\min \|b - Ax\| \quad \text{mit } x \in x^0 + \mathcal{K}_k \quad (6.9)$$

zurück und gehen davon aus, dass die Lösung x^* des linearen Gleichungssystems $Ax = b$ im k -ten Iterationsschritt noch nicht gefunden wurde. Da die Spalten der Matrix V_k aus (6.4) aufgrund des Lemmas 6.9 dann eine Orthonormalbasis des Krylov-Raumes \mathcal{K}_k bilden, lässt sich jeder Vektor $x \in x^0 + \mathcal{K}_k$ schreiben in der Gestalt

$$x = x^0 + V_k y$$

für ein $y \in \mathbb{R}^k$. Mit $e_1 := (1, 0, \dots, 0)^T \in \mathbb{R}^{k+1}$ und Lemma 6.11 folgt daher

$$\begin{aligned} \|b - Ax\| &= \|b - A(x^0 + V_k y)\| \\ &= \|r^0 - AV_k y\| \\ &= \|\|r^0\|v^1 - AV_k y\| \\ &= \|\|r^0\|v^1 - V_{k+1}\overline{H}_k y\| \\ &= \|V_{k+1}(\|r^0\|e_1 - \overline{H}_k y)\| \\ &= \|\|r^0\|e_1 - \overline{H}_k y\|, \end{aligned} \tag{6.10}$$

wobei die letzte Gleichung aus $V_{k+1}^T V_{k+1} = I$ folgt und \overline{H}_k natürlich die Matrix aus dem Lemma 6.11 bezeichnet.

Also können wir das Optimierungsproblem (6.9) ersetzen durch das Ausgleichsproblem

$$\min \|\beta e_1 - \overline{H}_k y\|, \quad y \in \mathbb{R}^k, \tag{6.11}$$

mit $\beta := \|r^0\|$. Damit ist die Grundversion des GMRES-Verfahrens äquivalent zu der folgenden Vorgehensweise:

Wähle $x^0 \in \mathbb{R}^n$ und setze $r^0 := b - Ax^0$, $\beta := \|r^0\|$.

FOR $k = 1, 2, \dots$

Berechne die Matrix \overline{H}_k aus (6.7).

Bestimme eine Lösung y^k des Ausgleichsproblems (6.11).

Setze $x^k := x^0 + V_k y^k$.

END

Nach Lemma 6.10 ergibt sich die hierbei zu berechnende Matrix \overline{H}_k unmittelbar durch Anwendung des Arnoldi-Verfahrens 6.7. Benutzen wir stattdessen das hierzu äquivalente modifizierte Arnoldi-Verfahren, so erhalten wir die folgende Spezifizierung des gerade beschriebenen Verfahrens.

Wähle $x^0 \in \mathbb{R}^n$ und setze $r^0 := b - Ax^0$, $\beta := \|r^0\|$, $v^1 := r^0/\beta$.

FOR $k = 1, 2, \dots$

$w^k := Av^k$;

FOR $i = 1 : k$

$h_{ik} := (v^i)^T w^k$;

$w^k := w^k - h_{ik}v^i$;

END

$h_{k+1,k} := \|w^k\|$;

$v^{k+1} := w^k/h_{k+1,k}$;

Bestimme eine Lösung y^k des Ausgleichsproblems (6.11).

Setze $x^k := x^0 + V_k y^k$.

END

Damit bleibt noch zu untersuchen, wie man das Ausgleichsproblem (6.11) lösen kann. Im Prinzip ist dies mit jedem der Verfahren aus dem Kapitel 3

möglich. Aufgrund der oberen Hessenberg-Gestalt von \overline{H}_k liefern die Givens-Rotationen allerdings die effizienteste Möglichkeit. Aus der Gestalt

$$\overline{H}_k = \begin{pmatrix} * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ & * & * & \cdots & * \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & * \\ 0 & & & & * \end{pmatrix}$$

folgt nämlich, dass sich die Matrix \overline{H}_k durch Anwendung von insgesamt k Givens-Rotationen $G_{21}, G_{32}, \dots, G_{k+1,k} \in \mathbb{R}^{(k+1) \times (k+1)}$ auf obere Dreiecksgestalt transformieren lässt:

$$G_{k+1,k} \cdots G_{32} G_{21} \overline{H}_k = \begin{pmatrix} * & * & * & \cdots & * \\ & * & * & \cdots & * \\ & & * & \cdots & * \\ & & & \ddots & \vdots \\ & & & & * \\ & & & & 0 \end{pmatrix} =: \overline{R}_k =: \begin{pmatrix} R_k \\ 0 \end{pmatrix} \quad (6.12)$$

mit einer oberen Dreiecksmatrix $R_k \in \mathbb{R}^{k \times k}$. Hat man diese Berechnungen in der Iteration k durchgeführt, so stellt sich die Situation im nächsten Iterationsschritt $k+1$ sogar noch einfacher dar, denn die Matrix \overline{H}_{k+1} unterscheidet sich von \overline{H}_k im Prinzip nur dadurch, dass eine neue Spalte am Ende hinzugefügt wird:

$$\overline{H}_{k+1} = \left(\begin{array}{c|c} & * \\ & \vdots \\ & \vdots \\ \overline{H}_k & \vdots \\ & \vdots \\ & \vdots \\ & * \\ \hline 0 \cdots 0 & * \end{array} \right) \in \mathbb{R}^{(k+2) \times (k+1)}.$$

Um diese Matrix auf obere Dreiecksgestalt zu bringen, hat man wieder nur die untere Nebendiagonale zu eliminieren. Für den \overline{H}_k -Block haben wir dies aber schon im vorherigen Schritt k getan, so dass die dort berechneten Givens-Rotationen $G_{21}, \dots, G_{k+1,k}$ (formal jetzt als Matrizen im $\mathbb{R}^{(k+2) \times (k+2)}$ aufgefasst) nur noch auf die letzte Spalte von \overline{H}_{k+1} angewendet werden müssen. Ferner benötigt man in der Iteration $k+1$ dann natürlich noch eine neue Givens-Rotation $G_{k+2,k+1}$, um schließlich auch das in der letzten Zeile von \overline{H}_{k+1} stehende Element zu eliminieren. Wendet man diese Givens-Rotationen auch auf die jeweilige rechte Seite des Ausgleichsproblems an, so lässt sich dieses durch einfache Rückwärtssubstitution lösen.

Mit (6.10) erhalten wir auf diese Weise dann

$$\|b - Ax\| = \|\beta e_1 - \overline{H}_k y\| = \|\overline{z}^k - \overline{R}_k y\|,$$

wenn \overline{z}^k den sich durch Anwendung der Givens-Rotationen $G_{21}, \dots, G_{k+1,k}$ auf βe_1 ergebenden Vektor bezeichnet, der sich per Definition dann rekursiv durch

$$\overline{z}^k = G_{k+1,k} \begin{pmatrix} \overline{z}^{k-1} \\ 0 \end{pmatrix}$$

fortschreiben lässt. Partitionieren wir diesen Vektor in der Form

$$\overline{z}^k = \begin{pmatrix} z^k \\ \zeta_k \end{pmatrix} \in \mathbb{R}^{k+1},$$

so folgt mit der Notation aus (6.12) unmittelbar

$$\|b - Ax^k\|^2 = \|\overline{z}^k - \overline{R}_k y^k\|^2 = \|z^k - R_k y^k\|^2 + |\zeta_k|^2 = |\zeta_k|^2,$$

denn y^k ergibt sich gerade durch Rückwärtssubstitution als Lösung des linearen Gleichungssystems $R_k y = z^k$. Also lässt sich das Residuum $\|b - Ax^k\|$ im k -ten Iterationsschritt durch die Größe $|\zeta_k|$ angeben, die auch ohne Kenntnis von x^k vorliegt. Insbesondere braucht das Ausgleichsproblem (6.11) erst dann gelöst zu werden, wenn diese Größe hinreichend klein ist. Daher muss auch der eigentlich interessierende Vektor $x^k := x^0 + V_k y^k$ nicht in jeder Iteration berechnet werden, womit man immerhin eine ganze Reihe von Flops einsparen kann.

Überschreiben wir den Vektor \overline{z}^k in jeder Iteration (wobei dieser pro Iteration um eine Komponente größer wird) und speichern die jeweiligen Givens-Rotationen $G_{k+1,k}$ in den sie definierenden Größen c_k und s_k ab, so ergibt sich aus den obigen Überlegungen zusammenfassend jetzt die endgültige Fassung des GMRES-Verfahrens.

Algorithmus 6.12. (GMRES-Verfahren)

Wähle Startvektor $x^0 \in \mathbb{R}^n$ und Abbruchparameter $\varepsilon \geq 0$.

Setze $r^0 := b - Ax^0$, $\beta := \|r^0\|$, $v^1 := r^0/\beta$ und $z_1 := \beta$.

FOR $k = 1, 2, \dots$

 % Modifizierter Arnoldi-Algorithmus zur Berechnung von \overline{H}_k

$w^k := Av^k$;

 FOR $i = 1 : k$

$h_{ik} := (v^i)^T w^k$;

$w^k := w^k - h_{ik} v^i$;

 END

$h_{k+1,k} := \|w^k\|$;

$v^{k+1} := w^k/h_{k+1,k}$;

 % Anwendung von Givens-Rotationen auf (6.11)

 % Wende alte Givens-Rotationen auf k -te Spalte von \overline{H}_k an

```

FOR i = 1 : (k - 1)
    
$$\begin{pmatrix} h_{ik} \\ h_{i+1,k} \end{pmatrix} := \begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix} \begin{pmatrix} h_{ik} \\ h_{i+1,k} \end{pmatrix};$$

END
% Bestimme neue Givens-Rot. zur Elimination von  $\overline{H}_k(k+1, k)$ 
 $\tau := |h_{kk}| + |h_{k+1,k}|;$ 
 $\nu := \tau \cdot \sqrt{(h_{kk}/\tau)^2 + (h_{k+1,k}/\tau)^2};$ 
 $c_k := h_{kk}/\nu;$ 
 $s_k := h_{k+1,k}/\nu;$ 
% Anwendung der neuen Givens-Rotation auf  $\overline{H}_k$ 
 $h_{kk} := \nu;$ 
 $h_{k+1,k} := 0;$ 
% Anwendung der neuen Givens-Rotation auf rechte Seite  $z^k$ 
 $z_{k+1} := -s_k z_k;$ 
 $z_k := c_k z_k;$ 
% Abbruchkriterium
IF  $|z_{k+1}|/\beta \leq \varepsilon$ : STOP
END
% Rückwärtssubstitution zur Berechnung von  $y^k$ 
 $y_k := z_k/h_{kk};$ 
FOR i = (k - 1) : -1 : 1
     $y_i = (z_i - \sum_{j=i+1}^k h_{ij}y_j)/h_{ii};$ 
END
% Berechnung der Näherungslösung  $x^k := x^0 + V_k y^k$ 
 $x^k := x^0 + \sum_{i=1}^k y_i v^i;$ 

```

Der Algorithmus 6.12 kann in der obigen Form sofort implementiert werden. Als Abbruchkriterium haben wir (wie schon früher) das relative Residuum als Genauigkeit genommen, alternativ könnte man auch die Abfrage $|z_{k+1}| \leq \varepsilon$ wählen, die den absoluten Fehler im aktuellen Residuum misst. Die im obigen Algorithmus benutzte Vorschrift $h_{k+1,k} := 0$ kann im Prinzip auch entfallen, da das untere Dreieck der transformierten Matrix \overline{H}_k letztlich nicht mehr von Interesse ist.

Als Testbeispiel betrachten wir jetzt die zweidimensionale diskretisierte Konvektions–Diffusions–Gleichung $C_{N \times N} v = h^2 f$ aus dem Abschnitt 1.9 mit $N = 32$ (also $n = N^2 = 1.024$), dem Vektor

$$b = \left(\cos \frac{\pi}{4}, \sin \frac{\pi}{4} \right)^T$$

sowie der rechten Seite

$$f := (1, \dots, 1)^T.$$

Als Startvektor x^0 dient wieder der Nullvektor, und der Abbruchparameter ε wurde auch hier als 10^{-6} gewählt. Das GMRES–Verfahren benötigt dann 80 Iterationen zur Lösung dieses Problems. Der zugehörige Verlauf des relativen Residuums ist in der Abbildung 6.1 angegeben.

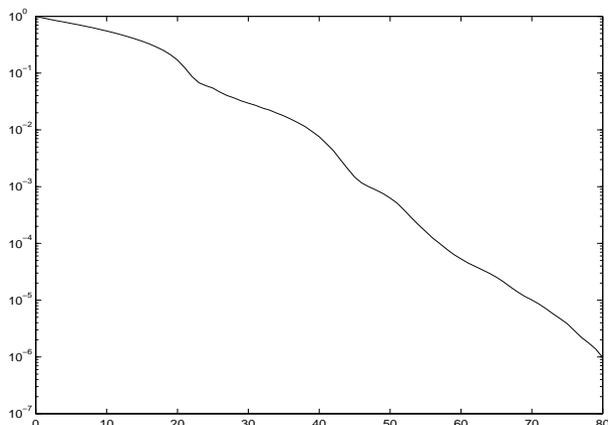


Abbildung 6.1. Verlauf des relativen Residuums beim GMRES-Verfahren

Wir beschließen diesen Abschnitt mit einigen Bemerkungen.

Bemerkung 6.13. (a) Bei der Konstruktion einer Orthonormalbasis des Krylov-Raumes \mathcal{K}_k benutzen wir hier das Arnoldi-Verfahren. Von Walker [65] stammt ein Vorschlag, bei dem man alternativ Householder-Transformationen benutzen kann. Dieser Zugang ist zwar etwas aufwendiger im Hinblick auf die benötigten Flops, gilt in der Praxis aber als numerisch stabiler und wird deshalb manchmal bevorzugt. Wir gehen auf diese Variante von GMRES auch im Rahmen der Aufgabe 6.7 ein.

(b) Aus dem Arnoldi-Verfahren erhalten wir eine Matrix $V_k = (v^1, \dots, v^k) \in \mathbb{R}^{n \times k}$, deren Spaltenvektoren eine Basis des Krylov-Raumes \mathcal{K}_k bilden. Macht man nun einen Ritz-Galerkin-Ansatz

$$x^k \in x^0 + \mathcal{K}_k \quad \text{und} \quad b - Ax^k \perp \mathcal{K}_k$$

und schreibt $x^k = x^0 + V_k y^k$ mit einem $y^k \in \mathbb{R}^k$, so lässt sich dies äquivalent schreiben als

$$0 = V_k^T (b - Ax^k) = V_k^T (r^0 - AV_k y^k) = \|r^0\| e_1 - V_k^T AV_k y^k = \|r^0\| e_1 - H_k y^k.$$

Hieraus erhält man y^k als Lösung eines linearen Gleichungssystems und damit schließlich auch x^k . Diese Vorgehensweise wird als *FOM-Verfahren* bezeichnet (FOM = Full Orthogonalization Method). Für weitere Details verweisen wir hierzu auf [53, 44, 62].

6.3 GMRES(m) für reguläre Systeme

Aufgrund der Herleitung ist klar, dass durch das GMRES-Verfahren nach spätestens n Schritten die exakte Lösung des linearen Gleichungssystems $Ax = b$ gefunden wird. Für großes n bedeutet dies aber einen viel zu großen Aufwand, da beispielsweise die Matrix V_k explizit abgespeichert werden muss. Da es sich hierbei im Allgemeinen um eine vollbesetzte Matrix handelt (selbst bei schwach besetztem A), ist dies für größere Werte von k viel zu teuer.

Um diesen Nachteil zu vermeiden, ersetzt man das GMRES-Verfahren durch ein so genanntes GMRES-Verfahren mit Restarts, auch GMRES(m)-Verfahren genannt. Dabei ist m eine vorgegebene natürliche Zahl, die die maximal erlaubte Anzahl an Spalten in der Matrix V_k begrenzt. Sobald also $k = m$ ist, berechnet man die zugehörige aktuelle iterierte x^m analog zum Vorgehen beim GMRES-Verfahren aus dem Algorithmus 6.12. Genügt dieses x^m noch nicht der vorgegebenen Genauigkeit, so startet man das GMRES-Verfahren erneut, allerdings mit dem aktuellen Punkt x^m als Startvektor. Gibt man sich noch eine maximale Zahl von Restarts vor, so erhält man auf diese Weise das nachstehende Verfahren. (Man beachte hierbei, dass das ursprüngliche Anfangsresiduum gesondert abgespeichert werden muss, da x^0 etc. mit jedem Restart überschrieben werden.)

Algorithmus 6.14. (GMRES(m)-Verfahren)

Wähle Startvektor $x^0 \in \mathbb{R}^n$, $\varepsilon \geq 0$, $m \in \mathbb{N}$ und $restart_{\max} \in \mathbb{N}$.

Setze $restart := 0$.

WHILE $restart < restart_{\max}$

Setze $r^0 := b - Ax^0$, $\beta := \|r^0\|$, $v^1 := r^0/\beta$ und $z_1 := \beta$.

IF $restart=0$

$\beta_0 := \beta$;

END

FOR $k = 1 : m$

% Modifizierter Arnoldi-Alg. zur Berechnung von \overline{H}_k

$w^k := Av^k$;

FOR $i = 1 : k$

$h_{ik} := (v^i)^T w^k$;

$w^k := w^k - h_{ik}v^i$;

END

$h_{k+1,k} := \|w^k\|$;

$v^{k+1} := w^k/h_{k+1,k}$;

% Anwendung von Givens-Rotationen auf (6.11)

% Wende alte Givens-Rot. auf k -te Spalte von \overline{H}_k an

FOR $i = 1 : (k-1)$

$\begin{pmatrix} h_{ik} \\ h_{i+1,k} \end{pmatrix} := \begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix} \begin{pmatrix} h_{ik} \\ h_{i+1,k} \end{pmatrix}$;

END

% Neue Givens-Rotation zur Elimination von $\overline{H}_k(k+1, k)$

```

    τ := |hkk| + |hk+1,k|;
    ν := τ · √((hkk/τ)2 + (hk+1,k/τ)2);
    ck := hkk/ν;
    sk := hk+1,k/ν;
    % Anwendung der neuen Givens-Rotation auf  $\overline{H}_k$ 
    hkk := ν;
    hk+1,k := 0;
    % Anwendung der neuen Givens-Rot. auf rechte Seite zk
    zk+1 := -skzk;
    zk := ckzk;
    % Abbruchkriterium
    IF |zk+1|/β ≤ ε: STOP
END
% Rückwärtssubstitution zur Berechnung von yk
yk := zk/hkk;
FOR i = (k - 1) : -1 : 1
    yi = (zi - ∑j=i+1k hijyj)/hii;
END
% Berechnung der Näherungslösung xk := x0 + Vkyk
xk := x0 + ∑i=1k yivi;
% Überprüfung auf Restart
IF |zk+1|/β0 > ε
    x0 := xk;
    restart := restart + 1;
END
END

```

Unter Verwendung des GMRES(m)-Verfahrens kann man zwar den maximal benötigten Speicherplatz kontrollieren, allerdings besitzt dieses Verfahren im Allgemeinen nicht mehr die Eigenschaft des eigentlichen GMRES-Verfahrens, nach spätestens n Schritten mit einer Lösung abzubrechen. Es ist noch nicht einmal garantiert, dass die durch das GMRES(m)-Verfahren erzeugte Folge gegen die Lösung x^* des linearen Gleichungssystems $Ax = b$ konvergiert. Für den Spezialfall einer positiv definiten (nicht notwendig symmetrischen) Matrix geben wir allerdings ein Konvergenzresultat an. Zu diesem Zweck beginnen wir zunächst mit einem Resultat für das GMRES-Verfahren.

Lemma 6.15. *Seien $A \in \mathbb{R}^{n \times n}$ eine positiv definite (nicht notwendig symmetrische) Matrix und x^k die durch das GMRES-Verfahren aus dem Algorithmus 6.12 erzeugte Iterierte. Dann ist*

$$\|r^k\| \leq \left(1 - \frac{\mu^2}{\sigma^2}\right)^{1/2} \|r^0\|$$

mit den Residuen

$$r^k := b - Ax^k, \quad r^0 := b - Ax^0$$

sowie den vom Iterationsindex k unabhängigen Konstanten

$$\mu := \lambda_{\min}\left(\frac{1}{2}(A + A^T)\right), \quad \sigma := \|A\|_2.$$

Beweis. Wegen $r^0 \in \mathcal{K}_k(r^0, A)$ ist

$$x^0 + \alpha r^0 \in x^0 + \mathcal{K}_k(r^0, A) \quad \forall \alpha \in \mathbb{R}.$$

Da x^k gemäß Definition des GMRES-Verfahrens gerade das Residuum auf dem Raum $x^0 + \mathcal{K}_k(r^0, A)$ (in der euklidischen Norm) minimiert, folgt somit

$$\begin{aligned} \|r^k\|^2 &\leq \|b - A(x^0 + \alpha r^0)\|^2 \\ &= \|r^0 - \alpha Ar^0\|^2 \\ &= \|r^0\|^2 - 2\alpha(r^0)^T Ar^0 + \alpha^2 \|Ar^0\|^2 \end{aligned}$$

für alle $\alpha \in \mathbb{R}$. Speziell für

$$\alpha := \frac{(r^0)^T Ar^0}{\|Ar^0\|^2}$$

erhält man daher

$$\begin{aligned} \|r^k\|^2 &\leq \|r^0\|^2 - \left(\frac{(r^0)^T Ar^0}{\|Ar^0\|}\right)^2 \\ &= \|r^0\|^2 \left(1 - \underbrace{\left(\frac{(r^0)^T Ar^0}{\|r^0\|^2}\right)^2}_{\geq \mu} \underbrace{\left(\frac{\|r^0\|}{\|Ar^0\|}\right)^2}_{\geq 1/\sigma}\right) \\ &\leq \|r^0\|^2 \left(1 - \frac{\mu^2}{\sigma^2}\right) \end{aligned}$$

wegen

$$(r^0)^T Ar^0 = (r^0)^T \left(\frac{1}{2}(A + A^T)\right) r^0 \stackrel{\text{Kor. 1.11}}{\geq} \mu \|r^0\|^2$$

und

$$\sigma = \|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \geq \frac{\|Ar^0\|}{\|r^0\|}.$$

Hieraus folgt die Behauptung. \square

Unter Verwendung des Lemmas 6.15 können wir jetzt auch einen Konvergenzsatz für das GMRES(m)-Verfahren beweisen.

Satz 6.16. *Sei $A \in \mathbb{R}^{n \times n}$ positiv definit (nicht notwendig symmetrisch). Dann konvergiert das GMRES(m)-Verfahren für jedes $m \geq 1$.*

Beweis. Aufgrund des Lemmas 6.15 gilt zunächst

$$\|r^m\| \leq \left(1 - \frac{\mu^2}{\sigma^2}\right)^{1/2} \|r^0\|$$

mit gewissen Konstanten μ und σ . Nach m Schritten wird das GMRES(m)-Verfahren dann neu gestartet mit dem Anfangsresiduum r^m . Daher ergibt sich aus dem Lemma 6.15 auch

$$\|r^{2m}\| \leq \left(1 - \frac{\mu^2}{\sigma^2}\right)^{1/2} \|r^m\|.$$

So fortfahrend, erhält man die Gültigkeit der Ungleichungen

$$\|r^{jm}\| \leq \left(1 - \frac{\mu^2}{\sigma^2}\right)^{1/2} \|r^{(j-1)m}\|$$

für $j = 1, 2, \dots$. Hieraus folgt

$$\|r^{jm}\| \leq \left(1 - \frac{\mu^2}{\sigma^2}\right)^{j/2} \|r^0\|.$$

Wegen

$$\left(1 - \frac{\mu^2}{\sigma^2}\right)^{j/2} \rightarrow 0 \quad \text{für } j \rightarrow \infty$$

ergibt sich die Behauptung. □

In der Praxis wird das GMRES(m)-Verfahren dem eigentlichen GMRES-Verfahren meist vorgezogen, da Rechenaufwand und Speicherplatz-Bedarf beim GMRES-Verfahren einfach zu groß sind. Wir testen das GMRES(m)-Verfahren mit $m = 20$ ebenfalls an dem Beispiel der zweidimensionalen diskretisierten Konvektions-Diffusions-Gleichung $C_{N \times N} v = h^2 f$ mit $N = 32$ (also $n = N^2 = 1.024$) sowie

$$b = \left(\cos \frac{\pi}{4}, \sin \frac{\pi}{4}\right)^T \quad \text{und} \quad f := (1, \dots, 1)^T.$$

Als Startvektor x^0 dient wieder der Nullvektor, und der Abbruchparameter ε wurde erneut als 10^{-6} gewählt. Das GMRES(m)-Verfahren benötigt insgesamt 178 Iterationen, wobei die einzelnen Iterationen (für $k > 20$) natürlich wesentlich günstiger sind als beim GMRES-Verfahren. Der Verlauf des relativen Residuums für das GMRES(m)-Verfahren befindet sich in der Abbildung 6.2.

Bemerkung 6.17. (a) Einige der Hinweise aus der Bemerkung 6.13 gelten natürlich auch für das GMRES(m)-Verfahren. So kann man den Arnoldi-Teil

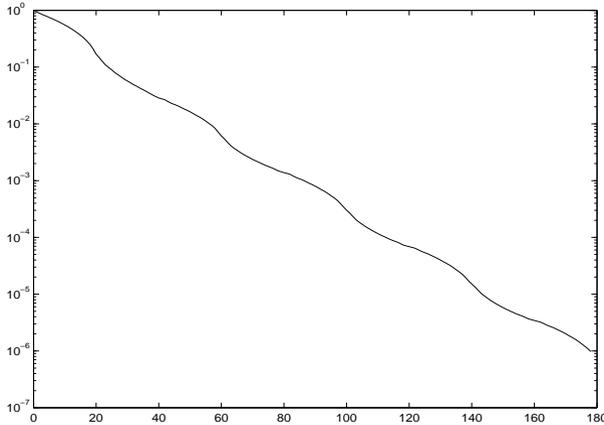


Abbildung 6.2. Verlauf des relativen Residuums beim GMRES(m)–Verfahren

beispielsweise wieder durch geeignete Householder–Transformationen ersetzen.

(b) Das GMRES(m)–Verfahren kontrolliert den Rechenaufwand und Speicherplatzbedarf von GMRES, indem nach jeweils m Schritten ein völliger Neustart durchgeführt wird. Alternativ kann man im Arnoldi–Verfahren auch nur auf die letzten m Vektoren $v^k, v^{k-1}, \dots, v^{k-m+1}$ zurückgreifen, womit man ebenfalls den Rechen– und Speicheraufwand beschränken kann. Für weitere Einzelheiten zu dieser Modifikation sei auf [53] verwiesen.

(c) Eine Reihe von weiteren Varianten des GMRES–Verfahrens werden beispielsweise in den beiden Büchern [53] von Saad und [62] von van der Vorst besprochen.

6.4 MINRES für symmetrische Systeme

Betrachte wieder das lineare Gleichungssystem $Ax = b$ mit $b \in \mathbb{R}^n$ und einer regulären Matrix $A \in \mathbb{R}^{n \times n}$, die jetzt zusätzlich als symmetrisch vorausgesetzt wird. Die Vorgehensweise zur Herleitung des *MINRES–Verfahrens* (MINRES=MINimal RESidual) von Paige und Saunders [47] kann zunächst analog zum GMRES–Verfahren erfolgen. Dazu wählt man wieder $B = I$ in dem Algorithmus 6.2, womit sich die folgenden Vorschriften als Grundversion des MINRES–Verfahrens ergeben:

Wähle $x^0 \in \mathbb{R}^n$ und setze $r^0 := b - Ax^0$.

FOR $k = 1, 2, \dots$

Bestimme $x^k \in \mathbb{R}^n$ als Lösung von

$$\text{minimiere } \frac{1}{2} \|b - Ax\|^2 \text{ für } x \in x^0 + \mathcal{K}_k.$$

END

Man beachte hierbei, dass diese Grundversion des MINRES-Verfahrens völlig übereinstimmt mit derjenigen des GMRES-Verfahrens. Unterschiede treten erst bei der Lösung des Minimierungsproblems auf. Aufgrund der jetzt vorliegenden Symmetrie der Matrix A ergeben sich beim MINRES-Verfahren dabei erhebliche Vereinfachungen gegenüber dem GMRES-Verfahren, die das MINRES-Verfahren wesentlich kostengünstiger machen als das GMRES-Verfahren zur Lösung unsymmetrischer Gleichungssysteme.

Zwecks Berechnung von x^k bestimmen wir zunächst wieder eine Orthonormalbasis des Krylov-Raumes

$$\mathcal{K}_k := \mathcal{K}_k(r^0, A) := \text{span}\{r^0, Ar^0, \dots, A^{k-1}r^0\}.$$

Dazu kann erneut der Arnoldi-Algorithmus 6.7 (bzw. seine Variante aus dem Algorithmus 6.8) benutzt werden. Wegen der Symmetrie von A folgt mit den in (6.6) definierten Matrizen H_k dann

$$H_k = V_k^T A V_k = V_k^T A^T V_k = (V_k^T A V_k)^T = H_k^T.$$

Die obere Hessenberg-Matrix H_k aus dem Lemma 6.10 ist also symmetrisch und somit notwendigerweise von Tridiagonalgestalt. Betrachten wir hier aus numerischen Gründen nur den modifizierten Arnoldi-Algorithmus 6.8, so reduziert sich die innere Schleife aufgrund der obigen Beobachtung auf die Vorschriften

$$\begin{aligned} h_{j-1,j} &:= (v^{j-1})^T w^j, \\ w^j &:= w^j - h_{j-1,j} v^{j-1}, \\ h_{jj} &:= (v^j)^T w^j, \\ w^j &:= w^j - h_{jj} v^j. \end{aligned}$$

Andererseits impliziert die Symmetrie von H_k auch

$$h_{j-1,j} = h_{j,j-1} = \|w^{j-1}\|,$$

wobei die zweite Gleichung aus den Vorschriften des modifizierten Arnoldi-Verfahrens 6.8 folgt. Mit der Notation

$$\alpha_j := h_{jj} \quad \text{und} \quad \beta_j := h_{j,j-1} \quad (= h_{j-1,j} = \|w^{j-1}\|)$$

ergibt sich aus dem modifizierten Arnoldi-Verfahren 6.8 der nachstehende Algorithmus zur Bestimmung einer Orthonormalbasis des Krylov-Raums \mathcal{K}_k im Falle einer symmetrischen Matrix, der in der Literatur üblicherweise als Lanczos-Verfahren bezeichnet wird.

Algorithmus 6.18. (Lanczos-Verfahren)

Wähle $x^0 \in \mathbb{R}^n$.

Setze $r^0 := b - Ax^0$, $v^0 := 0$, $v^1 := r^0 / \|r^0\|$, $\beta_1 := 0$.

FOR $j = 1 : k$

$$w^j := Av^j - \beta_j v^{j-1};$$

$$\alpha_j := (w^j)^T v^j;$$

$$w^j := w^j - \alpha_j v^j;$$

$$\beta_{j+1} := \|w^j\|;$$

$$v^{j+1} := w^j / \beta_{j+1};$$

END

Passen wir noch die Bezeichnungen aus dem letzten Abschnitt an und schreiben

$$T_k := H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & 0 \\ \beta_2 & \alpha_2 & & \\ & & \ddots & \\ 0 & & & \beta_k & \alpha_k \end{pmatrix} \in \mathbb{R}^{k \times k},$$

$$\bar{T}_k := \bar{H}_k = \begin{pmatrix} \alpha_1 & \beta_2 & & 0 \\ \beta_2 & \alpha_2 & & \\ & & \ddots & \\ & & & \beta_k & \alpha_k \\ 0 & & & & \beta_{k+1} \end{pmatrix} \in \mathbb{R}^{(k+1) \times k} \quad (6.13)$$

sowie

$$V_k := \begin{pmatrix} | & & | \\ v^1 & \dots & v^k \\ | & & | \end{pmatrix} \in \mathbb{R}^{n \times k},$$

so gilt aufgrund des Lemmas 6.11 die Beziehung

$$AV_k = V_{k+1} \bar{T}_k. \quad (6.14)$$

Da die Spalten von V_k wieder eine Orthonormalbasis des Krylov-Raumes \mathcal{K}_k bilden, lässt sich jeder Vektor $x \in x^0 + \mathcal{K}_k$ schreiben als $x = x^0 + V_k y$ für ein $y \in \mathbb{R}^k$. Mit $e_1 := (1, 0, \dots, 0)^T \in \mathbb{R}^{k+1}$ und (6.14) folgt dann (analog zur Vorgehensweise in (6.10))

$$\begin{aligned} \|b - Ax\| &= \|b - A(x^0 + V_k y)\| \\ &= \|r^0 - AV_k y\| \\ &= \|\|r^0\|v^1 - AV_k y\| \\ &= \|\|r^0\|v^1 - V_{k+1} \bar{T}_k y\| \\ &= \|V_{k+1} (\|r^0\|e_1 - \bar{T}_k y)\| \\ &= \|\|r^0\|e_1 - \bar{T}_k y\|, \end{aligned}$$

wobei die letzte Gleichheit wieder aus der Orthogonalität $V_{k+1}^T V_{k+1} = I$ folgt.

Also können wir die Lösung des in der Grundversion des MINRES-Verfahrens auftretenden Ausgleichsproblems wieder ersetzen durch das Minimierungsproblem

$$\min \|\zeta_0 e_1 - \overline{T}_k y\|, \quad y \in \mathbb{R}^k \tag{6.15}$$

mit $\zeta_0 := \|r^0\|$. Damit ist die Grundversion des MINRES-Verfahrens äquivalent zu der folgenden Vorgehensweise:

```

Wähle  $x^0 \in \mathbb{R}^n$ .
Setze  $r^0 := b - Ax^0$ ,  $\zeta_0 := \|r^0\|$ .
FOR  $k = 1, 2, \dots$ 
    Berechne die Matrix  $\overline{T}_k$  aus (6.13).
    Bestimme eine Lösung  $y^k$  des Ausgleichsproblems (6.15).
    Setze  $x^k := x^0 + V_k y^k$ .
END
    
```

Die Elemente der Matrix \overline{T}_k lassen sich aber mit dem Lanczos-Verfahren aus dem Algorithmus 6.18 berechnen, so dass sich das obige Verfahren wie folgt spezifizieren lässt:

```

Wähle  $x^0 \in \mathbb{R}^n$ .
Setze  $r^0 := b - Ax^0$ ,  $\zeta_0 := \|r^0\|$ ,  $v^0 := 0$ ,  $v^1 := r^0/\zeta_0$  und  $\beta_1 := 0$ .
FOR  $k = 1, 2, \dots$ 
     $w^k := Av^k - \beta_k v^{k-1}$ ;
     $\alpha_k := (w^k)^T v^k$ ;
     $w^k := w^k - \alpha_k v^k$ ;
     $\beta_{k+1} := \|w^k\|$ ;
     $v^{k+1} := w^k/\beta_{k+1}$ ;
    Bestimme eine Lösung  $y^k$  des Ausgleichsproblems (6.15).
    Setze  $x^k := x^0 + V_k y^k$ .
END
    
```

Soweit ist die Vorgehensweise im Prinzip analog zu jener beim GMRES-Verfahren. Da \overline{T}_k als Matrix von der Form

$$\overline{T}_k = \begin{pmatrix} * & * & & & 0 \\ * & * & * & & \\ & * & \ddots & \ddots & \\ & & \ddots & \ddots & * \\ 0 & & & \ddots & * \\ & & & & * \end{pmatrix} \in \mathbb{R}^{(k+1) \times k}$$

aber eine einfachere Gestalt hat als die entsprechende Matrix \overline{H}_k beim GMRES-Verfahren, ergibt sich hier eine wesentlich kostengünstigere Methode zur Lösung des Ausgleichsproblems (6.15). Im Prinzip hat man wieder

k Givens-Rotationen $G_{21}, G_{32}, \dots, G_{k+1,k}$ anzuwenden, um die untere Nebendiagonale von \bar{T}_k zu eliminieren. Dies erzeugt im oberen Dreieck allerdings eine weitere Nebendiagonale, so dass sich nach erfolgter Anwendung der Givens-Rotationen eine Matrix der folgenden Gestalt ergibt:

$$G_{k+1,k} \dots G_{32} G_{21} \bar{T}_k = \begin{pmatrix} * & * & * & 0 \\ & * & * & \ddots \\ & & * & \ddots & * \\ & & & \ddots & * \\ & & & & * \\ 0 & & & & 0 \end{pmatrix} =: \bar{R}_k =: \begin{pmatrix} R_k \\ 0 \end{pmatrix} \quad (6.16)$$

mit einer oberen Dreiecksmatrix $R_k \in \mathbb{R}^{k \times k}$, bei der allerdings nur drei Diagonalen besetzt sind. Hat man diese Berechnungen in der Iteration k durchgeführt, so stellt sich die Situation im nächsten Iterationsschritt $k + 1$ erneut einfacher dar, denn die Matrix \bar{T}_{k+1} unterscheidet sich von der Matrix \bar{T}_k im Prinzip wieder nur dadurch, dass eine neue Spalte am Ende hinzugefügt wird:

$$\bar{T}_{k+1} = \begin{pmatrix} & & & 0 \\ & & & \vdots \\ & & & 0 \\ & \bar{T}_k & & \beta_{k+1} \\ & & & \alpha_{k+1} \\ \hline 0 & \dots & 0 & \beta_{k+2} \end{pmatrix}$$

Um diese Matrix auf obere Dreiecksgestalt zu bringen, hat man wieder nur die untere Nebendiagonale zu eliminieren. Für den \bar{T}_k -Block haben wir dies aber schon im vorherigen Schritt k getan, so dass die dort berechneten Givens-Rotationen $G_{21}, \dots, G_{k+1,k}$ nur noch auf die ersten $k + 1$ Elemente der letzten Spalte von \bar{T}_{k+1} angewendet werden müssen. Im Gegensatz zur Vorgehensweise beim GMRES-Verfahren stehen in dieser letzten Spalte aber fast nur Nullen, so dass die Anwendung der meisten dieser Givens-Rotationen auf diese letzte Spalte gar nichts bewirkt. Tatsächlich sorgen nur die Givens-Rotationen $G_{k,k-1}$ und $G_{k+1,k}$ für Veränderungen in der letzten Spalte von \bar{T}_{k+1} . Alle anderen zuvor berechneten Givens-Rotationen, also $G_{21}, \dots, G_{k-1,k-2}$, benötigt man im Folgenden nicht mehr. Insbesondere brauchen diese Givens-Rotationen (oder die sie definierenden Elemente) auch nicht abgespeichert zu werden.

Natürlich hat man in der Iteration $k + 1$ dann noch eine neue Givens-Rotation $G_{k+2,k+1}$ zu berechnen, um damit das Element β_{k+2} in dem unteren Eck von \bar{T}_{k+1} zu eliminieren.

Insgesamt braucht man in einer Iteration des MINRES-Verfahrens daher nur 3 Givens-Rotationen an Stelle der k Givens-Rotationen im k -ten Iterationsschritt des GMRES-Verfahrens. Damit haben wir zwar eine effiziente

Methode zur Bestimmung einer Lösung y^k des Ausgleichsproblems (6.15) gefunden, für die Berechnung des eigentlich interessierenden Vektors x^k über die Formel

$$x^k = x^0 + V_k y^k \quad (6.17)$$

verwendet man allerdings alle Spaltenvektoren der Matrix V_k , was für symmetrische Matrizen A einen viel zu großen Speicherplatz–Aufwand darstellt. Tatsächlich ist es möglich, den Vektor x^k mittels einer kurzen Rekursion aus bereits bekannten Daten herzuleiten. Zu diesem Zweck erinnern wir daran, dass sich y^k in der obigen Notation als Lösung des Ausgleichsproblems

$$\min \|\zeta_0 e_1 - \overline{T}_k y\|, \quad y \in \mathbb{R}^k,$$

ergab, das mit Hilfe geeigneter Givens–Rotationen äquivalent formuliert wurde als

$$\min \|\overline{z}^k - \overline{R}_k y\|.$$

Hierbei ist \overline{R}_k natürlich die Matrix aus (6.16) und

$$\overline{z}^k =: \begin{pmatrix} z^k \\ \zeta_k \end{pmatrix} \in \mathbb{R}^{k+1} \quad (6.18)$$

der sich entsprechend aus $\zeta_0 e_1$ durch sukzessive Multiplikation mit den Givens–Rotationen $G_{21}, G_{32}, \dots, G_{k+1,k}$ ergebende Vektor. Mit (6.16)–(6.18) folgt daher

$$\begin{aligned} \|b - Ax^k\|^2 &= \|\zeta_0 e_1 - \overline{T}_k y^k\|^2 \\ &= \|\overline{z}^k - \overline{R}_k y^k\|^2 \\ &= \left\| \begin{pmatrix} z^k \\ \zeta_k \end{pmatrix} - \begin{pmatrix} R_k \\ 0 \end{pmatrix} y^k \right\|^2 \\ &= \|z^k - R_k y^k\|^2 + |\zeta_k|^2 \\ &= |\zeta_k|^2, \end{aligned}$$

denn y^k ergibt sich gerade durch Rückwärtssubstitution aus dem linearen Gleichungssystem $R_k y = z^k$. Insbesondere gibt die Größe

$$|\zeta_k|,$$

also der Betrag der letzten Komponente des Vektors \overline{z}^k , gerade das aktuelle Residuum an. Dies ist konsistent mit der Definition von $\zeta_0 = \|r^0\|$.

Für spätere Zwecke vermerken wir an dieser Stelle auch noch, dass sich der Vektor $\overline{z}^k \in \mathbb{R}^{k+1}$ per Definition rekursiv aus

$$\overline{z}^k = G_{k+1,k} \begin{pmatrix} \overline{z}^{k-1} \\ 0 \end{pmatrix} \in \mathbb{R}^{k+1} \quad (6.19)$$

ergibt. Ist hierbei $G_{k+1,k} = G_{k+1,k}(c_k, s_k)$ und schreiben wir

$$\bar{z}^k = \begin{pmatrix} z^k \\ \zeta_k \end{pmatrix} \in \mathbb{R}^{k+1}, \quad \bar{z}^{k-1} = \begin{pmatrix} z^{k-1} \\ \zeta_{k-1} \end{pmatrix} \in \mathbb{R}^{(k-1)+1}$$

in Analogie zu (6.18), so folgt aus der Beziehung (6.19) unmittelbar

$$\begin{pmatrix} z^k \\ \zeta_k \end{pmatrix} = \bar{z}^k = G_{k+1,k} \begin{pmatrix} \bar{z}^{k-1} \\ 0 \end{pmatrix} = G_{k+1,k} \begin{pmatrix} z^{k-1} \\ \zeta_{k-1} \\ 0 \end{pmatrix} = \begin{pmatrix} z^{k-1} \\ c_k \zeta_{k-1} \\ -s_k \zeta_{k-1} \end{pmatrix},$$

denn die Givens-Rotation $G_{k+1,k}(c_k, s_k)$ wirkt nur auf die letzten zwei Komponenten von \bar{z}^k , ändert also nur die letzte Komponente von z^k . Also ist

$$z^k = \begin{pmatrix} z^{k-1} \\ c_k \zeta_{k-1} \end{pmatrix} \quad \text{und} \quad \zeta_k = -s_k \zeta_{k-1}. \tag{6.20}$$

Wir setzen zur Abkürzung jetzt

$$P_k := V_k R_k^{-1} \iff P_k R_k = V_k$$

und bezeichnen die Spalten von P_k mit $p^1, p^2 \dots p^k$:

$$P_k = \begin{pmatrix} | & | & & | \\ p^1 & p^2 & \dots & p^k \\ | & | & & | \end{pmatrix}.$$

Ferner ist R_k von der speziellen Gestalt

$$R_k = \begin{pmatrix} r_{11} & r_{12} & r_{13} & & & & 0 \\ & r_{22} & r_{23} & r_{24} & & & \\ & & r_{33} & \ddots & \ddots & & \\ & & & \ddots & \ddots & & \\ & & & & \ddots & & r_{k-2,k} \\ & & & & & r_{k-1,k-1} & r_{k-1,k} \\ 0 & & & & & & r_{k,k} \end{pmatrix},$$

vergleiche (6.16). Man beachte hierbei, dass zunächst sowohl die Spalten p^j von P_k als auch die Elemente r_{ij} von R_k vom Index k abhängen. Die Konstruktion von R_k impliziert allerdings, dass die führende $(k-1) \times (k-1)$ -Blockmatrix von R_k gerade R_{k-1} ist, so dass die Elemente der Matrix R_k letztlich doch unabhängig vom Index k sind. Hieraus folgt dann wiederum, dass auch die Spalten der Matrix P_k nicht von k abhängen, denn aus $P_k R_k = V_k$ ergibt sich, dass die Spalten von P_k nichts anderes als eine Linearkombination der Spalten von V_k sind. Wegen

$$V_k = \begin{pmatrix} | & | & & | \\ v^1 & v^2 & \dots & v^k \\ | & | & & | \end{pmatrix}$$

und der speziellen Gestalt von R_k ergibt sich aus $P_k R_k = V_k$ nämlich die Beziehung

$$\begin{aligned} r_{11}p^1 &= v^1, \\ r_{12}p^1 + r_{22}p^2 &= v^2, \\ r_{13}p^1 + r_{23}p^2 + r_{33}p^3 &= v^3, \\ r_{24}p^2 + r_{34}p^3 + r_{44}p^4 &= v^4, \\ &\vdots \\ r_{k-2,k}p^{k-2} + r_{k-1,k}p^{k-1} + r_{kk}p^k &= v^k. \end{aligned}$$

Für $r_{kk} \neq 0$ liefert dies die Rekursionsvorschrift

$$p^k = (v^k - r_{k-2,k}p^{k-2} - r_{k-1,k}p^{k-1})/r_{kk}. \tag{6.21}$$

Wegen (6.20) folgt außerdem

$$\begin{aligned} x^k &= x^0 + V_k y^k \\ &= x^0 + V_k R_k^{-1} z^k \\ &= x^0 + P_k z^k \\ &= x^0 + [P_{k-1} \mid p^k] \begin{bmatrix} z^{k-1} \\ c_k \zeta_{k-1} \end{bmatrix} \\ &= \underbrace{x^0 + P_{k-1} z^{k-1}}_{=x^{k-1}} + c_k \zeta_{k-1} p^k \\ &= x^{k-1} + c_k \zeta_{k-1} p^k. \end{aligned}$$

Dies ist die gewünschte Aufdatierungsformel für den Iterationsvektor x^k , die ohne Abspeicherung aller Vektoren v^1, \dots, v^k auskommt, da p^k wegen (6.21) ebenfalls einer kurzen Rekursion genügt.

Ferner muss man y^k bei dieser Vorgehensweise gar nicht berechnen, da die eigentlich interessierende Größe x^k explizit vorliegt. Aus diesem Grunde braucht auch die obere Dreiecksmatrix R_k nicht weiter abgespeichert zu werden, sondern lediglich die drei von Null verschiedenen Einträge $r_{k-2,k}$, $r_{k-1,k}$ und r_{kk} aus der letzten Spalte, die wir im Folgenden einfach mit ρ_1 , ρ_2 und ρ_3 bezeichnen. Außerdem ist es nicht nötig, die Elemente c_k und s_k aller Givens-Rotationen $G_{k+1,k}$ zu speichern, da pro Iteration lediglich drei solcher Matrizen benötigt werden. Aus diesem Grunde benutzen wir hierfür die Bezeichnungen c_{old} , s_{old} , c , s und c_{new} , s_{new} . Unter Verwendung dieser Notation liefert die Anwendung der beiden alten Givens-Rotationen $G_{k-1,k-2}$ und $G_{k,k-1}$ auf die letzte Spalte der Matrix \bar{T}_k beispielsweise

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \beta_k \\ \alpha_k \\ \beta_{k+1} \end{pmatrix} \cdot G_{k-1,k-2} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ s_{\text{old}}\beta_k \\ c_{\text{old}}\beta_k \\ \alpha_k \\ \beta_{k+1} \end{pmatrix} \cdot G_{k,k-1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ s_{\text{old}}\beta_k \\ c \cdot c_{\text{old}} \cdot \beta_k + s \cdot \alpha_k \\ -s \cdot c_{\text{old}} \cdot \beta_k + c \cdot \alpha_k \\ \beta_{k+1} \end{pmatrix} =: \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \rho_1 \\ \rho_2 \\ \tilde{\rho}_3 \\ \beta_{k+1} \end{pmatrix}.$$

Dabei entspricht $\tilde{\rho}_3$ noch nicht der finalen Größe ρ_3 , denn $\tilde{\rho}_3$ wird ja noch durch die anschließende neue Givens-Rotation $G_{k+1,k}$ zur Elimination von β_{k+1} verändert. Dies geschieht durch Verwendung der Vorschriften

$$\begin{aligned}\tau &:= |\tilde{\rho}_3| + |\beta_{k+1}|, \\ \nu &:= \tau \cdot \sqrt{(\tilde{\rho}_3/\tau)^2 + (\beta_{k+1}/\tau)^2}, \\ c_{\text{new}} &:= \tilde{\rho}_3/\nu, \\ s_{\text{new}} &:= \beta_{k+1}/\nu,\end{aligned}$$

mit deren Hilfe sich

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \rho_1 \\ \rho_2 \\ \tilde{\rho}_3 \\ \beta_{k+1} \end{pmatrix} \xrightarrow{G_{k+1,k}} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \rho_1 \\ \rho_2 \\ \nu \\ 0 \end{pmatrix} =: \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \rho_1 \\ \rho_2 \\ \rho_3 \\ 0 \end{pmatrix} \in \mathbb{R}^{k+1}$$

ergibt. Schreiben wir schließlich $p^{\text{new}}, p, p^{\text{old}}$ an Stelle von p^k, p^{k-1}, p^{k-2} und verfahren ähnlich bei einigen anderen Größen, so erhalten wir durch Zusammenfassung der obigen Ausführungen die nachstehende Realisierung des MINRES-Verfahrens.

Algorithmus 6.19. (MINRES-Verfahren)

Wähle Startvektor $x^0 \in \mathbb{R}^n$ und Abbruchparameter $\varepsilon \geq 0$.

Setze $k := 0, r^0 := b - Ax^0, \zeta_0 := \|r^0\|, \zeta := \zeta_0, v := 0, v^{\text{new}} := r^0/\zeta_0, \beta_{\text{new}} := 0$.

Setze $c := 1, s := 0, c_{\text{new}} := 1, s_{\text{new}} := 0$.

Setze $p := 0, p^{\text{new}} := 0$ und $x := x^0$.

FOR $k = 1, 2, \dots$

 % Lanczos-Algorithmus zur Berechnung von \bar{T}_k

$\beta := \beta_{\text{new}};$

$v^{\text{old}} := v, v := v^{\text{new}};$

$v^{\text{new}} := Av - \beta v^{\text{old}};$

$\alpha := (v^{\text{new}})^T v;$

$v^{\text{new}} := v^{\text{new}} - \alpha v;$

$\beta_{\text{new}} := \|v^{\text{new}}\|;$

$v^{\text{new}} := v^{\text{new}}/\beta_{\text{new}};$

 % Wende alte Givens-Rotationen auf letzte Spalte von \bar{T}_k an

$c_{\text{old}} := c, s_{\text{old}} := s, c := c_{\text{new}}, s := s_{\text{new}};$

$\rho_1 := s_{\text{old}} \cdot \beta;$

$\rho_2 := c \cdot c_{\text{old}} \cdot \beta + s \cdot \alpha;$

$\tilde{\rho}_3 := c \cdot \alpha - s \cdot c_{\text{old}} \cdot \beta;$

```

% Neue Givens-Rotation zur Elimination von  $\bar{T}_k(k+1, k)$ 
 $\tau := |\tilde{\rho}_3| + |\beta_{new}|;$ 
 $\nu := \tau \cdot \sqrt{(\tilde{\rho}_3/\tau)^2 + (\beta_{new}/\tau)^2};$ 
 $c_{new} := \tilde{\rho}_3/\nu;$ 
 $s_{new} := \beta_{new}/\nu;$ 
 $\rho_3 := \nu;$ 
% Aufdatierung von  $P_k$ 
 $p^{old} := p, p := p^{new};$ 
 $p^{new} := (v - \rho_1 p^{old} - \rho_2 p)/\rho_3;$ 
% Berechnung der neuen Iterierten  $x^k$ 
 $x := x + c_{new} \cdot \zeta \cdot p^{new};$ 
% Abbruchkriterium
 $\zeta := -s_{new} \cdot \zeta;$ 
IF  $|\zeta|/\zeta_0 \leq \varepsilon$ : STOP
END

```

Numerisch mag es sinnvoll sein, die Anweisung

$$v^{new} := v^{new}/\beta_{new}$$

erst in der nachfolgenden Iteration durchzuführen, da der Vektor v^{new} (im Gegensatz zu der skalaren Größe β_{new}) auch erst dort benötigt wird und die Division durch β_{new} kurz vor Abbruch des Verfahrens kritisch sein kann.

Das MINRES-Verfahren benötigt zur Durchführung neben dem Bedarf zur Ausführung des Matrix-Vektor-Produktes Av einen Speicherplatz von im Wesentlichen 7 Vektoren, nämlich für v^{old} , v , v^{new} , p^{old} , p , p^{new} und x . Das Matrix-Vektor-Produkt Av wird nur einmal gebraucht und kann daher auf dem Platz von v^{new} zwischengespeichert werden.

Als wesentlicher Rechenaufwand ergeben sich für den Algorithmus 6.19

- 1 Level 2-Operation und
- 7 Level 1-Operationen.

Die Level 2-Operation ist hierbei gerade das Matrix-Vektor-Produkt Av , und die 7 Level 1-Operationen setzen sich zusammen aus

- 2 Skalarprodukten (zur Berechnung von α und β_{new}) sowie
- 5 saxpy's (für x , v^{new} und p^{new}),

wobei die Berechnung der Vektoren v^{new} und p^{new} jeweils zwei saxpy's benötigt.

Testen wir das MINRES-Verfahren aus dem Algorithmus 6.19 an dem Beispiel der zweidimensionalen diskretisierten Helmholtz-Gleichung

$$H_{N \times N} v = h^2 f$$

aus dem Abschnitt 1.8 mit $\lambda = 3.0$, $N = 32$ (also $H_{N \times N} \in \mathbb{R}^{1024 \times 1024}$), der rechten Seite $f := (1, \dots, 1)^T$, dem Startvektor $x^0 := (0, \dots, 0)^T$ und dem

Parameter $\varepsilon = 10^{-6}$, so bricht das Verfahren nach 190 Iterationen mit der gewünschten Genauigkeit ab. Der zugehörige Verlauf der relativen Residuen ist in der Abbildung 6.3 wiedergegeben.

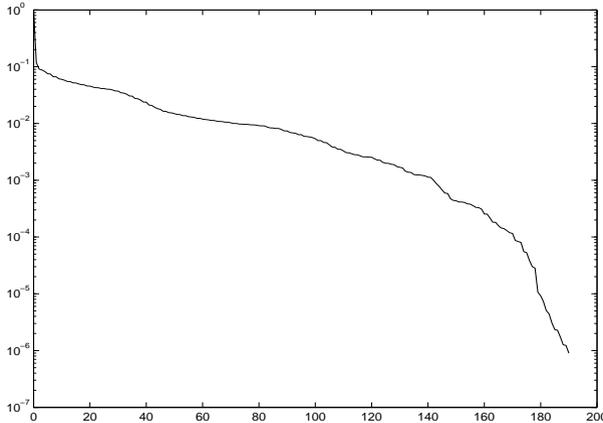


Abbildung 6.3. Verlauf des relativen Residuums beim MINRES-Verfahren

Übrigens wird dieses Beispiel auch mit dem CG-Verfahren innerhalb von 199 Iterationen gelöst. Der Verlauf der zugehörigen relativen Residuen ist allerdings wesentlich stärker oszillierend als beim MINRES-Verfahren, vergleiche die zugehörige Abbildung 6.4. Tatsächlich lassen sich symmetrische (nicht notwendig positiv definite) Gleichungssysteme oft auch mit dem CG-Verfahren lösen, allerdings ist der Residuenverlauf dann stets erheblichen Schwankungen ausgesetzt, die letztlich dadurch entstehen, dass die Koeffizientenmatrix negative Eigenwerte besitzt.

Bemerkung 6.20. (a) Von Paige und Saunders [47] stammt noch eine zweite Methode zur Lösung von linearen Gleichungssystemen mit einer symmetrischen Matrix, das sie als SYMMLQ bezeichnen. Dabei handelt es sich um ein Ritz-Galerkin-Verfahren. SYMMLQ bestimmt die Iterierte x^k also aus den Bedingungen

$$x^k \in x^0 + \mathcal{K}_k \quad \text{und} \quad b - Ax^k \perp \mathcal{K}_k.$$

Benutzen wir die Bezeichnungen aus diesem Abschnitt und schreiben $x^k = x^0 + V_k y^k$ mit einem $y^k \in \mathbb{R}^k$, so sind diese Bedingungen äquivalent zu

$$0 = V_k^T (b - Ax^k) = V_k^T (r^0 - AV_k y^k) = \|r^0\| e_1 - V_k^T AV_k y^k = \|r^0\| e_1 - T_k y^k.$$

Da T_k tridiagonal ist, kann man y^k hieraus sehr leicht berechnen. Mit einem ähnlichen Trick wie beim MINRES-Verfahren erhält man hieraus sogar direkt

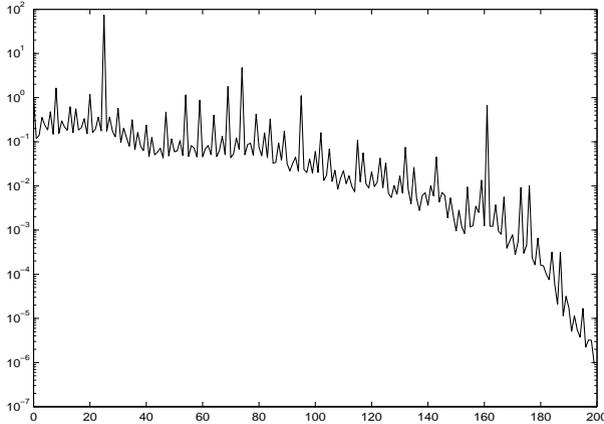


Abbildung 6.4. Verlauf des relativen Residuums beim CG-Verfahren

eine Aufdatierungsvorschrift für die Iterierten x^k , vergleiche [47].

(b) Der hier benutzte Zugang zur Herleitung des MINRES-Verfahrens kann auch zur Konstruktion des CG-Verfahrens benutzt werden. Weitere Einzelheiten hierzu findet man beispielsweise in dem Buch [17] von Demmel.

6.5 Präkonditionierte Verfahren

Will man die Effizienz und Robustheit eines iterativen Verfahrens zur Lösung eines linearen Gleichungssystems

$$Ax = b \tag{6.22}$$

mit $A \in \mathbb{R}^{n \times n}$ regulär und $b \in \mathbb{R}^n$ verbessern, so wendet man die iterativen Methoden üblicherweise auf ein präkonditioniertes System an, welches die gleiche Lösung wie das Ausgangssystem (6.22) besitzt. Sind $P_L, P_R \in \mathbb{R}^{n \times n}$ etwa zwei reguläre Matrizen, so ist (6.22) beispielsweise äquivalent zu jedem der folgenden Systeme:

$$P_L^{-1}Ax = P_L^{-1}b \quad (\text{Links-Präkonditionierung}), \tag{6.23}$$

$$AP_R^{-1}y = b, \quad x = P_R^{-1}y \quad (\text{Rechts-Präkonditionierung}), \tag{6.24}$$

$$P_L^{-1}AP_R^{-1}y = P_L^{-1}b, \quad x = P_R^{-1}y \quad (\text{zweiseitige Präkonditionierung}). \tag{6.25}$$

Präkonditionierte Verfahren entstehen im Prinzip dadurch, dass man das jeweils gewünschte Verfahren auf das entsprechend präkonditionierte Gleichungssystem anwendet. Ähnlich wie beim CG-Verfahren sollte die hierbei auftretende Koeffizientenmatrix $P_L^{-1}A, AP_R^{-1}$ bzw. $P_L^{-1}AP_R^{-1}$ dabei zum

einen möglichst nahe an der Identität I_n liegen, und zum anderen müssen zusätzlich auftretende lineare Gleichungssysteme mit den Koeffizientenmatrizen P_L oder P_R leicht lösbar sein.

In diesem Abschnitt gehen wir auf die explizite Wahl von P_L und P_R nicht weiter ein und verweisen hierfür auf den kommenden Abschnitt 6.6 oder den schon im Zusammenhang mit dem CG-Verfahren behandelten Splitting-basierten Präkonditioniern aus dem Abschnitt 5.5. Stattdessen überlegen wir uns hier, wie sich die in diesem Kapitel besprochenen GMRES-, GMRES(m)- und MINRES-Verfahren auf ein präkonditioniertes Gleichungssystem übertragen lassen.

Dazu untersuchen wir zunächst das GMRES-Verfahren, wobei entsprechende Überlegungen natürlich auch für das GMRES(m)-Verfahren gelten. Da das GMRES-Verfahren prinzipiell auf beliebige reguläre Matrizen anwendbar ist, können wir dieses Verfahren auf jedes der vorkonditionierten Gleichungssysteme (6.23), (6.24) oder (6.25) anwenden.

Benutzen wir das GMRES-Verfahren beispielsweise für das links-präkonditionierte System (6.23), so ergibt eine direkte Übertragung aus dem Algorithmus 6.12 das nachstehende Verfahren, wobei wir hier $q^k := P_L^{-1}(b - Ax^k)$ für das Residuum des präkonditionierten Systems schreiben.

Algorithmus 6.21. (*Links-präkonditioniertes GMRES-Verfahren*)

Wähle Startvektor $x^0 \in \mathbb{R}^n$, $P_L \in \mathbb{R}^{n \times n}$ regulär und Abbruchparameter $\varepsilon \geq 0$. Bestimme q^0 aus $P_L q^0 = b - Ax^0$, und setze $\beta := \|q^0\|$, $v^1 := q^0/\beta$ und $z_1 := \beta$.

FOR $k = 1, 2, \dots$

 % Modifizierter Arnoldi-Algorithmus zur Berechnung von \overline{H}_k

 Bestimme w^k aus $P_L w^k = Av^k$;

 FOR $i = 1 : k$

$$h_{ik} := (v^i)^T w^k;$$

$$w^k := w^k - h_{ik} v^i;$$

 END

$$h_{k+1,k} := \|w^k\|;$$

$$v^{k+1} := w^k / h_{k+1,k};$$

 % Anwendung von Givens-Rotationen auf (6.11)

 % Wende alte Givens-Rotationen auf k -te Spalte von \overline{H}_k an

 FOR $i = 1 : (k - 1)$

$$\begin{pmatrix} h_{ik} \\ h_{i+1,k} \end{pmatrix} := \begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix} \begin{pmatrix} h_{ik} \\ h_{i+1,k} \end{pmatrix};$$

 END

 % Bestimme neue Givens-Rot. zur Elimination von $\overline{H}_k(k+1, k)$

$$\tau := |h_{kk}| + |h_{k+1,k}|;$$

$$\nu := \tau \cdot \sqrt{(h_{kk}/\tau)^2 + (h_{k+1,k}/\tau)^2};$$

$$c_k := h_{kk}/\nu;$$

$$s_k := h_{k+1,k}/\nu;$$

 % Anwendung der neuen Givens-Rotation auf \overline{H}_k

```

    hkk := ν;
    hk+1,k := 0;
    % Anwendung der neuen Givens-Rotation auf rechte Seite zk
    zk+1 := -skzk;
    zk := ckzk;
    % Abbruchkriterium
    IF |zk+1|/β ≤ ε: STOP
END
% Rückwärtssubstitution zur Berechnung von yk
yk := zk/hkk;
FOR i = (k - 1) : -1 : 1
    yi = (zi - ∑j=i+1k hijyj)/hii;
END
% Berechnung der Näherungslösung xk := x0 + Vkyk
xk := x0 + ∑i=1k yivi;

```

Ebenso elementar ist die Übertragung des GMRES-Verfahrens auf das rechts- bzw. zweiseitig präkonditionierte System (6.24) bzw. (6.25). Die Abbruchkriterien der drei präkonditionierten GMRES-Verfahren sind allerdings nicht direkt vergleichbar. Das ursprüngliche GMRES-Verfahren beispielsweise benutzt das relative Residuum

$$\frac{\|r^k\|}{\|r^0\|} \leq \varepsilon \quad \text{mit } r^k := b - Ax^k,$$

das links-präkonditionierte GMRES-Verfahren hingegen testet die Ungleichung

$$\frac{\|q^k\|}{\|q^0\|} \leq \varepsilon \quad \text{mit } q^k := P_L^{-1}(b - Ax^k),$$

und das rechts-präkonditionierte GMRES-Verfahren überprüft die Bedingung

$$\frac{\|p^k\|}{\|p^0\|} \leq \varepsilon \quad \text{mit } p^k := b - AP_R^{-1}y^k.$$

Mit $x^k := P_R^{-1}y^k$ ist die letztgenannte Ungleichung äquivalent zum Abbruchkriterium beim eigentlichen GMRES-Verfahren, während der Test beim links-präkonditionierten GMRES-Verfahren von dem des ursprünglichen GMRES-Verfahrens verschieden ist. Wir werden auf diese Problematik in Form der Aufgabe 7.9 noch einmal zurückkommen.

Hier dagegen wenden wir uns nun dem präkonditionierten MINRES-Verfahren zu. Das MINRES-Verfahren ist per Konstruktion nur auf symmetrische Gleichungssysteme anwendbar. Da die Symmetrie von A durch Links- oder Rechts-Multiplikation mit einer regulären Matrix im Allgemeinen aber zerstört wird, kommen die präkonditionierten Systeme (6.23) und (6.24) schon mal nicht in Frage. Stattdessen wenden wir das MINRES-Verfahren auf das beidseitig präkonditionierte System

$$C^{-1}AC^{-T}C^T x = C^{-1}b \quad (6.26)$$

mit einer regulären Matrix C an. Dies entspricht dem linearen Gleichungssystem (6.25) mit $P_L := C$ und $P_R := C^T$. Die Vorgehensweise beim MINRES-Verfahren ist dabei ähnlich zu der des prädiktionierten CG-Verfahrens, wobei wie dort die Matrix C nicht explizit vorliegt, sondern im Allgemeinen nur die Matrix

$$P := CC^T.$$

Insbesondere kommen für das MINRES-Verfahren daher nur symmetrische und positiv definite Prädiktionierer P in Frage.

Zur Herleitung des prädiktionierten MINRES-Verfahrens haben wir vor allem das Lanczos-Verfahren aus dem Algorithmus 6.18 geeignet anzupassen. Hierzu schreiben wir das lineare Gleichungssystem (6.26) in der Form

$$\tilde{A}\tilde{x} = \tilde{b} \quad (6.27)$$

mit

$$\tilde{A} := C^{-1}AC^{-T}, \tilde{x} = C^T x, \tilde{b} := C^{-1}b.$$

Wenden wir das Lanczos-Verfahren auf das System (6.27) an und versehen alle hierbei entstehenden Größen mit dem $\tilde{\cdot}$ -Symbol, so ergeben sich die folgenden Vorschriften:

Wähle $\tilde{x}^0 \in \mathbb{R}^n$.

Setze $\tilde{r}^0 := \tilde{b} - \tilde{A}\tilde{x}^0$, $\tilde{v}^0 := 0$, $\tilde{v}^1 := \tilde{r}^0 / \|\tilde{r}^0\|$, $\tilde{\beta}_1 := 0$.

FOR $j = 1 : k$

$$\tilde{w}^j := \tilde{A}\tilde{v}^j - \tilde{\beta}_j\tilde{v}^{j-1};$$

$$\tilde{\alpha}_j := (\tilde{w}^j)^T \tilde{v}^j;$$

$$\tilde{w}^j := \tilde{w}^j - \tilde{\alpha}_j\tilde{v}^j;$$

$$\tilde{\beta}_{j+1} := \|\tilde{w}^j\|;$$

$$\tilde{v}^{j+1} := \tilde{w}^j / \tilde{\beta}_{j+1};$$

END

Setzen wir

$$w^j := C\tilde{w}^j, v^j := C\tilde{v}^j, z^j := P^{-1}v^j, y^j := P^{-1}w^j, \alpha_j := \tilde{\alpha}_j, \beta_j := \tilde{\beta}_j,$$

so lassen sich die obigen Aufdatierungen schreiben als

$$\begin{aligned} w^j &= C\tilde{w}^j = C\tilde{A}\tilde{v}^j - \beta_j C\tilde{v}^{j-1} \\ &= CC^{-1}AC^{-T}C^{-1}v^j - \beta_j CC^{-1}v^{j-1} \\ &= AP^{-1}v^j - \beta_j v^{j-1} = Az^j - \beta_j v^{j-1}, \\ \alpha_j &= (\tilde{w}^j)^T \tilde{v}^j = (C^{-1}w^j)^T (C^{-1}v^j) \\ &= (w^j)^T C^{-T}C^{-1}v^j = (w^j)^T P^{-1}v^j = (w^j)^T z^j, \\ w^j &= C\tilde{w}^j = C\tilde{w}^j - \alpha_j C\tilde{v}^j = w^j - \alpha_j v^j, \end{aligned}$$

$$\begin{aligned}
\beta_{j+1} &= \|\tilde{w}^j\| = \sqrt{(\tilde{w}^j)^T \tilde{w}^j} \\
&= \sqrt{(C^{-1}w^j)^T (C^{-1}w^j)} = \sqrt{(w^j)^T C^{-T} C^{-1} w^j} \\
&= \sqrt{(w^j)^T P^{-1} w^j} = \sqrt{(w^j)^T y^j}, \\
v^{j+1} &= C\tilde{v}^{j+1} = C\tilde{w}^j / \beta_{j+1} = w^j / \beta_{j+1}.
\end{aligned}$$

Aus dem Algorithmus 6.19 erhält man somit das nachstehende vorkonditionierte MINRES-Verfahren.

Algorithmus 6.22. (*Präkonditioniertes MINRES-Verfahren*)

Wähle Startvektor $x^0 \in \mathbb{R}^n$, $P \in \mathbb{R}^{n \times n}$ symmetrisch positiv definit und Abbruchparameter $\varepsilon \geq 0$.

Setze $k := 0$, $r^0 := b - Ax^0$, und bestimme z aus $Pz = r^0$.

Setze $\zeta_0 := \|r^0\|$, $\zeta := \zeta_0$, $z := z/\zeta_0$, $v := 0$, $v^{new} := r^0/\zeta_0$, $\beta_{new} := 0$.

Setze $c := 1$, $s := 0$, $c_{new} := 1$, $s_{new} := 0$.

Setze $p := 0$, $p^{new} := 0$ und $x := x^0$.

FOR $k = 1, 2, \dots$

 % Lanczos-Algorithmus zur Berechnung von \bar{T}_k

$\beta := \beta_{new}$;

$v^{old} := v$, $v := v^{new}$;

$v^{new} := Az - \beta v^{old}$;

$\alpha := (v^{new})^T z$;

$v^{new} := v^{new} - \alpha v$;

 Bestimme z aus $Pz = v^{new}$;

$\beta_{new} := \sqrt{(v^{new})^T z}$;

$v^{new} := v^{new} / \beta_{new}$;

$z := z / \beta_{new}$;

 % Wende alte Givens-Rotationen auf letzte Spalte von \bar{T}_k an

$c_{old} := c$, $s_{old} := s$, $c := c_{new}$, $s := s_{new}$;

$\rho_1 := s_{old} \cdot \beta$;

$\rho_2 := c \cdot c_{old} \cdot \beta + s \cdot \alpha$;

$\tilde{\rho}_3 := c \cdot \alpha - s \cdot c_{old} \cdot \beta$;

 % Neue Givens-Rotation zur Elimination von $\bar{T}_k(k+1, k)$

$\tau := |\tilde{\rho}_3| + |\beta_{new}|$;

$\nu := \tau \cdot \sqrt{(\tilde{\rho}_3/\tau)^2 + (\beta_{new}/\tau)^2}$;

$c_{new} := \tilde{\rho}_3/\nu$;

$s_{new} := \beta_{new}/\nu$;

$\rho_3 := \nu$;

 % Aufdatierung von P_k

$p^{old} := p$, $p := p^{new}$;

$p^{new} := (v - \rho_1 p^{old} - \rho_2 p) / \rho_3$;

 % Berechnung der neuen Iterierten x^k

$x := x + c_{new} \cdot \zeta \cdot p^{new}$;

 % Abbruchkriterium

$\zeta := -s_{new} \cdot \zeta$;

 IF $|\zeta|/\zeta_0 \leq \varepsilon$: STOP

END

Der Algorithmus 6.22 unterscheidet sich vom eigentlichen MINRES-Verfahren im Wesentlichen nur darin, dass in jeder Iteration jetzt zusätzlich ein lineares Gleichungssystem mit dem Prädiktionierer P als Koeffizientenmatrix zu lösen ist.

Abschließend noch ein paar Worte zu einigen numerischen Resultaten der hier vorgestellten präkonditionierten Verfahren. Wendet man beispielsweise das präkonditionierte GMRES-Verfahren mit dem Gauß-Seidel-Vorkonditionierer $P_L := D - L$ (vergleiche Abschnitt 5.5) auf die Konvektions-Diffusions-Gleichung an, so benötigt das Verfahren lediglich 67 Iterationen gegenüber den 80 Iterationen beim nicht vorkonditionierten GMRES-Verfahren. Benutzt man denselben Prädiktionierer bei GMRES(m) mit $m = 20$, so werden nur 100 Iterationen statt der ursprünglichen 178 Iterationen benötigt.

Wenden wir das präkonditionierte MINRES-Verfahren mit dem SGS-Prädiktionierer (vergleiche wieder den Abschnitt 5.5) auf die zweidimensionale diskretisierte Poisson-Gleichung an, so werden jetzt 26 Iterationen gebraucht, während das eigentliche MINRES-Verfahren hierfür immerhin noch 50 Iterationen benötigt.

Man erkennt hieran, dass bereits sehr einfache Prädiktionierer die Iterationszahlen zum Teil drastisch verringern können.

6.6 Unvollständige LR-Zerlegung

Wir betrachten wieder ein lineares Gleichungssystem $Ax = b$ mit einer regulären Matrix $A \in \mathbb{R}^{n \times n}$ und einer beliebigen rechten Seite $b \in \mathbb{R}^n$. Wie im vorigen Abschnitt erwähnt, ist dieses Gleichungssystem äquivalent zu dem präkonditionierten System

$$P_L^{-1}AP_R^{-1}y = P_L^{-1}b, \quad x = P_R^{-1}y \quad (6.28)$$

mit regulären Matrizen $P_L, P_R \in \mathbb{R}^{n \times n}$. In diesem Abschnitt beschreiben wir nun eine weitere Möglichkeit für die Wahl von geeigneten Matrizen P_L und P_R . Diese basiert auf der so genannten *unvollständigen LR-Zerlegung* (engl.: incomplete LU decomposition, kurz: ILU) von A . Hierbei bestimmt man eine normierte untere Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$ und eine obere Dreiecksmatrix $R \in \mathbb{R}^{n \times n}$, die beide nur schwach besetzt sein sollen, so dass

$$A \approx LR \quad (6.29)$$

gilt. Man bestimmt also keine exakte LR-Zerlegung von A , da dies unter Umständen zu einem erheblichen fill-in führen kann, vergleiche die entsprechenden Ausführungen im Abschnitt 5.6 zur unvollständigen Cholesky-Zerlegung.

Hat man eine Zerlegung der Gestalt (6.29) gefunden, so lässt sich diese prinzipiell auf drei verschiedene Weisen für das präkonditionierte System (6.28) anwenden: Die Wahl von

$$P_L := LR, \quad P_R := I_n$$

liefert ein links-präkonditioniertes System, die Wahl von

$$P_L := I_n, \quad P_R := LR$$

ein rechts-präkonditioniertes System, und die Wahl von

$$P_L := L, \quad P_R := R$$

ergibt ein zweiseitig vorkonditioniertes Gleichungssystem.

Zur Bestimmung einer unvollständigen LR-Zerlegung gehen wir ähnlich vor wie bei der unvollständigen Cholesky-Zerlegung im Abschnitt 5.6. Dazu bezeichnet $\mathcal{B}(A) := \{(i, j) \mid a_{ij} \neq 0\}$ wieder die Besetztheitsstruktur von A . Ferner sei $\mathcal{M} \subseteq \{(i, j) \mid i, j = 1, \dots, n\}$ auch hier ein vorgegebenes Belegungsmuster, das üblicherweise die Menge $\mathcal{B}(A)$ umfasst. Die Menge \mathcal{M} soll angeben, an welchen Stellen die Faktoren L und R der Zerlegung (6.29) von Null verschiedene Einträge haben dürfen. Eine unmittelbare Übertragung des Algorithmus 2.18 liefert dann folgendes Verfahren.

Algorithmus 6.23. (*Unvollständige LR-Zerlegung*)

Wähle $\mathcal{M} \subseteq \{(i, j) \mid i, j = 1, \dots, n\}$ mit $(i, i) \in \mathcal{M}$ für alle $i = 1, \dots, n$.

```

FOR  $k = 1 : (n - 1)$ 
  FOR  $i = (k + 1) : n$ 
    IF  $(i, k) \in \mathcal{M}$ 
       $a_{ik} := a_{ik} / a_{kk}$ ;
      FOR  $j = (k + 1) : n$ 
        IF  $(i, j) \in \mathcal{M}$ 
           $a_{ij} := a_{ij} - a_{ik} \cdot a_{kj}$ ;
        END
      END
    END
  END
END

```

In der vorliegenden Form überschreibt der Algorithmus 6.23 die Matrix A mit den (unvollständigen) Faktoren L (ohne die Diagonale) und R . Die Dreiecksmatrizen L und R ergeben sich daher aus den Vorschriften

$$l_{ij} := \begin{cases} a_{ij} & \text{für } i > j \text{ mit } (i, j) \in \mathcal{M}, \\ 0 & \text{für } i > j \text{ mit } (i, j) \notin \mathcal{M}, \\ 1 & \text{für } i = j, \\ 0 & \text{für } i < j \end{cases}$$

und

$$r_{ij} := \begin{cases} a_{ij} & \text{für } i \leq j \text{ mit } (i, j) \in \mathcal{M}, \\ 0 & \text{für } i \leq j \text{ mit } (i, j) \notin \mathcal{M}, \\ 0 & \text{für } i > j. \end{cases}$$

Die Existenz einer unvollständigen LR-Zerlegung von A und damit die Durchführbarkeit des Algorithmus 6.23 ist natürlich nicht gewährleistet, schon gar nicht für ein beliebiges Belegungsmuster \mathcal{M} . Selbst für die Wahl $\mathcal{M} = \{(i, j) \mid i, j = 1, \dots, n\}$, bei der die unvollständige LR-Zerlegung mit der eigentlichen LR-Zerlegung aus dem Algorithmus 2.18 übereinstimmt, ist die Wohldefiniertheit des Algorithmus 6.23 keineswegs gesichert, da bekanntlich nicht jede Matrix eine LR-Zerlegung besitzt, vergleiche den Satz 2.14.

Für die Klasse der so genannten M-Matrizen haben Meijerink und van der Vorst [43] allerdings zeigen können, dass die unvollständige LR-Zerlegung bei beliebig gegebenem Belegungsmuster $\mathcal{M} \subseteq \{(i, j) \mid i, j = 1, \dots, n\}$ stets durchführbar ist. Wir gehen auf dieses Ergebnis auch in den Aufgaben 6.8–6.10 ein. Eine Verallgemeinerung dieses Resultates findet sich bei Manteuffel [42]. Ansonsten gibt es in der Literatur diverse Varianten der hier vorgestellten unvollständigen LR-Zerlegung, siehe [53].

Aufgaben

Aufgabe 6.1. Verifizieren Sie explizit, dass die durch das Arnoldi-Verfahren aus dem Algorithmus 6.7 erzeugten Vektoren v^1, \dots, v^k eine Orthonormalbasis des Krylov-Raumes \mathcal{K}_k bilden.

Aufgabe 6.2. Zeigen Sie, dass das allgemeine Minimum-Residuum-Verfahren aus dem Algorithmus 6.2 stets wohldefiniert ist.

Aufgabe 6.3. Seien $A \in \mathbb{R}^{n \times n}$ eine gegebene Matrix sowie \mathcal{K} und \mathcal{L} zwei Unterräume des \mathbb{R}^n . Zeigen Sie, dass die Matrix $B := W^T A V$ dann für alle Matrizen V und W , deren Spalten eine Basis von \mathcal{K} bzw. \mathcal{L} bilden, regulär ist, sofern eine der beiden folgenden Bedingungen erfüllt ist:

- (a) A ist symmetrisch und positiv definit und $\mathcal{L} = \mathcal{K}$.
- (b) A ist regulär und $\mathcal{L} = A\mathcal{K}$.

Aufgabe 6.4. Betrachten Sie das lineare Gleichungssystem $Ax = b$ mit

$$A := \begin{pmatrix} 0 & 1 & 0 \\ & 0 & \ddots \\ & & \ddots & 1 \\ 1 & & & 0 \end{pmatrix} \in \mathbb{R}^{n \times n} \quad \text{und} \quad b := \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n$$

und der Lösung $x^* := (0, 1, 0, \dots, 0)^T \in \mathbb{R}^n$. Untersuchen Sie, wie viele Iterationen

- (a) das GMRES–Verfahren,
- (b) das CGNR–Verfahren,
- (c) das CGNE–Verfahren

zur Lösung dieses Gleichungssystems benötigen, wenn man jeweils $x^0 := 0$ als Startvektor wählt.

Aufgabe 6.5. (Konvergenz von GMRES für diagonalisierbare Matrizen)
Sei $A \in \mathbb{R}^{n \times n}$ *diagonalisierbar*, also $A = XDX^{-1}$ mit einer regulären Matrix $X \in \mathbb{R}^{n \times n}$ und einer Diagonalmatrix $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ (aufgrund des Spektralsatzes 1.8 sind dann insbesondere symmetrische Matrizen diagonalisierbar). Dann gilt

$$\|r^k\|_2 \leq \kappa_2(X) \max_{\lambda \in \sigma(A)} |p(\lambda)| \|r^0\|$$

für jedes Polynom p vom Höchstgrad k mit $p(0) = 1$, wobei wir das GMRES–Verfahren auf das Gleichungssystem $Ax = b$ mit einem beliebigem Startvektor x^0 anwenden und $\kappa_2(X)$ die Spektral–Kondition der Matrix X bezeichnet.

Aufgabe 6.6. Betrachte ein lineares Gleichungssystem $Ax = b$ mit einer diagonalisierbaren Matrix $A \in \mathbb{R}^{n \times n}$, welche die k verschiedenen Eigenwerte $\lambda_1, \dots, \lambda_k$ besitze. Dann bricht das auf dieses Gleichungssystem angewandte GMRES–Verfahren nach spätestens k Iterationen mit der exakten Lösung von $Ax = b$ ab.

Hinweis: Wähle ein geeignetes Polynom in Aufgabe 6.5.

Aufgabe 6.7. Seien $A \in \mathbb{R}^{n \times n}$ und $v \in \mathbb{R}^n \setminus \{0\}$ gegeben. Betrachte das folgende Householder–Arnoldi–Verfahren (siehe Walker [65]):

Setze $z^{(1)} := v$.

FOR $j = 1 : (k + 1)$

Bestimme Householder–Matrix $\overline{H}_j \in \mathbb{R}^{(n-j+1) \times (n-j+1)}$ mit

$$\overline{H}_j(z_j^{(j)}, z_{j+1}^{(j)}, \dots, z_n^{(j)})^T = (*, 0, \dots, 0)^T,$$

und setze

$$H_j := \begin{pmatrix} I_{j-1} & 0 \\ 0 & \overline{H}_j \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

$$h^{(j-1)} := H_j z^{(j)};$$

$$q^{(j)} := H_1 H_2 \cdots H_j e_j;$$

$$z^{(j+1)} := H_j H_{j-1} \cdots H_1 A q^{(j)}, \text{ falls } j \leq k;$$

END

Zeigen Sie der Reihe nach:

- (a) Es ist

$$H_k \cdots H_1 (v \ Aq^{(1)} \ Aq^{(2)} \ \cdots \ Aq^{(k)}) = (h^{(0)} \ h^{(1)} \ \cdots \ h^{(k)}).$$

- (b) Die Vektoren $q^{(1)}, q^{(2)}, \dots, q^{(k)}$ sind zueinander orthogonal.
 (c) Es gilt

$$Aq^{(j)} = \sum_{i=1}^{j+1} h_{ij}q^{(i)} \quad \forall j = 1, \dots, k, \tag{6.30}$$

wobei $h_{ij} := h_i^{(j)}$ die i -te Komponente des Vektors $h^{(j)}$ bezeichnet.

- (d) Die Vektoren $q^{(1)}, q^{(2)}, \dots, q^{(k)}$ bilden ein Orthonormalbasis des Krylov-Raums $\mathcal{K}_k(v, A)$, sofern $h_{j+1}^{(j)} \neq 0$ ist.

Bemerkung: Mit

$$\begin{aligned} V_k &:= (q^{(1)} \ q^{(2)} \ \dots \ q^{(k)}) \in \mathbb{R}^{n \times k}, \\ V_{k+1} &:= (q^{(1)} \ q^{(2)} \ \dots \ q^{(k)} \ q^{(k+1)}) \in \mathbb{R}^{n \times (k+1)}, \\ \overline{H}_k &:= (h_{ij})_{\substack{1 \leq i \leq k+1 \\ 1 \leq j \leq k}} \in \mathbb{R}^{(k+1) \times k} \end{aligned}$$

lässt sich (6.30) offenbar schreiben als $AV_k = V_{k+1}\overline{H}_k$, woraus man wie im Abschnitt 6.2 eine Art GMRES-Verfahren unter Verwendung von Householder-Spiegelungen herleiten kann. Man beachte hierbei, dass \overline{H}_k wieder die Gestalt einer oberen Hessenberg-Matrix hat!

Aufgabe 6.8. Sei $A \in \mathbb{R}^{n \times n}$ eine M -Matrix, d.h., A habe die folgenden Eigenschaften:

- A ist regulär,
- $a_{ij} \leq 0$ für alle $i, j = 1, \dots, n$ mit $i \neq j$,
- $A^{-1} \geq 0$.

Dann ist $a_{ii} > 0$ für alle $i = 1, \dots, n$.

Aufgabe 6.9. Sei $A \in \mathbb{R}^{n \times n}$ eine M -Matrix (vergleiche Aufgabe 6.8), so dass wegen Aufgabe 6.8 insbesondere $a_{11} \neq 0$ gilt und ein Schritt der Gauß-Elimination ohne Pivotisierung damit durchführbar ist. Mit

$$l_1 := \frac{1}{a_{11}} \begin{pmatrix} 0 \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \quad \text{und} \quad L_1 := I - l_1 e_1^T$$

liefern der erste Schritt der Gauß-Elimination die Matrix $A^{(2)} := L_1 A$. Zeigen Sie:

- (a) $A^{(2)}$ ist ebenfalls eine M -Matrix.
 (b) Die $(n-1) \times (n-1)$ -Matrix, die man aus $A^{(2)}$ durch Streichen der ersten Zeile und Spalte erhält, ist auch eine M -Matrix.

Aufgabe 6.10. (Durchführbarkeit der unvollständigen LR-Zerlegung)

Seien $A \in \mathbb{R}^{n \times n}$ eine M -Matrix und $\mathcal{M} \subseteq \{(i, j) \mid i, j = 1, \dots, n\}$ ein beliebig vorgegebenes Belegungsmuster, das zumindest alle Diagonalelemente (i, i) für $i = 1, 2, \dots, n$ enthalte. Dann ist der Algorithmus 6.23 durchführbar.

Hinweis: Aufgabe 6.9

Aufgabe 6.11. Implementieren Sie das GMRES-Verfahren und testen Sie es insbesondere am Beispiel der Konvektions-Diffusions-Gleichung. Schreiben Sie auch ein Programm für das GMRES(m)-Verfahren und testen Sie dieses an denselben Beispielen wie das GMRES-Verfahren, und zwar unter Verwendung von verschiedenen Werten von $m \in \mathbb{N}$.

Aufgabe 6.12. Implementieren Sie das MINRES-Verfahren und testen Sie es insbesondere am Beispiel der Helmholtz-Gleichung. Vergleichen Sie die numerischen Resultate auch mit dem auf dieselben Beispiele angewandten CG-Verfahren.

Aufgabe 6.13. Implementieren Sie ein präkonditioniertes GMRES-Verfahren unter Verwendung der unvollständigen LR-Zerlegung als Vorkonditionierer (etwa mit dem Belegungsmuster $\mathcal{M} = \mathcal{B}(A)$). Testen Sie das Verfahren insbesondere an der Konvektions-Diffusions-Gleichung. Wie verhält sich das Verfahren im Vergleich zum eigentlichen GMRES-Verfahren?