Das QMR-Verfahren

Bachelor-Arbeit
im 1-Fach Bachelorstudiengang Mathematik
der Mathematisch-Naturwissenschaftlichen Fakultät der
Christian-Albrechts-Universität zu Kiel

vorgelegt von
Janne Henningsen

Erstgutachter: Prof. Dr. Steffen Börm
Zweitgutachter: Prof. Dr. Malte Braack

Kiel im April 2018

Inhaltsverzeichnis

Notationen und Konventionen

Seien $m, n \in \mathbb{N}$.

- Für $x, y \in \mathbb{C}^n$ definieren wir das **euklidische Skalarprodukt** durch

$$\langle x, y \rangle := \sum_{i=1}^{n} \overline{x_i} y_i.$$

- Ebenso definieren wir für $x \in \mathbb{C}^n$ die **euklidische Norm** durch

$$||x|| := \sqrt{\langle x, x \rangle} = \sqrt{\sum_{i=1}^{n} \overline{x_i} x_i} = \sqrt{\sum_{i=1}^{n} |x_i|^2}.$$

  Darauf basierend bezeichnen wir im Folgenden für $M \in \mathbb{C}^{n \times m}$ mit $||M||$ die **von der euklidischen Norm induzierte Matrixnorm** von M und, sofern $m \leq n$ ist und M injektiv ist bzw. vollen Rang hat, $\kappa(M)$ als die **Konditionszahl von M bezüglich der euklidischen Norm**.

- Für $M \in \mathbb{C}^{n \times m}$ bezeichnen wir mit $M^* \in \mathbb{C}^{m \times n}$ die **Adjungierte von M**.

- Die **Einheitsmatrix** in $\mathbb{C}^{n \times n}$ bezeichnen wir mit I_n.

- Sei $M \in \mathbb{C}^{n \times m}$ und $m \leq n$. Falls $M^* M = I_m$ ist, nennen wir M **isometrisch**. Falls M isometrisch und zudem quadratisch ist, nennen wir M **unitär**.

- Wir setzen

$$\delta_{mn} := \begin{cases} 1, & \text{falls m = n gilt} \\ 0, & \text{sonst} \end{cases}.$$

- Mit Π_n bezeichnen wir die Menge aller Polynome über $\mathbb{C}$, deren Grad kleiner oder gleich n ist.

- Die **kanonischen Einheitsvektoren** in $\mathbb{C}^n$ bezeichnen wir mit $e_1, e_2, ..., e_n$.

1 Grundlagen

Seien $n \in \mathbb{N}$, $b \in \mathbb{C}^n$ und $A \in \mathbb{C}^{n \times n}$ eine reguläre Matrix. Das Ausgangsproblem, welches wir im Folgenden betrachten werden, ist das Lösen des linearen Gleichungssystems

$$Ax = b. \tag{1.1}$$

Wir suchen also eine Lösung $x^{(*)} \in \mathbb{C}^n$ mit $Ax^{(*)} = b$. Dazu benötigen wir zunächst ein paar Grundlagen.

Das Vorgehen in diesem Kapitel orientiert sich hauptsächlich an [1], [6] und [5].

1.1 Krylow-Räume

Alle in dieser Arbeit vorgestellten Verfahren zum Lösen von linearen Gleichungssystemen sind sogenannte *Krylow-Unterraum-Verfahren*, benannt nach speziellen Unterräumen des $\mathbb{C}^n$.

Definition 1.1 (Krylow-Raum) *Seien $M \in \mathbb{C}^{n \times n}$, $d \in \mathbb{C}^n$ und $m \in \mathbb{N}$. Dann heißt*

$$K_m(M, d) := span\{d, Md, M^2d, ..., M^{m-1}d\}$$

der m-te Krylow-Raum zu M und d.

Hieraus lassen sich unmittelbar ein paar Eigenschaften der Krylow-Räume ableiten.

Bemerkung 1.2 *Aus Definition 1.1 folgt direkt:*

- $dim(K_m(M, d)) \leq m$
- $K_m(M, d) = \{p(M)d \mid p \in \Pi_{m-1}\}$
- $K_1(M, d) \subseteq K_2(M, d) \subseteq \ldots \subseteq K_n(M, d)$.
  Die Krylow-Räume sind also geschachtelt. Man beachte, dass diese aufsteigende Folge von Teilräumen des $\mathbb{C}^n$ spätestens ab $K_n(M, d)$ stationär wird.

Das Besondere an den Krylow-Räumen ist jedoch die folgende, für die Behandlung von linearen Gleichungssystemen außerordentlich nützliche Eigenschaft, auf der sämtliche Krylow-Unterraum-Verfahren basieren.

Satz 1.3 *Seien $M \in \mathbb{C}^{n \times n}$ eine reguläre Matrix, $d \in \mathbb{C}^n$. Dann existiert ein $m \in \mathbb{N}$ mit $m \leq n$, sodass die Lösung des linearen Gleichungssystems $Mx = d$ ein Element des m-ten Krylow-Raums $K_m(M, d)$ ist.*

Beweis. Dieser Beweis basiert auf Abschnitt 1.3 in [1, S.11 -13].
Das charakteristische Polynom $p_M \in \Pi_n$ von M ist nicht das Nullpolynom. Außerdem gilt nach dem Satz von Cayley-Hamilton:

$$p_M(M) = 0.$$

Folglich existiert auch ein Polynom $p \in \Pi_n$ minimalen Grades mit diesen beiden Eigenschaften. Sei nun $g := deg(p)$. Dann ist $g \leq n$. Außerdem ist $g > 0$, denn sonst wäre p ein konstantes Polynom mit $p \neq 0$ und $p(M) = 0$. Weiter existieren Koeffizienten $c_0, c_1, c_2, ..., c_g \in \mathbb{C}$ mit

$$0 = p(M) = \sum_{i=0}^{g} c_i M^i.$$

Es gilt $c_0 \neq 0$, denn sonst wäre aufgrund der Regularität von M durch $q(t) := \sum_{i=1}^{g} c_i t^{i-1}$ ein Polynom $q \neq 0$ mit

$$q(M) = \sum_{i=1}^{g} c_i M^{i-1} = M^{-1} M \sum_{i=1}^{g} c_i M^{i-1} = M^{-1} \sum_{i=1}^{g} c_i M^i \overset{c_0 = 0}{=} M^{-1} p(M) = 0$$

und $deg(q) < g$ definiert, im Widerspruch zu der Minimalität von g. Damit erhalten wir

$$0 = \frac{1}{c_0} p(M) = \sum_{i=0}^{g} \frac{c_i}{c_0} M^i = \left[\sum_{i=1}^{g} \frac{c_i}{c_0} M^i \right] + I$$

und somit

$$I = \sum_{i=1}^{g} -\frac{c_i}{c_0} M^i = M \left[\sum_{i=1}^{g} -\frac{c_i}{c_0} M^{i-1} \right] = \left[\sum_{i=1}^{g} -\frac{c_i}{c_0} M^{i-1} \right] M.$$

Für die Inverse M^{-1} von M gilt also :

$$M^{-1} = \sum_{i=1}^{g} -\frac{c_i}{c_0} M^{i-1}.$$

Damit gilt insgesamt

$$M^{-1} d = \left[\sum_{i=1}^{g} -\frac{c_i}{c_0} M^{i-1} \right] d = \sum_{i=1}^{g} -\frac{c_i}{c_0} M^{i-1} d \in span\{d, Md, M^2 d, ..., M^{g-1} d\} = K_g(M, d).$$

□

Die gewonnenen, theoretischen Aussagen können wir nun verwenden, um unser Ausgangsproblem (1.1) in ein anderes Problem zu übersetzen.

1.2 Das Minimierungsproblem

Zur Konstruktion der Lösung $x^{(*)}$ von (1.1) wählen wir zunächst einen beliebigen Startvektor $x^{(0)} \in \mathbb{C}^n$ und definieren das anfängliche *Residuum* durch

$$r^{(0)} := b - Ax^{(0)}.$$

Nach Satz 1.3 befindet sich die Lösung $y^{(*)}$ des linearen Gleichungssystems

$$Ay = r^{(0)}$$

für ein $m \in \mathbb{N}_{\leq n}$ im m-ten Krylow-Raum $K_m(A, r^{(0)})$ zu A und $r^{(0)}$. Nun definieren wir

$$\begin{aligned} m_0 &:= max\{m \in \mathbb{N} \mid dim(K_m(A, r^{(0)})) = m\} \\ &= max\{m \in \mathbb{N} \mid r^{(0)}, Ar^{(0)}, ..., A^{m-1}r^{(0)} \text{ sind linear unabhängig}\} \\ &= min\{m \in \mathbb{N} \mid K_{m+1}(A, r^{(0)}) = K_m(A, r^{(0)})\}. \end{aligned}$$

Dann gilt insbesondere $K_{m_0}(A, r^{(0)}) = K_n(A, r^{(0)})$, sodass $y^{(*)}$ in $K_{m_0}(A, r^{(0)})$ enthalten sein muss. Außerdem gilt

$$A(x^{(0)} + y^{(*)}) = Ax^{(0)} + Ay^{(*)} = Ax^{(0)} + r^{(0)} = Ax^{(0)} + b - Ax^{(0)} = b.$$

Die Lösung $x^{(*)}$ von (1.1) ist also gegeben durch

$$x^{(*)} = x^{(0)} + y^{(*)} \in x^{(0)} + K_{m_0}(A, r^{(0)}) \tag{1.2}$$

und ist somit ein Element des affinen Unterraums $x^{(0)} + K_{m_0}(A, r^{(0)})$. Wenn wir nun nacheinander für m = 1,2,...,m_0 einen Vektor $x^{(m)} \in x^{(0)} + K_m(A, r^{(0)})$ mit der Eigenschaft

$$||x^{(*)} - x^{(m)}|| = min\{||x^{(*)} - x|| \mid x \in x^{(0)} + K_m(A, r^{(0)})\}$$

finden, so ist demnach spätestens die m_0-te Iterierte $x^{(m_0)} \in x^{(0)} + K_{m_0}(A, r^{(0)})$ die exakte Lösung $x^{(*)}$. Da wir $x^{(*)}$ im Allgemeinen natürlich nicht kennen, modifizieren wir diese Bedingung, indem wir die *Defektnorm* $x \mapsto ||Ax||$ verwenden. Wir suchen also nach einem Verfahren, welches nacheinander für $m = 1, ..., m_0$ einen Vektor $x^{(m)} \in x^{(0)} + K_m(A, r^{(0)})$ mit

$$||Ax^{(*)} - Ax^{(m)}|| = ||b - Ax^{(m)}|| = min\{||b - Ax|| \mid x \in x^{(0)} + K_m(A, r^{(0)})\} \tag{1.3}$$

konstruiert. Da die Krylow-Räume geschachtelt sind und $x^{(*)} \in x^{(0)} + K_{m_0}(A, r^{(0)})$ ist, gilt dann

$$||b - Ax^{(1)}|| \geq ||b - Ax^{(2)}|| \geq ... \geq ||b - Ax^{(m_0)}|| = 0.$$

Die Norm des *m-ten Residuums*

$$r^{(m)} := b - Ax^{(m)}$$

wird also mit jeder Iteration kleiner. Daher wird das gesuchte Verfahren in dieser Hinsicht mit jeder Iteration genauer und liefert spätestens nach m_0 Iterationen und somit insbesondere nach spätestens n Iterationen die exakte Lösung.
Das klassische Beispiel für ein solches Verfahren, welches für jede reguläre Matrix und somit auch für unser Ausgangsproblem (1.1) funktioniert, ist das sogenannte GMRES-Verfahren (generalized minimal residual method).

1.3 Die Konvektions-Diffusions-Gleichung

Bevor wir jedoch mit der Herleitung von Lösungsverfahren für unser Gleichungssystem (1.1) beginnen, wollen wir zunächst ein Modellproblem konstruieren, anhand dessen wir die verschiedenen vorgestellten Verfahren vergleichen können. Dazu verwenden wir die zweidimensionale *Konvektions-Diffusions-Gleichung.*
Wir betrachten das beschränkte Gebiet

$$\Omega := (0,1)^2 \subseteq \mathbb{R}^2.$$

Gesucht ist eine Abbildung $u : \overline{\Omega} \to \mathbb{R}$ mit $u \in C(\overline{\Omega})$ und $u|_\Omega \in C^2(\Omega)$, welche die Eigenschaften

$$\langle a, \nabla u(x,y)\rangle - \epsilon \Delta u(x,y) = f(x,y) \text{ für alle } (x,y) \in \Omega, \tag{1.4}$$

$$u(x,y) = 0 \text{ für alle } (x,y) \in \partial\Omega \tag{1.5}$$

für ein $a \in \mathbb{R}^2$, $\epsilon \in \mathbb{R}_{>0}$ und eine Abbildung $f \in C(\Omega)$ erfüllt. Dabei entspricht

$$\nabla u(x,y) = \Big(\frac{\partial u}{\partial x}(x,y), \frac{\partial u}{\partial y}(x,y)\Big)^T$$

dem *Gradienten* von u und

$$\Delta u(x,y) = \frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y)$$

der Anwendung des *Laplace-Operators* auf u.
Nun diskretisieren wir unser Gebiet Ω: Wir wählen zunächst ein $N \in \mathbb{N}$ und definieren darauf basierend die Schrittweite

$$h := \frac{1}{N+1}.$$

Dann definieren wir

$$x_i := ih, \;\; y_i := ih \text{ für alle } i \in \{0,1,...,N+1\}$$

und

$$u_{ij} := u(x_i, y_j), \;\; f_{ij} := f(x_i, y_j) \text{ für alle } i,j \in \{0,1,...,N+1\}.$$

Außerdem approximieren wir die benötigten Ableitungen von u durch geeignete Differenzenquotienten. Wir verwenden für alle $i,j \in \{1,...,N\}$

$$\frac{\partial u}{\partial x}(x_i,y_j) \approx \frac{u_{ij} - u_{i-1,j}}{h}, \;\; \frac{\partial u}{\partial y}(x_i,y_j) \approx \frac{u_{ij} - u_{i,j-1}}{h},$$

$$-\frac{\partial^2 u}{\partial x^2}(x_i,y_j) - \frac{\partial^2 u}{\partial y^2}(x_i,y_j) \approx \frac{4u_{ij} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}}{h^2}.$$

Indem wir dies mit (1.4) kombinieren, erhalten wir für alle $i,j \in \{1,...,N\}$:

$$u_{ij}(a_1 h + a_2 h + 4\epsilon) + u_{i-1,j}(-a_1 h - \epsilon) - \epsilon u_{i+1,j} + u_{i,j-1}(-a_2 h - \epsilon) - \epsilon u_{i,j+1} \approx h^2 f_{ij}.$$

Unter Verwendung der lexikographischen Anordnung und der Randbedingung (1.5) ergibt sich damit das lineare Gleichungssystem

$$C_{N\times N}u = h^2 f$$

mit der im Allgemeinen nicht symmetrischen, schwach besetzten Matrix

$$C_{N\times N} := \begin{pmatrix} B_N & -\epsilon I_N & & \\ D_N & B_N & \ddots & \\ & \ddots & \ddots & -\epsilon I_N \\ & & D_N & B_N \end{pmatrix} \in \mathbb{R}^{N^2\times N^2},$$

welche gegeben ist durch

$$B_N := \begin{pmatrix} 4\epsilon + h(a_1+a_2) & -\epsilon & & \\ -a_1 h - \epsilon & 4\epsilon + h(a_1+a_2) & \ddots & \\ & \ddots & \ddots & -\epsilon \\ & & -a_1 h - \epsilon & 4\epsilon + h(a_1+a_2) \end{pmatrix} \in \mathbb{R}^{N\times N}$$

und

$$D_N := (-a_2 h - \epsilon) I_N \in \mathbb{R}^{N\times N},$$

und den Vektoren

$$u := (u_{11}, u_{21}, ..., u_{N1}, u_{12}, u_{22}, ..., u_{N2}..., u_{1N}, u_{2N}, ..., u_{NN})^T \in \mathbb{R}^{N^2},$$

und

$$f := (f_{11}, f_{21}, ..., f_{N1}, f_{12}, f_{22}, ..., f_{N2}..., f_{1N}, f_{2N}, ..., f_{NN})^T \in \mathbb{R}^{N^2}.$$

Eine genaue Herleitung dieses Gleichungssystems ist in [4, Kapitel 1] und [5, Kapitel 1] zu finden.

Bemerkung 1.4 (Wahl der Parameter) *Für unsere Tests verwenden wir*

- $a = (cos(\frac{\pi}{4}), sin(\frac{\pi}{4}))^T$,
- $\epsilon = 1.0$,
- $f = (1, 1, ..., 1)^T$,
- *den Nullvektor als Startvektor,*
- $N = 32$.

Unser Gleichungssystem besitzt also eine Problemdimension von $n = N^2 = 1024$. *Darüber hinaus werden wir für die Abbruchkriterien der betrachteten Verfahren eine Fehlerschranke von* 10^{-6} *verwenden.*

1.4 Das GMRES-Verfahren

Das GMRES-Verfahren beruht auf der Konstruktion von Orthonormalbasen der Krylow-Räume. Dies geschieht mit dem sogenannten *Arnoldi-Verfahren.*
Sei $m \in \mathbb{N}$.

Algorithmus 1.5 ***(Arnoldi)***

1: Wähle einen Vektor $p_1 \in \mathbb{C}^n$ mit $||p_1|| = 1$;
2: **for** $j = 1, 2, ..., m$ **do**
3: **for** $i = 1, 2, ..., j$ **do**
4: $h_{ij} \leftarrow \langle p_i, Ap_j \rangle$;
5: **end for**
6: $w_j \leftarrow Ap_j - \sum_{i=1}^{j} h_{ij} p_i$;
7: $h_{j+1,j} \leftarrow ||w_j||$;
8: **if** $h_{j+1,j} = 0$ **then**
9: stop;
10: **end if**
11: $p_{j+1} \leftarrow w_j / h_{j+1,j}$;
12: **end for**

Man beachte, dass es durch Z. 8 - 10 zu einem vorzeitigen Abbruch kommen kann.

Satz 1.6 *Der Arnoldi-Algorithmus breche nicht vor der Konstruktion von p_m ab. Dann ist $\{p_1, p_2, ..., p_m\}$ eine Orthonormalbasis des m-ten Krylow-Raums $K_m(A, p_1)$ zu A und p_1.*

Beweis. Dieser Beweis basiert auf dem Beweis von Satz 4.79 aus [5, S.166-167].
Wir zeigen per Induktion:

$$\{p_1, p_2, ..., p_m\} \subseteq K_m(A, p_1), \;\; \langle p_i, p_j \rangle = \delta_{ij} \;\; \forall i, j \in \{1, ..., m\}. \tag{1.6}$$

Induktionsanfang:
Es ist nach Definition $p_1 \in span\{p_1\} = K_1(A, p_1)$. Außerdem ist durch die Wahl von p_1 sichergestellt, dass $\langle p_1, p_1 \rangle = ||p_1||^2 = 1$ gilt.
Induktionsvoraussetzung:
Sei nun $k \in \{1, ..., m-1\}$ so gewählt, dass $\{p_1, p_2, ..., p_k\} \subseteq K_k(A, p_1)$ und $\langle p_i, p_j \rangle = \delta_{ij}$ für alle $i, j \in \{1, ..., k\}$ ist.
Induktionsschritt:
Nach der Induktionsvoraussetzung sind sowohl p_k als auch $\sum_{i=1}^{k} h_{ik} p_i$ Elemente von $K_k(A, p_1)$.
Damit folgt $Ap_k \in K_{k+1}(A, p_1)$ und somit auch $w_k = Ap_k - \sum_{i=1}^{k} h_{ik} p_i \in K_{k+1}(A, p_1)$. Also

ist $p_{k+1} = \frac{w_k}{h_{k+1,k}} \in K_{k+1}(A, p_1)$ und folglich $\{p_1, p_2, ..., p_{k+1}\} \subseteq K_{k+1}(A, p_1)$.
Außerdem folgt mit der Induktionsvoraussetzung für alle $j \in \{1, ..., k\}$

$$\langle p_j, p_{k+1}\rangle = \frac{1}{h_{k+1,k}}\langle p_j, w_k\rangle = \frac{1}{h_{k+1,k}}\langle p_j, Ap_k - \sum_{i=1}^{k} h_{ik}p_i\rangle = \frac{1}{h_{k+1,k}}\Big[\langle p_j, Ap_k\rangle - \sum_{i=1}^{k} h_{ik}\langle p_j, p_i\rangle\Big]$$
$$= \frac{1}{h_{k+1,k}}\Big[\langle p_j, Ap_k\rangle - h_{jk}\langle p_j, p_j\rangle\Big] = \frac{1}{h_{k+1,k}}\Big[\langle p_j, Ap_k\rangle - h_{jk}\Big] = \frac{1}{h_{k+1,k}}[h_{jk} - h_{jk}] = 0$$

und wegen $h_{k+1,k} = ||w_k||$ gilt

$$\langle p_{k+1}, p_{k+1}\rangle = \frac{\langle w_k, w_k\rangle}{h_{k+1,k}^2} = \frac{\langle w_k, w_k\rangle}{||w_k||^2} = 1.$$

Also ist $\langle p_i, p_j\rangle = \delta_{ij} \ \ \forall i, j \in \{1, ..., k+1\}$ und (1.6) somit bewiesen.
Aus (1.6) folgt insbesondere, dass $\{p_1, p_2, ..., p_m\}$ eine linear unabhängige Teilmenge von $K_m(A, p_1)$ und somit wegen $dim(K_m(A, p_1)) \leq m$ eine Basis von $K_m(A, p_1)$ ist. Damit ist $\{p_1, p_2, ..., p_m\}$ insgesamt eine Orthonormalbasis von $K_m(A, p_1)$.

□

Sofern es nicht zu einem vorzeitigen Abbruch kommt, gehen aus dem Arnoldi-Algorithmus außerdem die obere Hessenberg-Matrix

$$\hat{H}_m := \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & h_{2m} \\ 0 & h_{32} & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_{mm} \\ 0 & \dots & 0 & h_{m+1,m} \end{pmatrix} \in \mathbb{C}^{(m+1)\times m}$$

und die Matrizen

$$P_m := (p_1 ... p_m) \in \mathbb{C}^{n\times m}, \quad P_{m+1} := (p_1 ... p_{m+1}) \in \mathbb{C}^{n\times(m+1)}$$

hervor, was uns zu einer weiteren, wichtigen Aussage führt. Dabei bezeichne $H_m \in \mathbb{C}^{m\times m}$ die quadratische Matrix , die aus $\hat{H}_m$ durch Streichen der letzten Zeile hervorgeht.

Satz 1.7 *Der Arnoldi-Algorithmus breche nicht vor der Konstruktion von p_{m+1} ab. Dann gilt:*

$$AP_m = P_{m+1}\hat{H}_m,$$
$$P_m^* A P_m = H_m.$$

Beweis. Dieser Beweis basiert auf den Beweisen von Satz 4.80/4.81 aus [5, S.167-168].
Nach Zeile 6 und 11 des Arnoldi-Algorithmus gilt für alle $j \in \{1, ..., m\}$

$$AP_m e_j = Ap_j = \left[\sum_{i=1}^{j} h_{ij}p_i\right] + w_j = \left[\sum_{i=1}^{j} h_{ij}p_i\right] + h_{j+1,j}p_{j+1} = \sum_{i=1}^{j+1} h_{ij}p_i = P_{m+1}\hat{H}_m e_j.$$

Also gilt $AP_m = P_{m+1}\hat{H}_m$. Daraus und mit Satz 1.6 folgt für alle $i, j \in \{1, ..., m\}$

$$(P_m^* A P_m)_{ij} = \langle p_i, Ap_j \rangle = \left\langle p_i, \sum_{k=1}^{j+1} h_{kj} p_k \right\rangle = \sum_{k=1}^{j+1} h_{kj} \langle p_i, p_k \rangle = \begin{cases} h_{ij}, & falls\ \ i \leq j+1 \\ 0, & sonst \end{cases}$$

und damit $P_m^* A P_m = H_m$.

□

Bemerkung 1.8 *Falls das Arnoldi-Verfahren im m-ten Schritt abbricht, also $w_m = 0$ ist, gilt analog:*

$$AP_m = P_m H_m,$$
$$P_m^* A P_m = H_m.$$

Diese Resultate können wir nun verwenden, um unseren gesuchten Vektor $x^{(m)} \in x^{(0)} + K_m(A, r^{(0)})$ mit (1.3) zu konstruieren. Dazu gehen wir zunächst davon aus, dass der Arnoldi-Algorithmus nicht vor der Konstruktion von p_{m+1} abbricht.
Wir wählen für den ersten Schritt im Arnoldi-Algorithmus $p_1 = \frac{r^{(0)}}{||r^{(0)}||}$ und erhalten nach Satz 1.6 eine Orthonormalbasis $\{p_1, ..., p_m\}$ von $K_m(A, \frac{r^{(0)}}{||r^{(0)}||}) = K_m(A, r^{(0)})$. Also existiert für $x \in x^{(0)} + K_m(A, r^{(0)})$ ein $y \in \mathbb{C}^m$ mit $x = x^{(0)} + P_m y$. Mit Satz 1.7 und unserer Wahl von p_1 folgt somit

$$\begin{aligned} b - Ax = b - A(x^{(0)} + P_m y) &= b - Ax^{(0)} - AP_m y = r^{(0)} - P_{m+1}\hat{H}_m y \\ &= ||r^{(0)}||p_1 - P_{m+1}\hat{H}_m y = P_{m+1}(||r^{(0)}||e_1 - \hat{H}_m y). \end{aligned}$$

Da P_{m+1} aufgrund der Orthonormalität von $p_1, ..., p_{m+1}$ isometrisch ist, folgt nun

$$||b - Ax|| = \left|\left|P_{m+1}(||r^{(0)}||e_1 - \hat{H}_m y)\right|\right| = \left|\left|||r^{(0)}||e_1 - \hat{H}_m y\right|\right|. \tag{1.7}$$

Statt $||b - Ax||$ über $x \in x^{(0)} + K_m(A, r^{(0)})$ zu minimieren, können wir also $y^{(m)} \in \mathbb{C}^m$ mit

$$\left|\left|||r^{(0)}||e_1 - \hat{H}_m y^{(m)}\right|\right| = min\left\{ \left|\left|||r^{(0)}||e_1 - \hat{H}_m y\right|\right| \;\middle|\; y \in \mathbb{C}^m \right\} \tag{1.8}$$

berechnen und anschließend unser $x^{(m)}$ mit (1.3) vermöge

$$x^{(m)} := x^{(0)} + P_m y^{(m)} \tag{1.9}$$

konstruieren. Dies machen wir wie folgt:
Wir betrachten das lineare Gleichungssystem $\hat{H}_m y = ||r^{(0)}||e_1$ und nutzen aus, dass für jede unitäre Matrix $Q \in \mathbb{C}^{(m+1)\times(m+1)}$ gilt:

$$\left|\left|Q||r^{(0)}||e_1 - Q\hat{H}_m y)\right|\right| = \left|\left|Q(||r^{(0)}||e_1 - \hat{H}_m y)\right|\right| = \left|\left|||r^{(0)}||e_1 - \hat{H}_m y\right|\right|,$$

sodass der Fehler durch Multiplikation mit Q also unverändert bleibt.

Wir verwenden nun spezielle, unitäre Matrizen, sogenannte *Givens-Rotationen*, um dieses Gleichungssystem zu transformieren. Hierzu definieren wir zunächst

$$\Omega_0 := I_{m+1}, \;\; c_0 := 1, \;\; s_0 := 0.$$

Dann definieren wir induktiv für $i \in \{1, ..., m\}$

$$t_i := (\Omega_{i-1}\Omega_{i-2}...\Omega_0\hat{H}_m)_{ii}$$

und darauf basierend die i-ten *Givens-Koeffizienten*

$$c_i := \frac{\overline{t_i}}{\sqrt{|t_i|^2 + |h_{i+1,i}|^2}}, \qquad s_i := \frac{h_{i+1,i}}{\sqrt{|t_i|^2 + |h_{i+1,i}|^2}}$$

und die i-te *Givens-Rotation* $\Omega_i \in \mathbb{C}^{(m+1)\times(m+1)}$ durch

$$\Omega_i := \begin{array}{c} \\ \\ i \\ i+1 \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccccc} & & & i & i+1 & & & \end{array} \\ \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & c_i & \overline{s_i} & & & \\ & & & -s_i & \overline{c_i} & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix} \end{array}.$$

Wegen

$$|c_i|^2 + |s_i|^2 = \frac{|t_i|^2 + |h_{i+1,i}|^2}{|t_i|^2 + |h_{i+1,i}|^2} = 1 \quad \text{und} \quad c_i s_i - s_i c_i = \overline{c_i s_i} - \overline{s_i c_i} = 0$$

gilt $\Omega_i^* \Omega_i = I_{m+1}$. Also ist Ω_i unitär. Außerdem gilt

$$(\Omega_i\Omega_{i-1}...\Omega_0\hat{H}_m)_{ii} = c_i t_i + \overline{s_i} h_{i+1,i} = \frac{|t_i|^2 + |h_{i+1,i}|^2}{\sqrt{|t_i|^2 + |h_{i+1,i}|^2}} = \sqrt{|t_i|^2 + |h_{i+1,i}|^2},$$

$$(\Omega_i\Omega_{i-1}...\Omega_0\hat{H}_m)_{i+1,i} = -s_i t_i + \overline{c_i} h_{i+1,i} = \frac{t_i h_{i+1,i} - t_i h_{i+1,i}}{\sqrt{|t_i|^2 + |h_{i+1,i}|^2}} = 0.$$

Durch sukzessives Multiplizieren mit den unitären Givens-Rotationen $\Omega_1, ..., \Omega_m$, also mit der unitären Matrix

$$Q_m := \Omega_m\Omega_{m-1}...\Omega_1 \in \mathbb{C}^{(m+1)\times(m+1)},$$

können wir $\hat{H}_m$ also in eine obere Dreiecksmatrix

$$\hat{R}_m := (r_{ij})_{i\in\{1,...,m+1\}, j\in\{1,...,m\}} := Q_m\hat{H}_m \in \mathbb{C}^{(m+1)\times m}$$

transformieren.

$$\begin{pmatrix} h_{11} & h_{12} & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & h_{2m} \\ 0 & h_{32} & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_{mm} \\ 0 & \dots & 0 & h_{m+1,m} \end{pmatrix} \xrightarrow{Q_m} \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ 0 & r_{22} & \dots & r_{2m} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_{mm} \\ 0 & \dots & 0 & 0 \end{pmatrix}$$

Ebenso multiplizieren wir die rechte Seite $||r^{(0)}||e_1$ mit diesen Givens-Rotationen und erhalten den vollbesetzten Vektor

$$\hat{g}^{(m)} := Q_m||r^{(0)}||e_1 \in \mathbb{C}^{m+1},$$

sodass wir nun das lineare Gleichungssystem

$$\hat{R}_m y = \hat{g}^{(m)}$$

betrachten können.

Notation 1.9 *Im Folgenden definieren wir*

- $R_m \in \mathbb{C}^{m\times m}$ *als die Matrix, die aus* $\hat{R}_m$ *durch Streichen der letzten Zeile hervorgeht,*
- $g^{(m)} \in \mathbb{C}^m$ *als den Vektor* $\hat{g}^{(m)}$ *ohne dessen letzte Komponente* $\hat{g}^{(m)}_{m+1}$,
- $\gamma_{m+1} := \hat{g}^{(m)}_{m+1}$,
- $\hat{g}^{(0)} := (||r^{(0)}||) \in \mathbb{C}^1, \gamma_1 := ||r^{(0)}||$,
- $Q_0 := I_1 \in \mathbb{C}^{1\times 1}$.

Da wir davon ausgehen, dass der Arnoldi-Algorithmus nicht vor der Konstruktion von p_{m+1} abbricht, gilt $h_{i+1,i} \neq 0$ für alle $i \in \{1, ..., m\}$. Also gilt für alle $i \in \{1, ..., m\}$

$$(R_m)_{ii} = c_i t_i + \overline{s_i} h_{i+1,i} = \sqrt{|t_i|^2 + |h_{i+1,i}|^2} > 0.$$

Damit ist die obere Dreiecksmatrix R_m invertierbar und wir können das lineare Gleichungssystem

$$R_m y = g^{(m)}$$

exakt durch Rückwärtseinsetzen lösen. Da Q_m unitär ist, folgt also

$$\begin{aligned} min\left\{ \left|\left| ||r^{(0)}||e_1 - \hat{H}_m y \right|\right| \;\middle|\; y \in \mathbb{C}^m \right\} &= min\left\{ \left|\left| Q_m(||r^{(0)}||e_1 - \hat{H}_m y) \right|\right| \;\middle|\; y \in \mathbb{C}^m \right\} \\ = min\left\{ \left|\left| \hat{g}^{(m)} - \hat{R}_m y \right|\right| \;\middle|\; y \in \mathbb{C}^m \right\} &= min\left\{ \left|\left| \begin{pmatrix} g^{(m)} \\ \gamma_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \end{pmatrix} y \right|\right| \;\middle|\; y \in \mathbb{C}^m \right\} \\ = min\left\{ \left|\left| \begin{pmatrix} g^{(m)} \\ \gamma_{m+1} \end{pmatrix} - \begin{pmatrix} R_m y \\ 0 \end{pmatrix} \right|\right| \;\middle|\; y \in \mathbb{C}^m \right\} &= min\left\{ \left|\left| \begin{pmatrix} g^{(m)} - R_m y \\ \gamma_{m+1} \end{pmatrix} \right|\right| \;\middle|\; y \in \mathbb{C}^m \right\}. \end{aligned} \tag{1.10}$$

Für unser gesuchtes $y^{(m)} \in \mathbb{C}^m$ mit (1.8) gilt somit gerade

$$y^{(m)} = R_m^{-1} g^{(m)}.$$

Zusammen mit (1.9) erhalten wir für die gesuchte Näherungslösung nun die Darstellung

$$x^{(m)} = x^{(0)} + P_m y^{(m)} = x^{(0)} + P_m R_m^{-1} g^{(m)}.$$

Dabei gilt für die Norm des korrespondierenden Residuums nach (1.7) und (1.10)

$$||b - Ax^{(m)}|| = \left|\left| ||r^{(0)}|| e_1 - \hat{H}_m y^{(m)} \right|\right| = \left|\left| \begin{pmatrix} g^{(m)} - R_m y^{(m)} \\ \gamma_{m+1} \end{pmatrix} \right|\right| = |\gamma_{m+1}|. \tag{1.11}$$

Bei der Durchführung des GMRES-Verfahrens gehen wir induktiv vor:
Angenommen, $\hat{R}_{m-1}$ und $\hat{g}^{(m-1)}$ seien bereits konstruiert.[1] Zunächst fügen wir unten an $\hat{R}_{m-1}$ eine Nullzeile und unten an $\hat{g}^{(m-1)}$ eine Null an. Dann konstruieren wir durch den m-ten Schritt des Arnoldi-Verfahrens die m-te Spalte von $\hat{H}_m$ und fügen diese rechts an $\hat{R}_{m-1}$ an. Dann wenden wir die bisher verwendeten Givens-Rotationen auf diese neue Spalte an. Anschließend konstruieren wir die neue Rotation Ω_m und wenden diese sowohl auf die neue Spalte, als auch auf die rechte Seite $\begin{pmatrix} \hat{g}^{(m-1)} \\ 0 \end{pmatrix}$ an, um $\hat{R}_m$ und $\hat{g}_m$ zu konstruieren. Dann gilt

$$\hat{g}^{(m)} = \Omega_m \begin{pmatrix} \hat{g}^{(m-1)} \\ 0 \end{pmatrix} = \Omega_m \begin{pmatrix} g^{(m-1)} \\ \gamma_m \\ 0 \end{pmatrix} = \begin{pmatrix} g^{(m-1)} \\ c_m \gamma_m \\ -s_m \gamma_m \end{pmatrix},^2$$

also insbesondere

$$\gamma_{m+1} = -s_m \gamma_m. \tag{1.12}$$

Da wir den verbleibenden Fehler auf diese Weise mit sehr geringem Aufwand ohne explizite Berechnung der korrespondierenden Näherungslösung erhalten, können wir, sofern der Arnoldi-Algorithmus nicht vorzeitig abbricht, diesen Ablauf so lange wiederholen, bis wir mit dem Fehler $|\gamma_{m+1}|$ zufrieden sind und anschließend die Näherungslösung $x^{(m)}$ wie beschrieben durch Rückwärtseinsetzen und Verwendung der konstruierten Basisvektoren berechnen.
Um Skalierungsinvarianz zu erhalten, wählen wir hierfür eine Fehlerschranke $\epsilon \in \mathbb{R}_{>0}$ und brechen ab, sobald die relative Residuennorm diese Schranke unterschreitet, d.h. sobald

$$\frac{||r^{(m)}||}{||r^{(0)}||} = \frac{||b - Ax^{(m)}||}{||r^{(0)}||} = \frac{|\gamma_{m+1}|}{|\gamma_1|} < \epsilon$$

ist. Darauf basierend können wir nun das GMRES-Verfahren skizzieren.
Es bleibt zu untersuchen, was passiert, wenn der Arnoldi-Algorithmus im m-ten Schritt abbricht, d.h. wenn $h_{m+1,m} = 0$ ist, da genau dann auch das GMRES-Verfahren abbricht.

[1] $\hat{R}_0$ interpretieren wir hierbei als nichts.
[2] $g^{(0)}$ interpretieren wir ebenfalls als nichts.

Algorithmus 1.10 ***(GMRES)***

1: $r^{(0)} \leftarrow b - Ax^{(0)}$;
2: $\gamma_1 \leftarrow ||r^{(0)}||$;
3: $p_1 \leftarrow r^{(0)}/\gamma_1$;
4: $m \leftarrow 0$;
5: **while** $|\gamma_{m+1}| > |\gamma_1|\epsilon$ **do**
6: $\quad m \leftarrow m + 1$;
7: $\quad$ Berechne $h_{1m}, h_{2m}, ..., h_{m+1,m}$ mit dem Arnoldi-Algorithmus;
8: $\quad$ Wende $\Omega_1, \Omega_2, ..., \Omega_{m-1}$ auf die m-te Spalte von $\hat{H}_m$ an;
9: $\quad$ Berechne die Givens-Koeffizienten c_m und s_m;
10: $\quad \gamma_{m+1} \leftarrow -s_m\gamma_m$;
11: $\quad \gamma_m \leftarrow c_m\gamma_m$;
12: $\quad r_{mm} \leftarrow c_m t_m + \overline{s_m} h_{m+1,m} = \sqrt{|t_m|^2 + |h_{m+1,m}|^2}$;
13: **end while**
14: Berechne $y^{(m)} = R_m^{-1} g^{(m)}$ durch Rückwärtseinsetzen;
15: $x^{(m)} \leftarrow x^{(0)} + P_m y^{(m)}$;

Satz 1.11 (Lucky Breakdown) *Falls das Arnoldi-Verfahren im m-ten Schritt abbricht, kann der m-te Schritt des GMRES-Verfahrens dennoch durchgeführt werden. In diesem Fall ist $m = m_0$ und $x^{(m)}$ sogar die exakte Lösung.*

Beweis. Dieser Beweis basiert auf [1, S.106-107] und den Beweisen von Lemma 3.32 aus [1, S.111] und Proposition 6.10 aus [6, S.179].

Da das Arnoldi-Verfahren im m-ten Schritt abbricht, gilt

$$0 = h_{m+1,m} = ||w_m|| = \left|\left| Ap_m - \sum_{i=1}^{m} h_{im} p_i \right|\right|$$

und damit

$$Ap_m = \sum_{i=1}^{m} h_{im} p_i \in span\{p_1, ..., p_m\} = K_m(A, p_1).$$

Wegen $A^{m-1}p_1 \in K_m(A, p_1)$ existieren außerdem $\alpha_1, ..., \alpha_m \in \mathbb{C}$ mit $A^{m-1}p_1 = \sum_{i=1}^{m} \alpha_i p_i$.

Damit gilt $A^m p_1 = \sum_{i=1}^{m} \alpha_i A p_i$ und somit

$$A^m p_1 - \alpha_m A p_m = \sum_{i=1}^{m-1} \alpha_i A p_i \in K_m(A, p_1).$$

Damit folgt nun mit dem Basisaustauschsatz

$$\begin{aligned} K_{m+1}(A, p_1) = span\{p_1, Ap_1, ..., A^{m-1}p_1, A^m p_1\} &= span\{p_1, p_2, ..., p_m, A^m p_1\} \\ &= span\{p_1, p_2, ..., p_m, Ap_m\} \subseteq K_m(A, p_1) \end{aligned}$$

und damit $K_{m+1}(A, p_1) = K_m(A, p_1)$. Also ist $m = m_0$.
Sei nun $y \in \mathbb{C}^m$ mit $H_m y = 0$. Dann gilt für alle $t \in \mathbb{C}^m$ mit Bemerkung 1.8

$$0 = \langle t, H_m y \rangle = \langle t, P_m^* A P_m y \rangle = \langle P_m t, A P_m y \rangle.$$

Wegen $AP_m y \in K_{m+1}(A, p_1) = K_m(A, p_1) = Bild(P_m)$ existiert ein $\hat{t} \in \mathbb{C}^m$ mit

$$P_m \hat{t} = A P_m y.$$

Also folgt

$$0 = \langle P_m \hat{t}, A P_m y \rangle = \langle A P_m y, A P_m y \rangle = ||A P_m y||^2$$

und damit $AP_m y = 0$. Da $p_1, ..., p_m$ linear unabhängig sind, ist P_m injektiv. Außerdem ist A nach Voraussetzung injektiv und somit ist ebenfalls AP_m injektiv. Damit folgt $y = 0$. H_m ist also injektiv und als quadratische Matrix somit invertierbar.
Da $Q_{m-1} \in \mathbb{C}^{m \times m}$ unitär ist, ist somit auch die obere Dreiecksmatrix $Q_{m-1} H_m$ invertierbar. Damit muss $t_m = (Q_{m-1} H_m)_{mm} \neq 0$ gelten. Die m-te Givens-Rotation Ω_m kann also definiert werden mit

$$s_m = \frac{h_{m+1,m}}{\sqrt{|t_m|^2 + |h_{m+1,m}|^2}} = 0$$

und die obere Dreiecksmatrix R_m ist wegen

$$(R_m)_{mm} = \sqrt{|t_m|^2 + |h_{m+1,m}|^2} = \sqrt{|t_m|^2} = |t_m| > 0$$

auch in diesem Fall regulär.
Für ein beliebiges $x = x^{(0)} + P_m y \in x^{(0)} + K_m(A, r^{(0)})$ gilt nach Bemerkung 1.8 außerdem

$$\begin{aligned} b - Ax = b - A(x^{(0)} + P_m y) &= b - Ax^{(0)} - AP_m y = r^{(0)} - P_m H_m y \\ &= ||r^{(0)}||p_1 - P_m H_m y = P_m(||r^{(0)}||e_1 - H_m y) \end{aligned}$$

und wegen der Orthonormalität von $p_1, ..., p_m$ somit

$$||b - Ax|| = \left|\left|P_m(||r^{(0)}||e_1 - H_m y)\right|\right| = \left|\left||r^{(0)}||e_1 - H_m y\right|\right|.$$

Weiter gilt nach Voraussetzung $\hat{H}_m = \begin{pmatrix} H_m \\ 0 \dots 0 \;\; h_{m+1,m} \end{pmatrix} = \begin{pmatrix} H_m \\ 0 \end{pmatrix}$ und daher

$$\left|\left||r^{(0)}||e_1 - H_m y\right|\right| = \left|\left||r^{(0)}||e_1 - \hat{H}_m y\right|\right| \quad \forall y \in \mathbb{C}^m.$$

Für unsere Näherungslösung $x^{(m)} = x^{(0)} + P_m y^{(m)} = x^{(0)} + P_m R_m^{-1} g^{(m)}$ gilt also auch in diesem Fall wie zuvor mit (1.11)

$$||b - Ax^{(m)}|| = \left|\left||r^{(0)}||e_1 - H_m y^{(m)}\right|\right| = \left|\left||r^{(0)}||e_1 - \hat{H}_m y^{(m)}\right|\right| = |\gamma_{m+1}|.$$

Für den verbleibenden Fehler gilt also wegen $s_m = 0$ und (1.12) gerade

$$||b - Ax^{(m)}|| = |\gamma_{m+1}| = |-s_m \gamma_m| = 0.$$

Damit ist $x^{(m)}$ die exakte Lösung.

□

Wie wir bereits eingesehen haben, gilt für die GMRES-Residuen

$$||r^{(0)}|| \geq ||r^{(1)}|| \geq ... \geq ||r^{(m_0)}|| = 0.$$

Genauer gesagt gilt das folgende Resultat.

Korollar 1.12 *Für das m-te GMRES-Residuum* $r^{(m)} = b - Ax^{(m)}$ *gilt*

$$||r^{(m)}|| = |s_1 s_2 ... s_m| \ ||r^{(0)}||.$$

Beweis. Dieser Beweis basiert auf dem Beweis von Propositon 7.3 aus [6, S.238]. Durch sukzessive Anwendung von (1.12) erhalten wir wegen $\gamma_1 = ||r^{(0)}||$

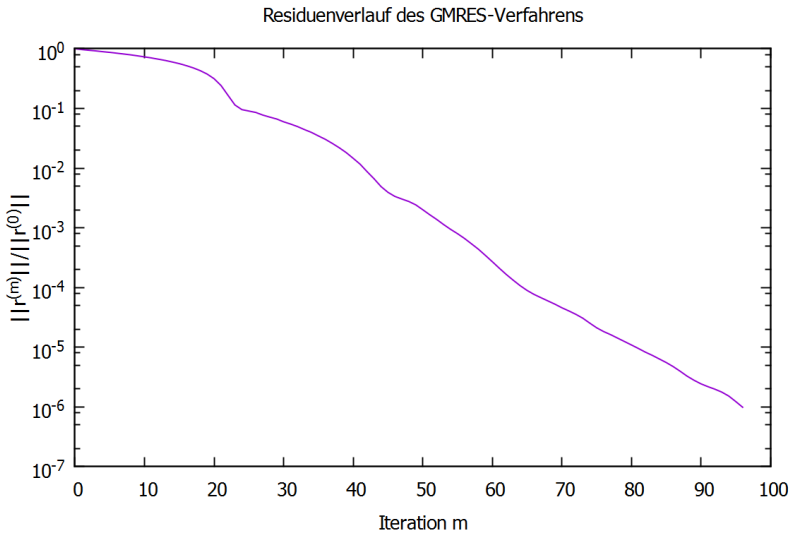
$$|\gamma_{m+1}| = |(-1)^m s_1 ... s_m \gamma_1| = |s_1 ... s_m \ ||r^{(0)}|| \ | = |s_1 s_2 ... s_m| \ ||r^{(0)}||.$$

Damit folgt mit (1.11) nun direkt

$$||r^{(m)}|| = ||b - Ax^{(m)}|| = |\gamma_{m+1}| = |s_1 s_2 ... s_m| \ ||r^{(0)}||.$$

□

Ein entsprechendes Bild liefert die Anwendung des GMRES-Verfahrens auf unser Modellproblem aus Abschnitt 1.3.



Tatsächlich lässt sich keine stärkere Aussage bezüglich des Residuenverlaufs des GMRES-Verfahrens treffen, ohne mehr als lediglich die Regularität der Matrix A zu fordern. Es gibt sogar Beispiele von linearen Gleichungssystemen, bei denen der Startfehler durch die ersten $n - 1$ Schritte des GMRES-Verfahrens unverändert bleibt, bevor im n-ten Schritt plötzlich die exakte Lösung berechnet wird, siehe [1, S.121-122].

2 Das QMR-Verfahren

Das GMRES-Verfahren hat ohne Zweifel viele Vorteile: Es funktioniert für jedes lineare Gleichungssystem mit einer regulären Matrix, ohne weitere Voraussetzungen, wird mit jeder Iteration genauer und berechnet nach spätestens n Iterationen die exakte Lösung. Allerdings hat es auch signifikante Nachteile:

- In Zeile 6 des Arnoldi-Algorithmus wird eine Gram-Schmidt-Orthogonalisierung durchgeführt, um sicherzustellen, dass der neu konstruierte Vektor senkrecht auf allen zuvor konstruierten Vektoren steht. Dafür werden natürlich alle zuvor konstruierten Vektoren benötigt. Für jeden Schritt des GMRES-Verfahrens muss also ein weiterer, in der Regel vollbesetzter Vektor des $\mathbb{C}^n$ gespeichert werden.
- Die im m-ten Schritt konstruierte Matrix $\hat{H}_m$ ist eine Hessenberg-Matrix. Daher müssen alle zuvor konstruierten Givens-Rotationen auf die vollbesetzte, neue Spalte angewendet und somit natürlich auch gespeichert werden. Ebenso wächst der Speicherbedarf für die neue Spalte mit jeder Iteration.
- Vor allem durch den Arnoldi-Algorithmus, aber auch durch die Anwendung der Givens-Rotationen und das Lösen des Gleichungssystems $R_m y = g^{(m)}$ durch Rückwärtseinsetzen steigt die Anzahl der benötigten Rechenoperationen für die Berechnung der m-ten Näherungslösung mit jeder Iteration stark an.

Sowohl der Speicherbedarf als auch der Rechenaufwand des GMRES-Verfahrens wachsen somit erheblich mit der Anzahl der durchgeführten Schritte.
Die Konstruktion einer orthonormalen Basis von $K_m(A, r^{(0)})$ muss also teuer erkauft werden: Falls die Problemdimension n sehr groß ist, kann das GMRES-Verfahren außerordentlich aufwändig oder sogar undurchführbar werden.
Die Idee des *QMR-Verfahrens (quasi-minimal residual method)* besteht nun darin, den Anstieg des Speicherbedarfs und des Rechenaufwands zu vermeiden, indem statt einer orthonormalen Basis eine im Allgemeinen nicht orthonormale Basis der Krylow-Räume verwendet wird.

Das Vorgehen in diesem Kapitel orientiert sich hauptsächlich an [6] und [5].

2.1 Der Bi-Lanczos-Algorithmus

Das QMR-Verfahren verwendet zur Konstruktion von Basen der Krylow-Räume als Alternative zur Arnoldi-Methode den sogenannten *Bi-Lanczos-Algorithmus*. Sei $m \in \mathbb{N}$.

Algorithmus 2.1 ***(Bi-Lanczos)***

1: Wähle zwei Vektoren $v_1, w_1 \in \mathbb{C}^n$ mit $\langle v_1, w_1 \rangle = 1$;
2: $\beta_1 \leftarrow 0, \delta_1 \leftarrow 0$;
3: $w_0 \leftarrow 0, v_0 \leftarrow 0$;
4: **for** $j = 1, 2, ..., m$ **do**
5: $\quad \alpha_j \leftarrow \langle Av_j, w_j \rangle$;
6: $\quad \hat{v}_{j+1} \leftarrow Av_j - \overline{\alpha_j} v_j - \overline{\beta_j} v_{j-1}$;
7: $\quad \hat{w}_{j+1} \leftarrow A^* w_j - \alpha_j w_j - \delta_j w_{j-1}$;
8: $\quad \delta_{j+1} \leftarrow |\langle \hat{v}_{j+1}, \hat{w}_{j+1} \rangle|^{1/2}$;
9: $\quad$ **if** $\delta_{j+1} = 0$ **then**
10: $\quad\quad$ stop;
11: $\quad$ **end if**
12: $\quad \beta_{j+1} \leftarrow \langle \hat{v}_{j+1}, \hat{w}_{j+1} \rangle / \delta_{j+1}$;
13: $\quad w_{j+1} \leftarrow \hat{w}_{j+1} / \beta_{j+1}$;
14: $\quad v_{j+1} \leftarrow \hat{v}_{j+1} / \overline{\delta_{j+1}}$;
15: **end for**

Sofern dieser Algorithmus nicht vor der Konstruktion von v_m und w_m abbricht, stehen die Vektoren $v_1, ..., v_m$ in einer besonderen Beziehung zu den Vektoren $w_1, ..., w_m$.

Satz 2.2 (Bi-Orthonormalität) *Angenommen, der Bi-Lanczos-Algorithmus breche nicht vor der Konstruktion von v_m und w_m ab. Dann gilt:*

$$\langle v_i, w_j \rangle = \delta_{ij} \qquad \forall i, j \in \{1, ..., m\}.$$

Die Vektoren $v_1, ..., v_m$ und $w_1, ..., w_m$ sind also biorthonormal.

Beweis. Dieser Beweis basiert auf dem Beweis von Proposition 7.1 aus [6, S.230-231].
Wir zeigen die Behauptung per Induktion.
Induktionsanfang:
Nach Wahl von v_1, w_1 gilt $\langle v_1, w_1 \rangle = 1$.
Induktionsvoraussetzung:
Sei also $k \in \{1, ..., m-1\}$ mit $\langle v_i, w_j \rangle = \delta_{ij}$ $\forall i, j \in \{1, ..., k\}$.
Induktionsschritt:
Nach Konstruktion gilt bereits

$$\langle v_{k+1}, w_{k+1} \rangle = \Big\langle \frac{\hat{v}_{k+1}}{\overline{\delta_{k+1}}}, \frac{\hat{w}_{k+1}}{\beta_{k+1}} \Big\rangle = \frac{\langle \hat{v}_{k+1}, \hat{w}_{k+1} \rangle}{\delta_{k+1} \beta_{k+1}} = \frac{\langle \hat{v}_{k+1}, \hat{w}_{k+1} \rangle}{\langle \hat{v}_{k+1}, \hat{w}_{k+1} \rangle} = 1.$$

Wir wollen nun zeigen, dass $\langle v_{k+1}, w_i \rangle = 0$ für alle $i \in \{1, ..., k\}$ gilt. Der Nachweis, dass $\langle v_i, w_{k+1} \rangle = 0$ für alle $i \in \{1, ..., k\}$ gilt, kann analog erbracht werden.
Es gilt nach der Induktionsvoraussetzung und mit Z.5, 6 und 14 des Bi-Lanczos-Algortihmus

$$\langle v_{k+1}, w_k \rangle = \frac{1}{\delta_{k+1}} \langle \hat{v}_{k+1}, w_k \rangle = \frac{1}{\delta_{k+1}} (\langle Av_k, w_k \rangle - \alpha_k \langle v_k, w_k \rangle - \beta_k \langle v_{k-1}, w_k \rangle) = \frac{1}{\delta_{k+1}} (\alpha_k - \alpha_k) = 0.$$

Wenn also $k = 1$ ist, folgt $\langle v_{k+1}, w_i\rangle = 0$ für alle $i \in \{1, ..., k\}$. Wir nehmen nun also an, dass $k \geq 2$ ist. Dann gilt für $i \in \{1, ..., k-1\}$ mit der Induktionsvoraussetzung und Z.6, 7, 13 und 14 des Bi-Lanczos-Algorithmus

$$\langle v_{k+1}, w_i\rangle = \frac{1}{\delta_{k+1}}\langle \hat{v}_{k+1}, w_i\rangle = \frac{1}{\delta_{k+1}}(\langle Av_k, w_i\rangle - \alpha_k\langle v_k, w_i\rangle - \beta_k\langle v_{k-1}, w_i\rangle)$$
$$= \frac{1}{\delta_{k+1}}(\langle v_k, A^*w_i\rangle - \beta_k\langle v_{k-1}, w_i\rangle) = \frac{1}{\delta_{k+1}}(\langle v_k, \beta_{i+1}w_{i+1} + \alpha_i w_i + \delta_i w_{i-1}\rangle - \beta_k\langle v_{k-1}, w_i\rangle). \tag{2.1}$$

Also folgt mit der Induktionsvoraussetzung insbesondere

$$\begin{aligned}\langle v_{k+1}, w_{k-1}\rangle &= \frac{1}{\delta_{k+1}}(\langle v_k, \beta_k w_k + \alpha_{k-1}w_{k-1} + \delta_{k-1}w_{k-2}\rangle - \beta_k\langle v_{k-1}, w_{k-1}\rangle)\\ &= \frac{1}{\delta_{k+1}}(\beta_k\langle v_k, w_k\rangle - \beta_k\langle v_{k-1}, w_{k-1}\rangle) = \frac{1}{\delta_{k+1}}(\beta_k - \beta_k) = 0.\end{aligned}$$

Wenn also $k = 2$ ist, folgt $\langle v_{k+1}, w_i\rangle = 0$ für alle $i \in \{1, ..., k\}$. Falls $k \geq 3$ ist, folgt mit (2.1) und der Induktionsvoraussetzung außerdem, dass für alle $i \in \{1, ..., k-2\}$ gerade $\langle v_{k+1}, w_i\rangle = 0$ gilt. Also gilt auch in diesem Fall $\langle v_{k+1}, w_i\rangle = 0$ für alle $i \in \{1, ..., k\}$.

□

Diese Eigenschaft können wir nun verwenden, um nachzuweisen, dass der Bi-Lanczos-Algorithmus sogar Basen für zwei Krylow-Räume konstruiert.

Satz 2.3 *Angenommen, der Bi-Lanczos-Algorithmus breche nicht vor der Konstruktion von v_m und w_m ab. Dann ist $\{v_1, ..., v_m\}$ eine Basis von $K_m(A, v_1)$ und $\{w_1, ..., w_m\}$ eine Basis von $K_m(A^*, w_1)$.*

Beweis. Dieser Beweis ist eine Abwandlung des Beweises von Satz 1.6 und verwendet zusätzlich den Beweis von Lemma 4.83 aus [5, S.175].
Wir zeigen per Induktion, dass $\{v_1, ..., v_m\}$ eine linear unabhängige Teilmenge von $K_m(A, v_1)$ und somit wegen $dim(K_m(A, v_1)) \leq m$ eine Basis von $K_m(A, v_1)$ ist. Der Nachweis, dass $\{w_1, ..., w_m\}$ eine Basis von $K_m(A^*, w_1)$ ist, kann analog erbracht werden.
Induktionsanfang:
Es ist $\{v_1\}$ eine linear unabhängige Teilmenge von $span\{v_1\} = K_1(A, v_1)$.
Induktionsvoraussetzung:
Sei $i \in \{1, ..., m-1\}$ so gewählt, dass $\{v_1, ..., v_i\}$ eine linear unabhängige Teilmenge von $K_i(A, v_1)$ ist.
Induktionsschritt:
Nach der Induktionsvoraussetzung sind v_{i-1} und v_i Elemente von $K_i(A, v_1)$. Damit sind v_{i-1}, v_i und Av_i Elemente von $K_{i+1}(A, v_1)$. Mit Z.6 und 14 des Bi-Lanczos-Algorithmus folgt nun

$$v_{i+1} = \frac{\hat{v}^{(i+1)}}{\overline{\delta_{i+1}}} = \frac{Av_i - \overline{\alpha_i}v_i - \overline{\beta_i}v_{i-1}}{\overline{\delta_{i+1}}} \in K_{i+1}(A, v_1)$$

und damit $\{v_1, ..., v_{i+1}\} \subseteq K_{i+1}(A, v_1)$.
Seien nun $c_1, ..., c_{i+1} \in \mathbb{C}$ mit $\sum_{k=1}^{i+1} c_k v_k = 0$. Dann folgt nach Satz 2.2 für alle $j \in \{1, ..., i+1\}$ gerade

$$0 = \left\langle \sum_{k=1}^{i+1} c_k v_k, w_j \right\rangle = \sum_{k=1}^{i+1} \overline{c_k} \langle v_k, w_j \rangle = \overline{c_j} \langle v_j, w_j \rangle = \overline{c_j}.$$

Damit folgt $c_1 = c_2 = ... = c_i = c_{i+1} = 0$. Also ist $\{v_1, ..., v_{i+1}\}$ eine linear unabhängige Teilmenge von $K_{i+1}(A, v_1)$.

□

Sofern der Bi-Lanczos-Algorithmus nicht vor der Konstruktion von v_m und w_m abbricht, geht aus dessen Durchführung die Tridiagonalmatrix

$$T_m := \begin{pmatrix} \overline{\alpha_1} & \overline{\beta_2} & & & \\ \overline{\delta_2} & \overline{\alpha_2} & \overline{\beta_3} & & \\ & \ddots & \ddots & \ddots & \\ & & \overline{\delta_{m-1}} & \overline{\alpha_{m-1}} & \overline{\beta_m} \\ & & & \overline{\delta_m} & \overline{\alpha_m} \end{pmatrix} \in \mathbb{C}^{m \times m}$$

hervor. Außerdem können dann die Matrizen

$$V_m := (v_1 ... v_m) \in \mathbb{C}^{n \times m}, \quad W_m := (w_1 ... w_m) \in \mathbb{C}^{n \times m}$$

definiert werden. Mit diesen Matrizen können wir nun ein weiteres, entscheidendes Resultat für das QMR-Verfahren beweisen.

Satz 2.4 *Der Bi-Lanczos-Algorithmus breche nicht vor der Berechnung von v_{m+1} und w_{m+1} ab. Dann gilt:*

$$AV_m = V_m T_m + \overline{\delta_{m+1}} v_{m+1} e_m^T,$$
$$A^* W_m = W_m T_m^* + \beta_{m+1} w_{m+1} e_m^T,$$
$$W_m^* A V_m = T_m.$$

Beweis. Dieser Beweis basiert auf dem Beweis von Satz 4.85 aus [5, S.178].
Nach Z.6 und 14 des Bi-Lanczos-Algorithmus gilt für alle $j \in \{1, ..., m-1\}$

$$AV_m e_j = Av_j = \hat{v}_{j+1} + \overline{\alpha_j} v_j + \overline{\beta_j} v_{j-1} = \overline{\delta_{j+1}} v_{j+1} + \overline{\alpha_j} v_j + \overline{\beta_j} v_{j-1} = V_m T_m e_j.$$

Ebenso ist nach Z.6 und 14

$$\begin{aligned} AV_m e_m = Av_m = \hat{v}_{m+1} + \overline{\alpha_m} v_m + \overline{\beta_m} v_{m-1} &= \overline{\delta_{m+1}} v_{m+1} + \overline{\alpha_m} v_m + \overline{\beta_m} v_{m-1} \\ &= V_m T_m e_m + \overline{\delta_{m+1}} v_{m+1}. \end{aligned}$$

Damit folgt $AV_m = V_m T_m + \overline{\delta_{m+1}} v_{m+1} e_m^T$. Analog gilt $A^* W_m = W_m T_m^* + \beta_{m+1} w_{m+1} e_m^T$. Außerdem folgt nun für $i, j \in \{1, ..., m\}$ mit Satz 2.2

$$(W_m^* A V_m)_{ij} = \langle w_i, A v_j \rangle = \langle w_i, \overline{\delta_{j+1}} v_{j+1} + \overline{\alpha_j} v_j + \overline{\beta_j} v_{j-1} \rangle = \begin{cases} \overline{\delta_{j+1}}, & falls \ \ i = j+1 \\ \overline{\alpha_j}, & falls \ \ i = j \\ \overline{\beta_j}, & falls \ \ i = j-1 \\ 0, & sonst \end{cases}$$

und damit $W_m^* A V_m = T_m$.

□

Die Vorteile des Bi-Lanczos-Algorithmus gegenüber der Arnoldi-Methode liegen auf der Hand: Während bei der Arnoldi-Methode wegen der Orthogonalisierung mit jedem Schritt ein weiterer Vektor des $\mathbb{C}^n$ zusätzlich abgespeichert werden muss, müssen für die Durchführung des Bi-Lanczos-Algorithmus nur 5 Vektoren gespeichert werden, unabhängig von der Anzahl der durchgeführten Schritte. Auch die Anzahl der benötigten Rechenoperationen ist in jedem Schritt gleich. Außerdem ist die Matrix T_m eine Tridiagonalmatrix im Gegensatz zu der Hessenberg-Matrix, die aus dem Arnoldi-Algorithmus hervorgeht. Die daraus resultierenden Vorteile werden anhand der nun folgenden Herleitung des QMR-Verfahrens ersichtlich.

2.2 Herleitung des Verfahrens

Für die Herleitung des QMR-Verfahrens wählen wir $v_1 := \frac{r^{(0)}}{||r^{(0)}||}$ und $w_1 := \frac{r^{(0)}}{||r^{(0)}||}$, um die Startbedingung für den Bi-Lanczos-Algorithmus

$$\langle v_1, w_1 \rangle = \frac{\langle r^{(0)}, r^{(0)} \rangle}{||r^{(0)}||^2} = 1$$

zu erfüllen und nehmen an, dass der Bi-Lanczos-Algorithmus nicht vor der Konstruktion von v_{m+1} und w_{m+1} abbricht. Wir definieren die Matrizen

$$\hat{T}_m := \begin{pmatrix} T_m \\ \overline{\delta_{m+1}} e_m^T \end{pmatrix} \in \mathbb{C}^{(m+1) \times m}, \ \ V_{m+1} := (v_1 ... v_{m+1}) \in \mathbb{C}^{n \times (m+1)}$$

und erhalten mit Satz 2.4

$$A V_m = V_m T_m + \overline{\delta_{m+1}} v_{m+1} e_m^T = V_{m+1} \hat{T}_m. \tag{2.2}$$

Nach Satz 2.3 ist $\{v_1, ..., v_m\}$ eine Basis von $K_m(A, v_1) = K_m(A, \frac{r^{(0)}}{||r^{(0)}||}) = K_m(A, r^{(0)})$. Für ein beliebiges $x \in x^{(0)} + K_m(A, r^{(0)})$ existiert also ein $y \in \mathbb{C}^m$ mit $x = x^{(0)} + V_m y$. Daher erhalten wir mit (2.2) und unserer Wahl von v_1 nun

$$b - Ax = b - A(x^{(0)} + V_m y) = r^{(0)} - A V_m y = ||r^{(0)}|| v_1 - V_{m+1} \hat{T}_m y = V_{m+1}(||r^{(0)}|| e_1 - \hat{T}_m y) \tag{2.3}$$

und somit

$$||b - Ax|| = \left|\left|V_{m+1}(||r^{(0)}||e_1 - \hat{T}_m y)\right|\right| \leq \left|\left|V_{m+1}\right|\right| \ \left|\left|||r^{(0)}||e_1 - \hat{T}_m y\right|\right|. \tag{2.4}$$

Unser Ziel ist es nun, $\left|\left|||r^{(0)}||e_1 - \hat{T}_m y\right|\right|$ über $y \in \mathbb{C}^m$ zu minimieren. Wir suchen also nach einem $y^{(m)} \in \mathbb{C}^m$, welches die Bedingung

$$\left|\left|||r^{(0)}||e_1 - \hat{T}_m y^{(m)}\right|\right| = min\left\{\left|\left|||r^{(0)}||e_1 - \hat{T}_m y\right|\right| \,\middle|\, y \in \mathbb{C}^m\right\} \tag{2.5}$$

erfüllt, um anschließend die *QMR-Approximation*

$$x^{(m)} := x^{(0)} + V_m y^{(m)}$$

zu konstruieren.
Diese Konstruktion ähnelt sehr dem Vorgehen bei dem GMRES-Verfahren. Allerdings entspricht $\left|\left|||r^{(0)}||e_1 - \hat{T}_m y^{(m)}\right|\right|$ nach (2.4) nicht exakt der minimalen Residuennorm, da die Spalten $v_1, ..., v_{m+1}$ von V_{m+1} im Gegensatz zu den im Arnoldi-Algorithmus konstruierten Basisvektoren im Allgemeinen nicht orthonormal sind. Durch die Berechnung von $\left|\left|||r^{(0)}||e_1 - \hat{T}_m y^{(m)}\right|\right|$ wird also, anders als bei GMRES, keine exakte Minimierung, sondern lediglich eine *Quasi-Minimierung* der Residuennorm durchgeführt. Dies rechtfertigt die folgende Definition.

Definition 2.5 (Quasi-Residuum) *Im Folgenden bezeichnen wir*

$$||r^{(0)}||e_1 - \hat{T}_m y^{(m)}$$

als das m-te Quasi-Residuum des QMR-Verfahrens.

Für die Konstruktion von $y^{(m)}$ betrachten wir das lineare Gleichungssystem

$$\hat{T}_m y = ||r^{(0)}||e_1.$$

$$\begin{pmatrix} \overline{\alpha_1} & \overline{\beta_2} & & & \\ \overline{\delta_2} & \overline{\alpha_2} & \overline{\beta_3} & & \\ & \ddots & \ddots & \ddots & \\ & & \overline{\delta_{m-1}} & \overline{\alpha_{m-1}} & \overline{\beta_m} \\ & & & \overline{\delta_m} & \overline{\alpha_m} \\ & & & & \overline{\delta_{m+1}} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} ||r^{(0)}|| \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$$

Dieses Gleichungssystem soll nun durch Multiplikation mit einer unitären Matrix in eine handlichere Form überführt werden, welche durch Rückwärtseinsetzen gelöst werden kann. Wie auch beim GMRES-Verfahren verwenden wir hierfür Givens-Rotationen.

Wir setzen also wieder

$$\Omega_0 := I_{m+1}, \;\; c_0 := 1, s_0 := 0$$

und definieren induktiv für $i \in \{1, ..., m\}$ zunächst

$$t_i := (\Omega_{i-1}\Omega_{i-2}...\Omega_0\hat{T}_m)_{ii}$$

und darauf basierend die i-te Givens-Rotation $\Omega_i \in \mathbb{C}^{(m+1)\times(m+1)}$ durch

$$\Omega_i := \begin{array}{c} \\ i \\ i+1 \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccccc} & & & & i & i+1 & & \end{array} \\ \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & & c_i & \overline{s_i} & & \\ & & & & -s_i & \overline{c_i} & & \\ & & & & & & 1 & \\ & & & & & & & \ddots \\ & & & & & & & & 1 \end{pmatrix} \end{array}$$

mit

$$c_i := \frac{\overline{t_i}}{\sqrt{|t_i|^2 + |\delta_{i+1}|^2}}, \qquad s_i := \frac{\overline{\delta_{i+1}}}{\sqrt{|t_i|^2 + |\delta_{i+1}|^2}}.$$

Wie wir in Kapitel 1 bereits gesehen haben, sind die Givens-Rotationen $\Omega_1, ..., \Omega_m$ und somit auch

$$Q_m := \Omega_m\Omega_{m-1}...\Omega_1 \in \mathbb{C}^{(m+1)\times(m+1)}$$

unitär und können verwendet werden, um die Matrix $\hat{T}_m$ schrittweise in die obere Dreiecksmatrix

$$\hat{R}_m := (r_{ij})_{i\in\{1,...,m+1\},j\in\{1,...,m\}} := (Q_m\hat{T}_m) \in \mathbb{C}^{(m+1)\times m}$$

zu transformieren.

$$\begin{pmatrix} \overline{\alpha_1} & \overline{\beta_2} & & & \\ \overline{\delta_2} & \overline{\alpha_2} & \overline{\beta_3} & & \\ & \ddots & \ddots & \ddots & \\ & & \overline{\delta_{m-1}} & \overline{\alpha_{m-1}} & \overline{\beta_m} \\ & & & \overline{\delta_m} & \overline{\alpha_m} \\ & & & & \overline{\delta_{m+1}} \end{pmatrix} \xrightarrow{Q_m} \begin{pmatrix} r_{11} & r_{12} & r_{13} & & & & \\ & r_{22} & r_{23} & r_{24} & & & \\ & & & \dots & & & \\ & & & r_{m-2,m-2} & r_{m-2,m-1} & r_{m-2,m} \\ & & & & r_{m-1,m-1} & r_{m-1,m} \\ & & & & & r_{mm} \\ & & & & & 0 \end{pmatrix}$$

Ebenso multiplizieren wir wieder die rechte Seite $||r^{(0)}||e_1$ des Gleichungssystems mit Q_m und erhalten so den vollbesetzten Vektor

$$\hat{g}^{(m)} := Q_m||r^{(0)}||e_1 \in \mathbb{C}^{m+1}.$$

Notation 2.6 *Im Folgenden definieren wir:*

- $R_m \in \mathbb{C}^{m\times m}$ *als die Matrix, die aus* $\hat{R}_m$ *durch Streichen der letzten Zeile hervorgeht,*
- $g^{(m)} \in \mathbb{C}^m$ *als den Vektor* $\hat{g}^{(m)}$ *ohne dessen letzte Komponente* $\hat{g}^{(m)}_{m+1}$,
- $\gamma_{m+1} := \hat{g}^{(m)}_{m+1}$,
- $\hat{g}^{(0)} := (||r^{(0)}||) \in \mathbb{C}^1, \gamma_1 := ||r^{(0)}||$,
- $Q_0 := I_1 \in \mathbb{C}^{1\times 1}$.

Da wir davon ausgehen, dass der Bi-Lanczos-Algorithmus nicht vor der Konstruktion von v_{m+1} und w_{m+1} abbricht, gilt $\delta_{i+1} \neq 0$ für alle $i \in \{1, ..., m\}$. Also gilt für alle $i \in \{1, ..., m\}$

$$(R_m)_{ii} = c_i t_i + \overline{s_i}\overline{\delta_{i+1}} = \sqrt{|t_i|^2 + |\delta_{i+1}|^2} > 0.$$

R_m ist also invertierbar und das transformierte Gleichungssystem

$$R_m y = g^{(m)}$$

kann exakt durch Rückwärtseinsetzen gelöst werden.
Mit der gleichen Argumentation wie bei GMRES in Kapitel 1 ergibt sich für unseren gesuchten Vektor $y^{(m)}$ mit (2.5) also gerade

$$y^{(m)} = R_m^{-1} g^{(m)}$$

und für die Norm des m-ten Quasi-Residuums folgt

$$\left|\left|\,||r^{(0)}||e_1 - \hat{T}_m y^{(m)}\right|\right| = ||\hat{g}^{(m)} - \hat{R}_m y^{(m)}|| = \left|\left| \begin{pmatrix} g^{(m)} - R_m y^{(m)} \\ \gamma_{m+1} \end{pmatrix} \right|\right| = |\gamma_{m+1}|. \quad (2.6)$$

Die Quasi-Minimierung verläuft also analog zu der Minimierung der Residuen-Norm im GMRES-Verfahren. Um unsere Iterierte $x^{(m)}$ analog zum GMRES-Verfahren direkt vermöge $x^{(m)} = x^{(0)} + V_m y^{(m)}$ zu berechnen, müssten wir allerdings alle konstruierten Basisvektoren $v_1, ..., v_m$ abspeichern und würden so einen der wesentlichen Vorteile des Bi-Lanczos-Algorithmus gegenüber der Arnoldi-Methode aufgeben. Anders als beim GMRES-Verfahren sollten wir $x^{(m)}$ also nicht erst dann berechnen, wenn wir mit $|\gamma_{m+1}|$ zufrieden sind. Wir müssen die Konstruktion von $x^{(m)}$ also entsprechend modifizieren.
Hierzu definieren wir die Matrix

$$P_m := V_m R_m^{-1} \in \mathbb{C}^{n\times m}$$

und bezeichnen dessen Spalten mit $p_1, ..., p_m$. Dann erhalten wir

$$x^{(m)} = x^{(0)} + V_m y^{(m)} = x^{(0)} + V_m R_m^{-1} g^{(m)} = x^{(0)} + P_m g^{(m)}. \quad (2.7)$$

Weiter gilt

$$P_m = V_m R_m^{-1} \Leftrightarrow P_m R_m = V_m \Leftrightarrow (P_m R_m)^T = V_m^T \Leftrightarrow R_m^T P_m^T = V_m^T.$$

Wenn wir also $P_m = (p_1 p_2 ... p_m)$ und $V_m = (v_1 v_2 ... v_m)$ als Zeilenvektoren ihrer Spalten interpretieren, können wir $R_m^T P_m^T = V_m^T$ als das lineare Gleichungssystem

$$R_m^T \begin{pmatrix} p_1^T \\ \vdots \\ p_m^T \end{pmatrix} = \begin{pmatrix} v_1^T \\ \vdots \\ v_m^T \end{pmatrix}$$

interpretieren, sodass wir $p_1, p_2, ..., p_m$ durch Vorwärtseinsetzen in die untere Dreiecksmatrix R_m^T berechnen können. Somit können wir für die Berechnung der m-ten QMR-Approximation $x^{(m)}$ das folgende, induktive Vorgehen wählen:
Wenn $\hat{R}_{m-1}, \hat{g}^{(m-1)}, P_{m-1}$ und $x^{(m-1)}$ bereits vorliegen,[1] fügen wir zunächst unten an $\hat{R}_{m-1}$ eine Nullzeile und unten an $\hat{g}^{(m-1)}$ eine Null an. Dann führen wir den m-ten Schritt des Bi-Lanczos-Algorithmus aus, um $\alpha_m, \delta_{m+1}, \beta_{m+1}, v_{m+1}$ und w_{m+1} zu erhalten. Dann fügen wir rechts an $\hat{R}_{m-1}$ die m-te Spalte von $\hat{T}_m$, d.h. $(0, ..., \overline{\beta_m}, \overline{\alpha_m}, \overline{\delta_{m+1}})^T$, an und multiplizieren diese mit den bisher verwendeten Givens-Rotationen. Da $\hat{T}_m$ eine Tridiagonalmatrix ist, müssen allerdings lediglich die beiden zuletzt verwendeten Rotationen Ω_{m-2} und Ω_{m-1} auf die neue Spalte angewendet werden. Dann bestimmen wir c_m und s_m für die nächste Givens-Rotation Ω_m und wenden diese auf die neue Spalte und die rechte Seite $\begin{pmatrix} \hat{g}^{(m-1)} \\ 0 \end{pmatrix}$ an, um so $\hat{R}_m$ und $\hat{g}^{(m)}$ zu erhalten. Dann bestimmen wir p_m durch Vorwärtseinsetzen in R_m^T. Wegen

$$\hat{g}^{(m)} = \Omega_m \begin{pmatrix} \hat{g}^{(m-1)} \\ 0 \end{pmatrix} = \Omega_m \begin{pmatrix} g^{(m-1)} \\ \gamma_m \\ 0 \end{pmatrix} = \begin{pmatrix} g^{(m-1)} \\ c_m \gamma_m \\ -s_m \gamma_m \end{pmatrix}^{2} \tag{2.8}$$

ist insbesondere $g^{(m)} = \begin{pmatrix} g^{(m-1)} \\ c_m \gamma_m \end{pmatrix}$, weshalb wir unsere Approximation in Kombination mit (2.7) anschließend vermöge

$$x^{(m)} = x^{(0)} + P_m g^{(m)} = x^{(0)} + P_{m-1} g^{(m-1)} + c_m \gamma_m p_m = x^{(m-1)} + c_m \gamma_m p_m$$

aus der vorherigen Approximation $x^{(m-1)}$ konstruieren können. Weiter folgt aus (2.8)

$$\gamma_{m+1} = -s_m \gamma_m. \tag{2.9}$$

Die Norm des Quasi-Residuums $|\gamma_{m+1}|$ kann also mit sehr geringem Aufwand berechnet werden, während die Berechnung der Norm des exakten Residuums $||b - Ax^{(m)}||$ eine aufwändige Matrix-Vektor-Multiplikation erfordern würde. Daher wählen wir für das Abbruchkriterium eine Fehlerschranke $\epsilon \in \mathbb{R}_{>0}$ und brechen ab, sobald die relative Quasi-Residuennorm diese Schranke unterschreitet, d.h. sobald

$$\frac{|\gamma_{m+1}|}{||r^{(0)}||} = \frac{|\gamma_{m+1}|}{|\gamma_1|} \leq \epsilon$$

gilt.
Darauf basierend formulieren wir nun den **QMR-Algorithmus**.

[1] $\hat{R}_0$ und P_0 interpretieren wir hierbei als nichts.
[2] $g^{(0)}$ interpretieren wir ebenfalls als nichts.

Algorithmus 2.7 ***(QMR)***

1: $r^{(0)} \leftarrow b - Ax^{(0)}$;
2: $\gamma_1 \leftarrow ||r^{(0)}||$;
3: $v_1 \leftarrow r^{(0)}/\gamma_1, w_1 \leftarrow r^{(0)}/\gamma_1$;
4: $m \leftarrow 0$;
5: **while** $|\gamma_{m+1}| > |\gamma_1|\epsilon$ **do**
6: $\quad m \leftarrow m + 1$;
7: $\quad$ Berechne $\alpha_m, \delta_{m+1}, \beta_{m+1}, v_{m+1}, w_{m+1}$ mit dem Bi-Lanczos-Algorithmus;
8: $\quad$ Wende Ω_{m-2} und Ω_{m-1} auf die m-te Spalte von $\hat{T}_m$ an;
9: $\quad$ Berechne die Rotationskoeffizienten c_m und s_m;
10: $\quad \gamma_{m+1} \leftarrow -s_m\gamma_m$;
11: $\quad \gamma_m \leftarrow c_m\gamma_m$;
12: $\quad r_{mm} \leftarrow c_m t_m + \overline{s_m}\overline{\delta_{m+1}} = \sqrt{|t_m|^2 + |\delta_{m+1}|^2}$;
13: $\quad p_m \leftarrow (v_m - \sum_{i=m-2}^{m-1} r_{im}p_i)/r_{mm}$;
14: $\quad x^{(m)} \leftarrow x^{(m-1)} + \gamma_m p_m$;
15: **end while**

Nun können wir die Vorteile des QMR-Verfahrens zusammenfassen:

- Wie bereits erwähnt, erfordert der Bi-Lanczos-Algorithmus in jedem Schritt den gleichen Speicherbedarf und Rechenaufwand.
- Wir benötigen in jedem Schritt nur die beiden zuvor verwendeten Givens-Rotationen. Daher müssen auch nur zwei Rotationen pro Schritt und somit eine konstante Anzahl von Rotationen abgespeichert werden. Zusammen mit der neuen Rotation müssen also auch nur eine konstante Anzahl an Rotationen durchgeführt werden.
- Da $\hat{T}_m$ eine Tridiagonal-Matrix ist, ist der Speicherbedarf für die neue Spalte konstant und durch die Givens-Rotationen wird nur eine Diagonale hinzugefügt, sodass in jeder Spalte von R_m nur bis zu 3 Einträge nicht 0 sind. Daher erfordert auch das Vorwärtseinsetzen in R_m^T lediglich einen konstanten Rechenaufwand.

Sowohl der Speicherbedarf als auch der pro Schritt erforderliche Rechenaufwand sind also unabhängig von der Anzahl der durchgeführten Iterationen. Somit ist das QMR-Verfahren auch für viel größere Dimensionen durchführbar als das GMRES-Verfahren.

2.3 Breakdown

Für die Durchführung des m-ten QMR-Schrittes haben wir vorausgesetzt, dass der Bi-Lanczos-Algorithmus nicht vor der Konstruktion von v_{m+1} und w_{m+1} abbricht. Daher wollen wir nun untersuchen, was passiert, falls es nach der Konstruktion von v_m und w_m zu einem Abbruch kommt. Dies ist genau dann der Fall, wenn

$$\delta_{m+1} = |\langle \hat{v}_{m+1}, \hat{w}_{m+1} \rangle|^{1/2} = 0$$

gilt, also genau dann, wenn $\hat{v}_{m+1}$ und $\hat{w}_{m+1}$ senkrecht aufeinander stehen.

Lemma 2.8 *Falls $\hat{v}_{m+1} = 0$ ist, gilt:*

$$AV_m = V_m T_m,$$

$$W_m^* A V_m = T_m.$$

Beweis. Dieser Beweis ist eine Abwandlung des Beweises von Satz 2.4.
Nach Z.6 und 14 des Bi-Lanczos-Algorithmus gilt wie zuvor für alle $j \in \{1, ..., m-1\}$

$$AV_m e_j = Av_j = \hat{v}_{j+1} + \overline{\alpha_j} v_j + \overline{\beta_j} v_{j-1} = \overline{\delta_{j+1}} v_{j+1} + \overline{\alpha_j} v_j + \overline{\beta_j} v_{j-1} = V_m T_m e_j.$$

Da $\hat{v}_{m+1} = 0$ ist, gilt außerdem mit Z.6 und 14

$$AV_m e_m = Av_m = \hat{v}_{m+1} + \overline{\alpha_m} v_m + \overline{\beta_m} v_{m-1} = \overline{\alpha_m} v_m + \overline{\beta_m} v_{m-1} = V_m T_m e_m.$$

Damit folgt $AV_m = V_m T_m$.
Außerdem folgt für $i \in \{1, ..., m\}$ und $j \in \{1, ..., m-1\}$ mit Satz 2.2

$$(W_m^* A V_m)_{ij} = \langle w_i, Av_j \rangle = \langle w_i, \overline{\delta_{j+1}} v_{j+1} + \overline{\alpha_j} v_j + \overline{\beta_j} v_{j-1} \rangle = \begin{cases} \overline{\delta_{j+1}}, & falls \;\; i = j+1 \\ \overline{\alpha_j}, & falls \;\; i = j \\ \overline{\beta_j}, & falls \;\; i = j-1 \\ 0, & sonst \end{cases}.$$

Ebenso folgt für $i \in \{1, ..., m\}$

$$(W_m^* A V_m)_{im} = \langle w_i, Av_m \rangle = \langle w_i, \overline{\alpha_m} v_m + \overline{\beta_m} v_{m-1} \rangle = \begin{cases} \overline{\alpha_m}, & falls \;\; i = m \\ \overline{\beta_m}, & falls \;\; i = m-1 \\ 0, & sonst \end{cases}.$$

Also ist $W_m^* A V_m = T_m$.

□

Falls $\hat{v}_{m+1} = 0$ ist, erhalten wir für ein beliebiges $x = x^{(0)} + V_m y \in x^{(0)} + K_m(A, r^{(0)})$ mit Lemma 2.8 und wegen $v_1 = \frac{r^{(0)}}{||r^{(0)}||}$ also

$$b - Ax = b - A(x^{(0)} + V_m y) = r^{(0)} - AV_m y = ||r^{(0)}||v_1 - V_m T_m y = V_m(||r^{(0)}||e_1 - T_m y).$$

Da wegen $\hat{v}_{m+1} = 0$ auch $\delta_{m+1} = 0$ und somit $\hat{T}_m = \begin{pmatrix} T_m \\ 0 \end{pmatrix}$ gilt, folgt also

$$||b-Ax|| = \left|\left|V_m(||r^{(0)}||e_1-T_m y)\right|\right| \leq \left|\left|V_m\right|\right| \ \left|\left||r^{(0)}||e_1-T_m y\right|\right| = \left|\left|V_m\right|\right| \ \left|\left||r^{(0)}||e_1-\hat{T}_m y\right|\right|. \tag{2.10}$$

Wir wollen nun nachweisen, dass der m-te QMR-Schritt auch in diesem Fall durchführbar ist. Dafür benötigen wir jedoch noch weitere Aussagen.

Lemma 2.9 *Falls $\hat{v}_{m+1} = 0$ ist, gilt*

$$K_m(A, v_1) = K_{m+1}(A, v_1).$$

Beweis. Dieser Beweis basiert auf [1, S.106-107].
Da $\hat{v}_{m+1} = 0$ ist, gilt nach Z.6 des Bi-Lanczos-Algorithmus

$$Av_m = \overline{\alpha_m} v_m + \overline{\beta_m} v_{m-1} \in span\{v_1, ..., v_m\} = K_m(A, v_1).$$

Wegen $A^{m-1}v_1 \in K_m(A, v_1)$ existieren außerdem $c_1, ..., c_m \in \mathbb{C}$ mit $A^{m-1}v_1 = \sum_{i=1}^{m} c_i v_i$. Damit gilt $A^m v_1 = \sum_{i=1}^{m} c_i A v_i$ und somit

$$A^m v_1 - c_m A v_m = \sum_{i=1}^{m-1} c_i A v_i \in K_m(A, v_1) = span\{v_1, ..., v_m\}.$$

Damit folgt nun mit dem Basisaustauschsatz

$$\begin{aligned} K_{m+1}(A, v_1) = span\{v_1, Av_1, ..., A^{m-1}v_1, A^m v_1\} &= span\{v_1, v_2, ..., v_m, A^m v_1\} \\ &= span\{v_1, v_2, ..., v_m, Av_m\} \subseteq K_m(A, v_1) \end{aligned}$$

und damit $K_{m+1}(A, v_1) = K_m(A, v_1)$.

□

Damit erhalten wir nun das folgende, wichtige Resultat.

Lemma 2.10 *Falls $\hat{v}_{m+1} = 0$ ist, ist T_m invertierbar.*

Beweis. Dieser Beweis ist eine Abwandlung des Beweises von Lemma 3.32 aus [1, S.111].
Sei $y \in \mathbb{C}^m$ mit $T_m y = 0$. Dann gilt mit Lemma 2.8 für alle $t \in \mathbb{C}^m$

$$0 = \langle t, T_m y\rangle = \langle t, W_m^* A V_m y\rangle = \langle W_m t, A V_m y\rangle.$$

Da $\{W_m t | t \in \mathbb{C}^m\} = span\{w_1, w_2, ..., w_m\}$ ist, gilt also $AV_m y \perp span\{w_1, w_2, ..., w_m\}$. Weiter ist $V_m y \in span\{v_1, v_2, ..., v_m\} = K_m(A, v_1)$. Also folgt mit Lemma 2.9

$$AV_m y \in K_{m+1}(A, v_1) = K_m(A, v_1) = span\{v_1, v_2, ..., v_m\}.$$

Also existieren $c_1, c_2, ..., c_m \in \mathbb{C}$ mit $AV_m y = \sum_{i=1}^{m} c_i v_i$. Wegen $\sum_{i=1}^{m} c_i w_i \in span\{w_1, w_2, ..., w_m\}$ folgt also mit Satz 2.2

$$0 = \left\langle AV_m y, \sum_{i=1}^{m} c_i w_i \right\rangle = \left\langle \sum_{i=1}^{m} c_i v_i, \sum_{i=1}^{m} c_i w_i \right\rangle = \sum_{i=1}^{m} \overline{c_i} c_i \langle v_i, w_i \rangle = \sum_{i=1}^{m} |c_i|^2$$

und damit $c_1 = c_2 = ... = c_m = 0$. Also gilt $AV_m y = \sum_{i=1}^{m} c_i v_i = 0$.
Da A regulär ist, ist A insbesondere injektiv. Da $v_1, v_2, ..., v_m$ linear unabhängig sind, ist darüber hinaus auch $V_m = (v_1 v_2 ... v_m)$ injektiv. Also ist auch AV_m injektiv und somit ist $y = 0$. Damit ist T_m injektiv und als quadratische Matrix somit invertierbar.

□

Nun stehen uns alle Argumente zur Verfügung, um zu beweisen, dass das QMR-Verfahren im Falle $\hat{v}_{m+1} = 0$ nicht nur durchführbar ist, sondern sogar die exakte Lösung unseres Gleichungssystems (1.1) berechnet.

Satz 2.11 (Lucky Breakdown) *Falls $\hat{v}_{m+1} = 0$ ist, kann die m-te QMR-Iterierte $x^{(m)}$ dennoch berechnet werden und ist sogar die exakte Lösung.*

Beweis Basiert auf dem Beweis von Proposition 6.10 aus [6, S.179], an QMR angepasst. Nach Lemma 2.10 ist T_m invertierbar. Da $Q_{m-1} \in \mathbb{C}^{m \times m}$ unitär und somit ebenfalls invertierbar ist, ist also auch die obere Dreiecksmatrix $Q_{m-1} T_m$ invertierbar. Damit muss $t_m = (Q_{m-1} T_m)_{mm} \neq 0$ gelten. Die m-te Givens-Rotation Ω_m kann also definiert werden mit

$$s_m = \frac{\overline{\delta_{m+1}}}{\sqrt{|t_m|^2 + |\delta_{m+1}|^2}} = \frac{0}{\sqrt{|t_m|^2 + 0}} = 0$$

und die obere Dreiecksmatrix R_m ist wegen

$$(R_m)_{mm} = \sqrt{|t_m|^2 + |\delta_{m+1}|^2} = \sqrt{|t_m|^2} = |t_m| > 0$$

auch in diesem Fall regulär. Die m-te QMR-Iterierte kann also wie zuvor durch $x^{(m)} = x^{(0)} + V_m y^{(m)}$ mit $y^{(m)} = R_m^{-1} g^{(m)}$ berechnet werden. Da $s_m = 0$ ist, folgt daher mit (2.10), (2.6) und (2.9)

$$||b - Ax^{(m)}|| \leq \left|\left|V_m\right|\right| \ \left|\left||r^{(0)}||e_1 - \hat{T}_m y^{(m)}\right|\right| = \left|\left|V_m\right|\right| \ |\gamma_{m+1}| = \left|\left|V_m\right|\right| \ |-s_m \gamma_m| = 0.$$

Damit ist $x^{(m)}$ die exakte Lösung.

□

Falls $\hat{v}_{m+1} = 0$ ist, ist der resultierende Abbruch des QMR-Verfahrens also absolut unproblematisch.
Dies ist jedoch nicht der Fall, wenn $\langle \hat{v}_{m+1}, \hat{w}_{m+1} \rangle = 0$ ist, obwohl $\hat{v}_{m+1} \neq 0$ ist. Diese Situation kann nämlich durchaus eintreten, bevor die exakte Lösung berechnet wird, sodass hier von einem sogenannten *Serious Breakdown* die Rede ist. Eine Möglichkeit, dennoch weitere Näherungslösungen berechnen zu können, bieten die in [6, S.231 -233] beschriebenen *Look-Ahead*-Varianten des Bi-Lanczos-Algorithmus:
Oftmals können v_{m+2} und w_{m+2} mit den gewünschten Eigenschaften auch ohne v_{m+1} und w_{m+1} konstruiert werden, sodass der Bi-Lanczos-Algorithmus anschließend bis zum nächsten Abbruch fortgeführt werden kann. Funktioniert dies nicht für v_{m+2} und w_{m+2}, so versucht man es mit v_{m+3} und w_{m+3}, usw. . Nach [6] sind diese Look-Ahead Varianten allerdings mit einer erhöhten Komplexität verbunden, und zwar insbesondere, weil die Matrix T_m durch Anwendung dieser Varianten sogar ihre Tridiagonalstruktur verliert. Als einfachere Alternative wird daher vorgeschlagen, die zuletzt berechnete Näherungslösung als neuen Startvektor zu verwenden, um darauf basierend einen Neustart des Bi-Lanczos-Algorithmus bzw. des QMR-Verfahrens vorzunehmen.
Wir halten fest: Falls der Bi-Lanczos-Algorithmus bzw. das QMR-Verfahren im m-ten Schritt vor der Konstruktion von v_{m+1} und w_{m+1} abbricht, gibt es zwei Möglichkeiten:

- Wenn $\hat{v}_{m+1} \neq 0$ ist, können wir ohne Weiteres den m-ten QMR-Schritt nicht durchführen und keine Aussage über die Genauigkeit der zuletzt berechneten Iterierten treffen, sodass ein Serious Breakdown entsteht.
- Gilt hingegen $\hat{v}_{m+1} = 0$, so wird nach Satz 2.11 als nächstes mit $x^{(m)}$ die exakte Lösung berechnet, sodass ein Lucky Breakdown entsteht.

Dass es spätestens im n-ten Schritt zu einem Abbruch kommen muss, ist eine direkte Folgerung aus Satz 2.3.

Korollar 2.12 *Der Bi-Lanczos-Algorithmus bricht vor der Konstruktion von v_{n+1} und w_{n+1} ab.*

Beweis.
Angenommen, der Bi-Lanczos-Algorithmus breche nicht vor der Konstruktion von v_{n+1} und w_{n+1} ab. Dann ist nach Satz 2.3 $\{v_1, v_2, ..., v_{n+1}\}$ eine Basis von $K_{n+1}(A, v_1)$. Da $K_{n+1}(A, v_1)$ ein Unterraum von $\mathbb{C}^n$ ist, gilt aber $dim(K_{n+1}(A, v_1)) \leq n$. $\lightning$ Es liegt also ein Widerspruch vor.

□

Wenn also kein Serious Breakdown auftritt, muss spätestens $\hat{v}_{n+1} = 0$ sein. Daraus folgt unmittelbar:

Korollar 2.13 *Angenommen, es komme nicht zu einem Serious Breakdown.*
Dann berechnet das QMR-Verfahren spätestens im n-ten Schritt die exakte Lösung.

2.4 Konvergenzeigenschaften

Wir wollen nun versuchen, für die m-te QMR-Approximation $x^{(m)}$ Aussagen über die Norm des Residuums $r^{(m)} = b - Ax^{(m)}$ zu treffen.
Da wir die möglichen Szenarien im Falle eines Abbruchs des QMR-Verfahrens bereits in Abschnitt 2.3 diskutiert haben, nehmen wir im Folgenden wieder an, dass der Bi-Lanczos-Algorithmus nicht vor der Konstruktion von v_{m+1} und w_{m+1} abbricht.

Satz 2.14 *Für das m-te QMR-Residuum $r^{(m)} = b - Ax^{(m)}$ gilt:*

$$||r^{(m)}|| \leq ||V_{m+1}||\ |\gamma_{m+1}| = ||V_{m+1}||\ \ |s_1 s_2 ... s_m|\ \ ||r^{(0)}||.$$

Beweis. Dieser Beweis basiert auf dem Beweis von Proposition 7.3 aus [6, S.238].
Nach (2.4) und (2.6) gilt

$$||b - Ax^{(m)}|| \leq \left|\left|V_{m+1}\right|\right|\ \ \left|\left||r^{(0)}||e_1 - \hat{T}_m y^{(m)}\right|\right| = \left|\left|V_{m+1}\right|\right|\ |\gamma_{m+1}|.$$

Außerdem erhalten wir durch wiederholte Anwendung von (2.9) und wegen $\gamma_1 = ||r^{(0)}||$

$$\gamma_{m+1} = (-1)^m s_1 s_2 ... s_m \gamma_1 = (-1)^m s_1 s_2 ... s_m ||r^{(0)}||.$$

Somit folgt insgesamt

$$||b - Ax^{(m)}|| \leq \left|\left|V_{m+1}\right|\right|\ \ \left|(-1)^m s_1 s_2 ... s_m\ ||r^{(0)}||\right| = \left|\left|V_{m+1}\right|\right|\ \ \left|s_1 s_2 ... s_m\right|\ \ ||r^{(0)}||.$$

□

Nun stellt sich die Frage, wie sich dieses Resultat auf die Konvergenz des QMR-Verfahrens auswirkt. Wir bemerken zunächst, dass wegen $|s_i| \leq 1$ für alle $i \in \{1, .., m\}$ auch $|s_1 s_2 ... s_m| \leq 1$ ist und somit einen positiven Effekt erzielt. Eine vergleichbare Aussage für $||V_{m+1}||$ können wir hingegen nicht treffen. Um dennoch eine praktische Abschätzung für $||V_{m+1}||$ zu erhalten, nehmen wir eine simple Modifikation des Bi-Lanczos-Algorithmus vor:
Um die benötigten Eigenschaften des Bi-Lanczos-Algorithmus nachzuweisen, müssen wir für $j \in \{1, ..., m\}$ von den Skalaren δ_{j+1} und β_{j+1} lediglich fordern, dass die Bedingung

$$\delta_{j+1}\beta_{j+1} = \langle \hat{v}_{j+1}, \hat{w}_{j+1} \rangle$$

erfüllt ist. Abgesehen davon können δ_{j+1} und β_{j+1} also völlig beliebig gewählt werden (vgl. [6, S.230]). Wählen wir nun

$$\delta_{j+1} = ||\hat{v}_{j+1}||, \quad \beta_{j+1} = \frac{\langle \hat{v}_{j+1}, \hat{w}_{j+1} \rangle}{\delta_{j+1}},$$

so gilt nicht nur die erforderliche Bedingung

$$\delta_{j+1}\beta_{j+1} = \delta_{j+1}\frac{\langle \hat{v}_{j+1}, \hat{w}_{j+1} \rangle}{\delta_{j+1}} = \langle \hat{v}_{j+1}, \hat{w}_{j+1} \rangle,$$

sondern auch

$$||v_{j+1}|| = \Big|\Big|\frac{\hat{v}_{j+1}}{\overline{\delta_{j+1}}}\Big|\Big| = \frac{||\hat{v}_{j+1}||}{||\hat{v}_{j+1}||} = 1.$$

Sofern wir also wie bisher $v_1 = \frac{r^{(0)}}{||r^{(0)}||}$ bereits normiert wählen, können wir durch eine alternative Wahl von δ_{j+1} sicherstellen, dass die Bi-Lanczos-Vektoren $v_1, ..., v_{m+1}$ normiert sind. Wegen $\beta_{m+1} = \frac{\langle \hat{v}_{m+1}, \hat{w}_{m+1} \rangle}{\delta_{m+1}}$ und Z.13 des Bi-Lanczos-Algorithmus kommt es dabei wie zuvor genau dann zu einem Abbruch im m-ten Schritt, wenn $\hat{v}_{m+1}$ und $\hat{w}_{m+1}$ senkrecht aufeinander stehen. Daher gelten sämtliche in diesem Kapitel getroffenen Aussagen auch auf Grundlage dieses modifizierten Bi-Lanczos-Algorithmus, sodass insbesondere das QMR-Verfahren auch in diesem Fall genau wie zuvor beschrieben durchgeführt werden kann (vgl. [4, S.301, Aufgabe 7.2]). Das Resultat ist also eine alternative Variante des QMR-Verfahrens, bei welcher die Spalten der Matrix V_{m+1} bzw. V_m normiert sind.

Satz 2.15 *Die Bi-Lanczos-Vektoren $v_1, ..., v_{m+1}$ seien normiert. Dann gilt*

$$||V_{m+1}|| \leq \sqrt{m+1}.$$

Beweis. Dieser Beweis basiert auf einer entsprechenden Aussage für das DQGMRES-Verfahren aus [6, S.182-183].
Sei $x \in \mathbb{C}^{m+1}$ mit $||x|| = 1$. Da $v_1, ..., v_{m+1}$ normiert sind, gilt dann

$$||V_{m+1}x|| = \Big|\Big| \sum_{i=1}^{m+1} x_i v_i \Big|\Big| \leq \sum_{i=1}^{m+1} ||x_i v_i|| = \sum_{i=1}^{m+1} |x_i| \, ||v_i|| = \sum_{i=1}^{m+1} |x_i|.$$

Seien nun $\tilde{x}, y \in \mathbb{C}^{m+1}$ mit $\tilde{x}_i := |x_i|$ und $y_i := 1$ für alle $i \in \{1, ..., m+1\}$. Dann folgt mit der Cauchy-Schwarz-Ungleichung

$$\sum_{i=1}^{m+1} |x_i| = |\langle \tilde{x}, y \rangle| \leq ||\tilde{x}|| \, ||y|| = \sqrt{\sum_{i=1}^{m+1} |x_i|^2} \sqrt{\sum_{i=1}^{m+1} 1} = ||x|| \, \sqrt{m+1} = \sqrt{m+1}.$$

Damit ist $||V_{m+1}x|| \leq \sqrt{m+1}$ und somit insgesamt $||V_{m+1}|| \leq \sqrt{m+1}$.

□

Mit Satz 2.14 und 2.15 erhalten wir nun folglich:

Korollar 2.16 *Die Bi-Lanczos-Vektoren $v_1, ..., v_{m+1}$ seien normiert. Dann gilt für das m-te QMR-Residuum $r^{(m)} = b - Ax^{(m)}$:*

$$||r^{(m)}|| \leq \sqrt{m+1} \;\; |s_1 s_2 ... s_m| \;\; ||r^{(0)}||.$$

Als nächstes wollen wir die Konvergenz des QMR-Verfahrens mit der des GMRES-Verfahrens vergleichen, um herauszufinden, wie stark die Qualität unserer Näherungslösung durch die Verwendung von nicht-orthonormalen Basen der Krylow-Räume beeinträchtigt wird. Auch hier spielt die Matrix V_{m+1} wieder eine entscheidende Rolle. Das folgende Resultat gilt dabei für beide vorgestellten Varianten des QMR-Verfahrens.

Satz 2.17 *Seien $x_Q^{(m)}$ die m-te Näherungslösung des QMR-Verfahrens und $x_G^{(m)}$ die m-te Näherungslösung des GMRES-Verfahrens. Dann gilt:*

$$||b - Ax_Q^{(m)}|| \leq \kappa(V_{m+1})\ ||b - Ax_G^{(m)}||.$$

Beweis. Dieser Beweis basiert auf Theorem 7.4 aus [6, S.239] und dem Beweis von Theorem 6.11 aus [6, S.185].
Sei zunächst

$$\mathcal{R} := \{b - Ax | x \in x^{(0)} + K_m(A, r^{(0)})\} \overset{(2.3)}{=} \{V_{m+1}(||r^{(0)}||e_1 - \hat{T}_m y) | y \in \mathbb{C}^m\} \tag{2.11}$$

die Menge aller Residuen bezüglich $x^{(0)} + K_m(A, r^{(0)})$. Dann sind

$$r_Q^{(m)} := b - Ax_Q^{(m)} \text{ und } r_G^{(m)} := b - Ax_G^{(m)}$$

Elemente von $\mathcal{R}$.
Weiter sei wie zuvor $y^{(m)} \in \mathbb{C}^m$ mit (2.5) und außerdem

$$t^{(m)} := ||r^{(0)}||e_1 - \hat{T}_m y^{(m)} \tag{2.12}$$

das m-te Quasi-Residuum.
Es existieren eine unitäre Matrix $Q \in \mathbb{C}^{n\times n}$ und eine obere Dreiecksmatrix $R \in \mathbb{C}^{n\times(m+1)}$ mit

$$V_{m+1} = QR.$$

Da $v_1, ..., v_{m+1}$ linear unabhängig sind, ist V_{m+1} injektiv und daher ist auch R injektiv, weil Q als unitäre Matrix invertierbar ist. Somit existiert eine invertierbare Matrix $S \in \mathbb{C}^{(m+1)\times(m+1)}$ mit

$$RS = \begin{pmatrix} I_{m+1} \\ 0 \end{pmatrix} \in \mathbb{C}^{n\times(m+1)}.$$

Also ist

$$N := V_{m+1}S = QRS = Q\begin{pmatrix} I_{m+1} \\ 0 \end{pmatrix} \in \mathbb{C}^{n\times(m+1)}$$

isometrisch.
Weil S invertierbar ist, gilt $V_{m+1} = NS^{-1}$, also folgt

$$r_Q^{(m)} = b - Ax_Q^{(m)} = b - A(x^{(0)} + V_m y^{(m)}) \overset{(2.3)}{=} V_{m+1}(||r^{(0)}||e_1 - \hat{T}_m y^{(m)}) \overset{(2.12)}{=} V_{m+1}t^{(m)} = NS^{-1}t^{(m)} \tag{2.13}$$

und weil N isometrisch ist, gilt somit

$$||r_Q^{(m)}|| = ||NS^{-1}t^{(m)}|| = ||S^{-1}t^{(m)}|| \leq ||S^{-1}||\ \ ||t^{(m)}||. \tag{2.14}$$

Wegen

$$SN^*V_{m+1} = SN^*NS^{-1} = SI_{m+1}S^{-1} = I_{m+1}$$

ist außerdem

$$\{SN^*r \mid r \in \mathcal{R}\} \overset{(2.11)}{=} \{SN^*V_{m+1}(||r^{(0)}||e_1 - \hat{T}_m y) \mid y \in \mathbb{C}^m\} = \{||r^{(0)}||e_1 - \hat{T}_m y \mid y \in \mathbb{C}^m\}. \tag{2.15}$$

Da N isometrisch ist, folgt mit (2.13) außerdem

$$SN^*r_Q^{(m)} = SN^*NS^{-1}t^{(m)} = SS^{-1}t^{(m)} = t^{(m)}. \tag{2.16}$$

Da $||t^{(m)}|| = min\left\{\left|\left|\,||r^{(0)}||e_1 - \hat{T}_m y\right|\right| \;\middle|\; y \in \mathbb{C}^m\right\} \overset{(2.15)}{=} min\{||SN^*r|| \mid r \in \mathcal{R}\}$ ist, gilt also für alle $r \in \mathcal{R}$

$$||t^{(m)}|| \overset{(2.16)}{=} ||SN^*r_Q^{(m)}|| \leq ||SN^*r|| \leq ||S||\,||N^*||\,||r|| = ||S||\,||N||\,||r|| \overset{N\ isometrisch}{=} ||S||\,||r||.$$

Also gilt insbesondere auch $||t^{(m)}|| \leq ||S||\,||r_G^{(m)}||$ und daher mit (2.14)

$$||r_Q^{(m)}|| \leq ||S^{-1}||\,||t^{(m)}|| \leq ||S^{-1}||\,||S||\,||r_G^{(m)}|| = \kappa(S)\,||r_G^{(m)}||. \tag{2.17}$$

Wegen $V_{m+1} = NS^{-1}$ gilt außerdem für alle $y \in \mathbb{C}^{m+1}$

$$||V_{m+1}y|| = ||NS^{-1}y|| \overset{N\ isometrisch}{=} ||S^{-1}y||,$$

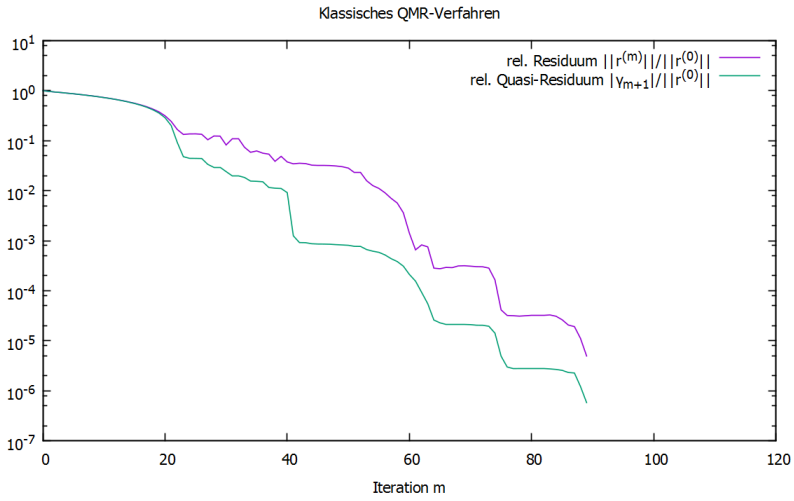
also

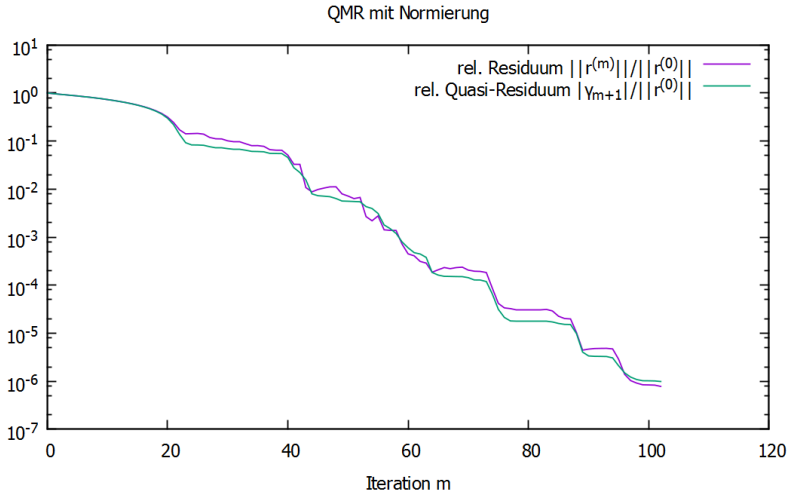
$$\kappa(V_{m+1}) = \frac{max\{||V_{m+1}y||\ |y \in \mathbb{C}^{m+1}, ||y|| = 1\}}{min\{||V_{m+1}y||\ |y \in \mathbb{C}^{m+1}, ||y|| = 1\}} = \frac{max\{||S^{-1}y||\ |y \in \mathbb{C}^{m+1}, ||y|| = 1\}}{min\{||S^{-1}y||\ |y \in \mathbb{C}^{m+1}, ||y|| = 1\}} = \kappa(S^{-1}) = \kappa(S).$$

Somit ergibt sich mit (2.17) insgesamt

$$||b - Ax_Q^{(m)}|| = ||r_Q^{(m)}|| \leq \kappa(S)\,||r_G^{(m)}|| = \kappa(V_{m+1})\,||b - Ax_G^{(m)}||.$$

□





Wenden wir das QMR-Verfahren auf unser Modellproblem aus Abschnitt 1.3 an, so macht sich im resultierenden Residuenverlauf vor allem bemerkbar, dass die Residuennorm anders als bei GMRES nicht kontinuierlich fällt oder stagniert, sondern auch hin und wieder ansteigt. Dies ist schlicht darauf zurückzuführen, dass hier statt einer Minimierung über geschachtelte Mengen lediglich eine Quasi-Minimierung durchgeführt wird.
Die Norm des Quasi-Residuums ist hingegen sehrwohl das Resultat einer exakten Minimierung über geschachtelte Mengen, sodass hier wie bei dem GMRES-Residuenverlauf kein Anstieg möglich ist und somit ein sehr glatter Verlauf entsteht.
Insgesamt ähnelt der QMR-Residuenverlauf durch die in Satz 2.14 und Satz 2.17 nachgewiesene, starke Abhängigkeit der QMR-Residuennormen $||r^{(m)}||$ von der Beschaffenheit der Matrizen V_{m+1} im Wesentlichen einem GMRES-Residuenverlauf mit stärkeren Oszillationen. Dabei weist die Variante mit Normierung erwartungsgemäß geringere Oszillationen auf und erreicht eine höhere Genauigkeit als die klassische QMR-Variante.

3 Transpose-Free QMR

Wie wir gesehen haben, besitzt das QMR-Verfahren durch den Verzicht auf Orthonormalbasen der Krylow-Räume die wesentlichen Nachteile des GMRES-Verfahrens nicht. Allerdings fordert das QMR-Verfahren auch etwas, was im GMRES-Verfahren nicht benötigt wird: Multiplikationen mit der Adjungierten Matrix A^*.
Abgesehen davon, dass die Multiplikationen mit A^* nicht direkt zur Berechnung der Näherungslösung $x^{(m)}$ beitragen, sondern nur zur Berechnung der Skalare α_m, β_{m+1} und δ_{m+1} benötigt werden, können in manchen Situationen Multiplikationen mit A^* gar nicht durchgeführt werden, z.B. wenn A nicht als explizite Matrix, sondern lediglich in Form einer Methode für die Berechnung von Multiplikationen mit A gegeben ist (vgl.[6, S.241]).
Ein Beispiel hierfür ist das mehrdimensionale *Newton-Verfahren*, angewendet auf eine total differenzierbare Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$. Sofern die Jacobi-Matrix $Df(x)$ von f in $x \in \mathbb{R}^n$ invertierbar ist, ist das mehrdimensionale Newton-Verfahren definiert durch

$$\Phi(x) := x - Df(x)^{-1} f(x).$$

Die Näherungslösung $\Phi(x)$ kann also durch das Lösen eines Gleichungssystems mit der Jacobi-Matrix $Df(x)$ berechnet werden.
Auch wenn diese Matrix nicht explizit bekannt ist, können Multiplikationen mit $Df(x)$ mit dem Differenzenquotienten vermöge

$$Df(x)v = \frac{f(x + \epsilon v) - f(x)}{\epsilon} \quad \forall v \in \mathbb{R}^n$$

mit $\epsilon \in \mathbb{R}_{>0}$ approximiert werden. Eine solche Formel existiert allerdings nicht für die Adjungierte $Df(x)^*$ von $Df(x)$ (vgl. [6, S.241]).
Es lohnt sich also, nach einer Variante des QMR-Verfahrens zu suchen, welche ohne Multiplikationen mit A^* auskommt. Dazu betrachten wir zunächst ein weiteres Krylow-Raum-Verfahren, welches auf dem Bi-Lanczos-Algorithmus basiert und eng mit dem QMR-Verfahren zusammenhängt, das sogenannte *BiCG-Verfahren (biconjugate gradients)*.

Das Vorgehen in diesem Kapitel orientiert sich hauptsächlich an [6] und [5].

3.1 Das BiCG-Verfahren

Das BiCG-Verfahren konstruiert eine Iterierte $x^{(m)} \in x^{(0)} + K_m(A, r^{(0)})$ mit der Eigenschaft

$$b - Ax^{(m)} \perp K_m(A^*, r^{(0)}).$$

Somit existiert auch für diese Näherungslösung ein $y^{(m)} \in \mathbb{C}^m$ mit $x^{(m)} = x^{(0)} + V_m y^{(m)}$. Wenn wir nun wieder $v^{(1)} = w^{(1)} = \frac{r^{(0)}}{||r^{(0)}||}$ wählen und davon ausgehen, dass der Bi-Lanczos-Algorithmus nicht vor der Konstruktion von v_{m+1} und w_{m+1} abbricht, oder alternativ, dass $\hat{v}_{m+1} = 0$ ist, und T_m zudem invertierbar ist, erhalten wir mit den Sätzen 2.2, 2.3 und 2.4 bzw. Lemma 2.8

$$\begin{aligned} b - Ax^{(m)} \perp K_m(A^*, r^{(0)}) &\iff W_m^*(b - Ax^{(m)}) = 0 \iff W_m^*(b - Ax^{(0)} - AV_m y^{(m)}) = 0 \\ &\iff W_m^*(r^{(0)} - AV_m y^{(m)}) = 0 \iff W_m^* AV_m y^{(m)} = W_m^* r^{(0)} \iff T_m y^{(m)} = W_m^* r^{(0)} \\ &\iff T_m y^{(m)} = W_m^* ||r^{(0)}|| v_1 \iff T_m y^{(m)} = ||r^{(0)}|| e_1 \iff y^{(m)} = ||r^{(0)}|| T_m^{-1} e_1. \end{aligned}$$

Damit folgt für $x^{(m)}$:

$$x^{(m)} = x^{(0)} + ||r^{(0)}|| V_m T_m^{-1} e_1.$$

Insbesondere ist $x^{(m)}$ eindeutig bestimmt. Da sich die Lösung $x^{(*)}$ von (1.1) nach (1.2) in $x^{(0)} + K_{m_0}(A, r^{(0)})$ befindet, gilt

$$b - Ax^{(*)} = 0 \perp K_{m_0}(A^*, r^{(0)}),$$

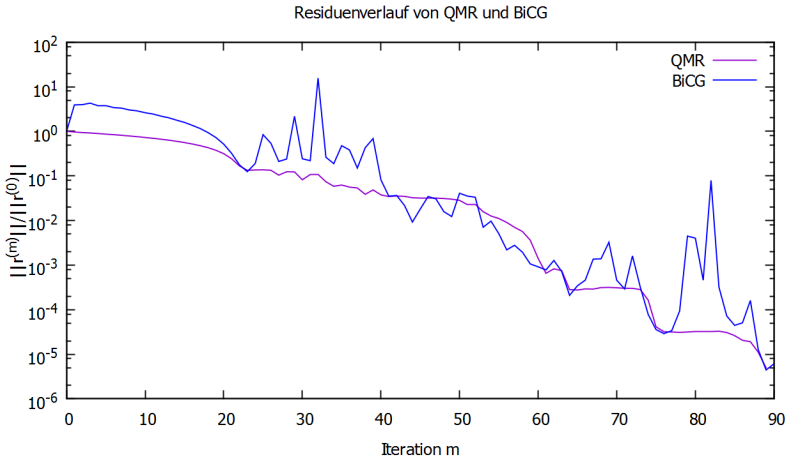
also berechnet das BiCG-Verfahren unter den gegebenen Voraussetzungen nach m_0 Schritten und somit spätestens nach n Schritten die exakte Lösung.
Für die Konstruktion des BiCG-Algorithmus wird zusätzlich vorausgesetzt, dass T_m eine LR-Zerlegung $T_m = L_m R_m$ besitzt. Unter Verwendung dieser LR-Zerlegung können anschließend Rekursionsgleichungen zur Aktualisierung der Näherungslösung $x^{(m)}$ und des zugehörigen Residuums $r^{(m)} = b - Ax^{(m)}$ hergeleitet werden.
Genauere Informationen zu der Herleitung und den Eigenschaften des Verfahrens sind in [5, S.194-201] und [4, S.269-278] zu finden.

Algorithmus 3.1 ***(BiCG)***

1: $r^{(0)} \leftarrow b - Ax^{(0)}, \tilde{r}^{(0)} \leftarrow r^{(0)}, p^{(0)} \leftarrow r^{(0)}, \tilde{p}^{(0)} \leftarrow r^{(0)}$;
2: $j \leftarrow 0$;
3: **while** $||r^{(j)}|| > ||r^{(0)}||\epsilon$ **do**
4: $\quad \alpha_j \leftarrow \langle r^{(j)}, \tilde{r}^{(j)} \rangle / \langle Ap^{(j)}, \tilde{p}^{(j)} \rangle$;
5: $\quad x^{(j+1)} \leftarrow x^{(j)} + \overline{\alpha_j} p^{(j)}$;
6: $\quad r^{(j+1)} \leftarrow r^{(j)} - \overline{\alpha_j} Ap^{(j)}$;
7: $\quad \tilde{r}^{(j+1)} \leftarrow \tilde{r}^{(j)} - \alpha_j A^* \tilde{p}^{(j)}$;
8: $\quad \beta_j \leftarrow \langle r^{(j+1)}, \tilde{r}^{(j+1)} \rangle / \langle r^{(j)}, \tilde{r}^{(j)} \rangle$;
9: $\quad p^{(j+1)} \leftarrow r^{(j+1)} + \overline{\beta_j} p^{(j)}$;
10: $\quad \tilde{p}^{(j+1)} \leftarrow \tilde{r}^{(j+1)} + \beta_j \tilde{p}^{(j)}$;
11: $\quad j \leftarrow j + 1$;
12: **end while**

In der Regel ist das QMR-Verfahren vorteilhafter als das BiCG-Verfahren, da für die Durchführung eines QMR-Schrittes weniger vorausgesetzt wird und es somit weniger

Gründe für einen vorzeitigen Abbruch gibt. Zudem besitzt das QMR-Verfahren durch die Minimierung des Quasi-Residuums einen wesentlich glatteren Residuenverlauf. Tatsächlich gibt es einen sehr starken Zusammenhang zwischen den Näherungslösungen des BiCG-Verfahrens und den Näherungslösungen des QMR-Verfahrens. Es kann sogar ein kurzer Algorithmus konstruiert werden, der die QMR-Iterierten auf Basis der BiCG-Iterierten berechnet. Dies ist der sogenannte *QMR-Smoothing-Algorithmus*, siehe [6, S.236-241]. Man kann also durchaus davon sprechen, dass das QMR-Verfahren einer Glättung der BiCG-Residuen entspricht.



Um eine Variante des QMR-Verfahrens zu konstruieren, die ohne Multiplikationen mit A^* auskommt, betrachten wir zunächst eine entsprechende Variante des BiCG-Verfahrens, das sogenannte *CGS-Verfahren (conjugate gradients squared)*.

3.2 Das CGS-Verfahren

Unter Verwendung von Z.6 und Z.9 und der Startbedingung $p^{(0)} = r^{(0)}$ des BiCG-Verfahrens kann gezeigt werden, dass für $j \in \{0, ..., m_0\}$ Polynome $\phi_j, \pi_j \in \Pi_j$ mit

$$r^{(j)} = \phi_j(A)r^{(0)} \quad \text{und} \quad p^{(j)} = \pi_j(A)r^{(0)}$$

existieren. Analog folgt mit Z.7 und Z.10

$$\tilde{r}^{(j)} = \overline{\phi_j}(A^*)\tilde{r}^{(0)} \quad \text{und} \quad \tilde{p}^{(j)} = \overline{\pi_j}(A^*)\tilde{r}^{(0)},$$

wobei $\overline{\phi_j}$ und $\overline{\pi_j}$ den Polynomen ϕ_j und π_j mit komplex konjugierten Koeffizienten entsprechen. Damit erhalten wir für die Skalare α_j, β_j

$$\alpha_j = \frac{\langle r^{(j)}, \tilde{r}^{(j)} \rangle}{\langle Ap^{(j)}, \tilde{p}^{(j)} \rangle} = \frac{\langle \phi_j(A)r^{(0)}, \overline{\phi_j}(A^*)\tilde{r}^{(0)} \rangle}{\langle A\pi_j(A)r^{(0)}, \overline{\pi_j}(A^*)\tilde{r}^{(0)} \rangle} = \frac{\langle \phi_j^2(A)r^{(0)}, \tilde{r}^{(0)} \rangle}{\langle A\pi_j^2(A)r^{(0)}, \tilde{r}^{(0)} \rangle},$$

$$\beta_j = \frac{\langle r^{(j+1)}, \tilde{r}^{(j+1)} \rangle}{\langle r^{(j)}, \tilde{r}^{(j)} \rangle} = \frac{\langle \phi_{j+1}(A) r^{(0)}, \overline{\phi_{j+1}}(A^*) \tilde{r}^{(0)} \rangle}{\langle \phi_j(A) r^{(0)}, \overline{\phi_j}(A^*) \tilde{r}^{(0)} \rangle} = \frac{\langle \phi_{j+1}^2(A) r^{(0)}, \tilde{r}^{(0)} \rangle}{\langle \phi_j^2(A) r^{(0)}, \tilde{r}^{(0)} \rangle},$$

also eine Darstellung von α_j und β_j ohne Verwendung von A^*.
Weiter gelten nach Z.6 und Z.9 des BiCG-Algorithmus die Rekursionen

$$\phi_{j+1}(t) = \phi_j(t) - \overline{\alpha_j} t \pi_j(t) \quad \text{und} \quad \pi_{j+1}(t) = \phi_{j+1}(t) + \overline{\beta_j} \pi_j(t) \tag{3.1}$$

mit $\phi_0(t) \equiv 1$ und $\pi_0(t) \equiv 1$. Definiert man anschließend das Residuum $r^{(j)}$ und den Vektor $p^{(j)}$ neu vermöge

$$r^{(j)} := \phi_j^2(A) r^{(0)} \quad \text{und} \quad p^{(j)} := \pi_j^2(A) r^{(0)},$$

so können mit Hilfe dieser Rekursionen und der Hilfsvektoren

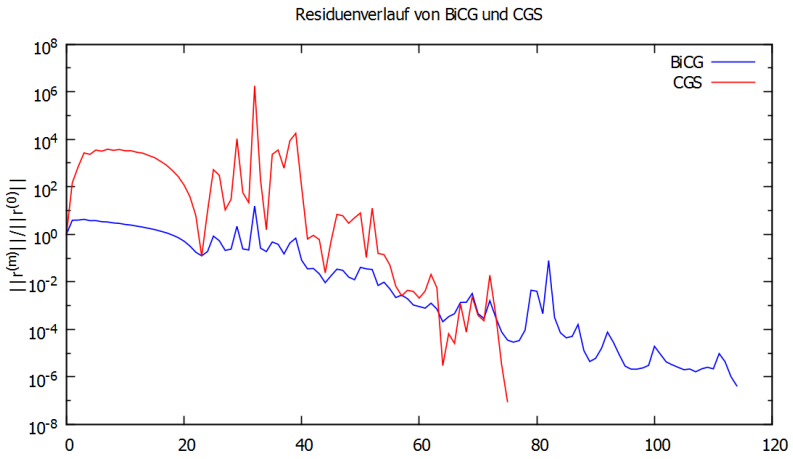
$$u^{(j)} := \phi_j(A) \pi_j(A) r^{(0)} \quad \text{und} \quad q^{(j)} := \phi_{j+1}(A) \pi_j(A) r^{(0)} \tag{3.2}$$

wiederum Rekursionen für dieses neue Residuum $r^{(j)}$ und die korrespondierende neue Näherungslösung $x^{(j)}$ hergeleitet werden. Das Resultat ist der CGS-Algorithmus. Eine genaue Herleitung ist in [6, S.241-244] und [5, S.201-204] zu finden.

Algorithmus 3.2 ***(CGS)***

$r^{(0)} \leftarrow b - Ax^{(0)}, \tilde{r}^{(0)} \leftarrow r^{(0)}, p^{(0)} \leftarrow r^{(0)}, u^{(0)} \leftarrow r^{(0)}$;
$j \leftarrow 0$;
while $||r^{(j)}|| > ||r^{(0)}|| \epsilon$ **do**
    $\alpha_j \leftarrow \langle r^{(j)}, \tilde{r}^{(0)} \rangle / \langle Ap^{(j)}, \tilde{r}^{(0)} \rangle$;
    $q^{(j)} \leftarrow u^{(j)} - \overline{\alpha_j} Ap^{(j)}$;
    $x^{(j+1)} \leftarrow x^{(j)} + \overline{\alpha_j}(u^{(j)} + q^{(j)})$;
    $r^{(j+1)} \leftarrow r^{(j)} - \overline{\alpha_j} A(u^{(j)} + q^{(j)})$;
    $\beta_j \leftarrow \langle r^{(j+1)}, \tilde{r}^{(0)} \rangle / \langle r^{(j)}, \tilde{r}^{(0)} \rangle$;
    $u^{(j+1)} \leftarrow r^{(j+1)} + \overline{\beta_j} q^{(j)}$;
    $p^{(j+1)} \leftarrow u^{(j+1)} + \overline{\beta_j}(q^{(j)} + \overline{\beta_j} p^{(j)})$;
    $j \leftarrow j + 1$;
end while

Statt einer Multiplikation mit A und einer mit A^* werden nun also zwei Multiplikationen mit A durchgeführt.
Aufgrund der Quadrierung der Polynome ist von dem CGS-Verfahren meistens eine wesentlich schnellere Konvergenz zu erwarten als von dem BiCG-Verfahren. Allerdings werden dadurch auch die im BiCG-Residuenverlauf auftretenden Oszillationen deutlich verstärkt (vgl. [4, S.281-282]) .



Wir wollen nun die im CGS-Verfahren konstruierten Vektoren und Skalare verwenden, um unsere A^* -freie QMR-Variante zu konstruieren. Dazu sei wie üblich $m \in \mathbb{N}$.
Wir definieren zunächst für alle $k \in \{1, ..., m\}$

$$v^{(k)} := \begin{cases} u^{(\lfloor \frac{k-1}{2} \rfloor)} & falls \;\; k \;\; ungerade \\ q^{(\lfloor \frac{k-1}{2} \rfloor)} & falls \;\; k \;\; gerade \end{cases} \tag{3.3}$$

und erhalten das folgende Resultat.

Satz 3.3 *Angenommen, der CGS-Algorithmus breche nicht vorzeitig ab. Dann gilt*

$$span\{v^{(1)}, ..., v^{(m)}\} = K_m(A, r^{(0)}).$$

Insbesondere ist $\{v^{(1)}, ..., v^{(m)}\}$ *eine Basis von* $K_m(A, r^{(0)})$, *sofern* $dim(K_m(A, r^{(0)})) = m$ *gilt.*

Beweis. Dieser Beweis basiert auf dem Beweis von Lemma 4.98 aus [5, S.210-211] und auf [4, S.288-289].
Sei $k \in \{1, ..., m\}$ beliebig.
Ist k ungerade, so existiert ein $j \in \mathbb{N}_0$ mit $k = 2j + 1$. Damit ist $\lfloor \frac{k-1}{2} \rfloor = j$ und somit nach (3.2) und (3.3)

$$v^{(k)} = u^{(j)} = \phi_j(A)\pi_j(A)r^{(0)}.$$

Da nach Voraussetzung der CGS-Algorithmus nicht vor der Konstruktion von $u^{(j)}$ abbricht, gilt außerdem $\alpha_i \neq 0$ für alle $i \in \{0, ..., j-1\}$. Nach den in (3.1) aufgeführten Rekursionsformeln haben also sowohl ϕ_j als auch π_j den Grad j. Damit gilt

$$deg(\phi_j \pi_j) = 2j = k - 1.$$

Ist k gerade, so existiert hingegen ein $j \in \mathbb{N}$ mit $k = 2j$. Damit ist $\lfloor \frac{k-1}{2} \rfloor = j - 1$ und es folgt mit (3.2) und (3.3) analog

$$v^{(k)} = q^{(j-1)} = \phi_j(A)\pi_{j-1}(A)r^{(0)}, \quad deg(\phi_j \pi_{j-1}) = 2j - 1 = k - 1.$$

Also existiert für alle $k \in \{1, ..., m\}$ ein Polynom $s_k \in \Pi_k$ mit $deg(s_k) = k - 1$ und $v^{(k)} = s_k(A)r^{(0)}$.
Damit folgt mit dem Basisaustauschsatz

$$K_m(A, r^{(0)}) = span\{r^{(0)}, Ar^{(0)}, ..., A^{m-1}r^{(0)}\} = span\{v^{(1)}, ..., v^{(m)}\}.$$

□

Nun bemerken wir, dass die j-te CGS-Iterierte $x^{(j)}$ nach Z.6 alternativ auch in zwei Halbschritten vermöge

$$x^{(j+\frac{1}{2})} := x^{(j)} + \overline{\alpha_j} u^{(j)} \quad \text{und} \quad x^{(j+1)} := x^{(j+\frac{1}{2})} + \overline{\alpha_j} q^{(j)}$$

aktualisiert werden kann. Zur Vereinfachung der im Folgenden verwendeten Notation verdoppeln wir nun die Indizes im CGS-Algorithmus. Wir schreiben also

$$\alpha_{2j} := \langle r^{(2j)}, \tilde{r}^{(0)} \rangle / \langle Ap^{(2j)}, \tilde{r}^{(0)} \rangle, \tag{3.4}$$
$$q^{(2j)} := u^{(2j)} - \overline{\alpha_{2j}} Ap^{(2j)}, \tag{3.5}$$
$$x^{(2j+1)} := x^{(2j)} + \overline{\alpha_{2j}} u^{(2j)}, \tag{3.6}$$
$$x^{(2j+2)} := x^{(2j+1)} + \overline{\alpha_{2j}} q^{(2j)}, \tag{3.7}$$
$$r^{(2j+2)} := r^{(2j)} - \overline{\alpha_{2j}} A(u^{(2j)} + q^{(2j)}), \tag{3.8}$$
$$\beta_{2j} := \langle r^{(2j+2)}, \tilde{r}^{(0)} \rangle / \langle r^{(2j)}, \tilde{r}^{(0)} \rangle, \tag{3.9}$$
$$u^{(2j+2)} := r^{(2j+2)} + \overline{\beta_{2j}} q^{(2j)}, \tag{3.10}$$
$$p^{(2j+2)} := u^{(2j+2)} + \overline{\beta_{2j}}(q^{(2j)} + \overline{\beta_{2j}} p^{(2j)}). \tag{3.11}$$

Anschließend definieren wir für jedes ungerade $i \in \mathbb{N}$

$$u^{(i)} := q^{(i-1)} \quad \text{und} \quad \alpha_i := \alpha_{i-1}, \tag{3.12}$$

sodass sich die m-te CGS-Iterierte mit (3.6) und (3.7) nun schreiben lässt als

$$x^{(m)} = x^{(m-1)} + \overline{\alpha_{m-1}} u^{(m-1)} \tag{3.13}$$

und das korrespondierende Residuum als

$$r^{(m)} = b - Ax^{(m)} = r^{(m-1)} - \overline{\alpha_{m-1}} Au^{(m-1)}, \tag{3.14}$$

unabhängig davon, ob m gerade oder ungerade ist.
Wir wollen diese CGS-Residuen im Folgenden jedoch verwenden, um die Iterierten eines anderen Verfahrens herzuleiten, sodass diese Residuen $r^{(m)}$ im Folgenden nicht mit der von uns betrachteten Näherungslösung korrespondieren werden. Um Verwechslungen zu vermeiden, bezeichnen wir daher von nun an diese CGS-Residuen mit $w^{(m)}$ statt $r^{(m)}$. Wir erhalten mit (3.14) also insbesondere

$$w^{(m)} = w^{(m-1)} - \overline{\alpha_{m-1}} Au^{(m-1)}. \tag{3.15}$$

Wenn wir annehmen, dass das CGS-Verfahren nicht vorzeitig abbricht, erhalten wir mit diesen neuen Schreibweisen nun nach (3.3) und Satz 3.3

$$K_m(A, r^{(0)}) = span\{v^{(1)}, ..., v^{(m)}\} = span\{u^{(0)}, ..., u^{(m-1)}\} \tag{3.16}$$

und außerdem, dass $\alpha_0, \alpha_1, ..., \alpha_{m-1} \neq 0$ sind, sodass wir nun die Matrizen

$$U_m := (u^{(0)}u^{(1)}...u^{(m-1)}) \in \mathbb{C}^{n\times m}, \;\; W_{m+1} := (w^{(0)}w^{(1)}...w^{(m)}) \in \mathbb{C}^{n\times(m+1)}$$

und die Bidiagonalmatrix

$$\hat{B}_m := \begin{pmatrix} 1/\overline{\alpha_0} & & & & \\ -1/\overline{\alpha_0} & 1/\overline{\alpha_1} & & & \\ & -1/\overline{\alpha_1} & 1/\overline{\alpha_2} & & \\ & & \ddots & \ddots & \\ & & & -1/\overline{\alpha_{m-2}} & 1/\overline{\alpha_{m-1}} \\ & & & & -1/\overline{\alpha_{m-1}} \end{pmatrix} \in \mathbb{C}^{(m+1)\times m}$$

definieren können. Darauf basierend können wir nun die folgende Aussage formulieren.

Satz 3.4 *Sofern das CGS-Verfahren nicht vorzeitig abbricht, gilt*

$$AU_m = W_{m+1}\hat{B}_m.$$

Beweis. Dieser Beweis basiert auf [6, S.248-249].
Sei $i \in \{1, ..., m\}$. Dann gilt nach (3.15)

$$w^{(i)} = w^{(i-1)} - \overline{\alpha_{i-1}}Au^{(i-1)}.$$

Damit folgt

$$AU_m e_i = Au^{(i-1)} = \frac{w^{(i-1)} - w^{(i)}}{\overline{\alpha_{i-1}}} = W_{m+1}\hat{B}_m e_i.$$

□

Wir nehmen außerdem an, dass $w^{(0)}, ..., w^{(m)} \neq 0$ sind und definieren darauf basierend die invertierbare Diagonalmatrix

$$\Lambda_{m+1} := \begin{pmatrix} ||w^{(0)}|| & & & \\ & ||w^{(1)}|| & & \\ & & \ddots & \\ & & & ||w^{(m)}|| \end{pmatrix} \in \mathbb{C}^{(m+1)\times(m+1)},$$

die wir verwenden wollen, um die Spalten von W_{m+1} zu normieren.
Sei nun wieder $x \in x^{(0)} + K_m(A, r^{(0)})$ beliebig. Dann existiert nach (3.16) ein $y \in \mathbb{C}^m$ mit $x = x^{(0)} + U_m y$. Damit erhalten wir nach Satz 3.4 und wegen $w^{(0)} = r^{(0)}$

$$\begin{aligned} b - Ax &= b - Ax^{(0)} - AU_m y = r^{(0)} - W_{m+1}\hat{B}_m y = W_{m+1}[e_1 - \hat{B}_m y] \\ &= W_{m+1}\Lambda_{m+1}^{-1}\Lambda_{m+1}[e_1 - \hat{B}_m y] = W_{m+1}\Lambda_{m+1}^{-1}\Big[||r^{(0)}||e_1 - \Lambda_{m+1}\hat{B}_m y\Big]. \end{aligned} \tag{3.17}$$

Definieren wir nun die Bidiagonalmatrix

$$\hat{H}_m := (h_{ij})_{i\in\{1,\dots,m+1\},j\in\{1,\dots,m\}} := \Lambda_{m+1}\hat{B}_m \in \mathbb{C}^{(m+1)\times m},$$

so erhalten wir mit (3.17)

$$||b - Ax|| = \left|\left|W_{m+1}\Lambda_{m+1}^{-1}\left[||r^{(0)}||e_1 - \hat{H}_m y\right]\right|\right| \leq \left|\left|W_{m+1}\Lambda_{m+1}^{-1}\right|\right| \ \left|\left|||r^{(0)}||e_1 - \hat{H}_m y\right|\right|. \tag{3.18}$$

Nun können wir das Ziel des sogenannten *TFQMR-Verfahrens (transpose-free quasi-minimal residual method)* formulieren:
Finde $y^{(m)} \in \mathbb{C}^m$ mit

$$\left|\left|||r^{(0)}||e_1 - \hat{H}_m y^{(m)}\right|\right| = min\left\{\left|\left|||r^{(0)}||e_1 - \hat{H}_m y\right|\right| \mid y \in \mathbb{C}^m\right\} \tag{3.19}$$

und konstruiere die m-te TFQMR-Iterierte vermöge

$$x^{(m)} := x^{(0)} + U_m y^{(m)}. \tag{3.20}$$

Damit ist unsere A^*-freie QMR-Variante nun definiert: Das TFQMR-Verfahren führt wie das QMR-Verfahren eine Quasi-Minimierung durch, verwendet allerdings für die Konstruktion der Krylow-Raum-Basen nun nicht mehr den Bi-Lanczos-Algorithmus, sondern das CGS-Verfahren, welches keinen Zugriff auf A^* erfordert.

Definition 3.5 (Quasi-Residuum) *Im Folgenden bezeichnen wir*

$$||r^{(0)}||e_1 - \hat{H}_m y^{(m)}$$

als das m-te Quasi-Residuum des TFQMR-Verfahrens.

3.3 Das TFQMR-Verfahren

Um die m-te TFQMR-Iterierte $x^{(m)}$ zu konstruieren, wollen wir die Norm des Quasi-Residuums $\left|\left|||r^{(0)}||e_1 - \hat{H}_m y\right|\right|$ minimieren und gehen dafür zunächst wie bei GMRES und QMR vor, indem wir geeignete Givens-Rotationen einführen, um die Bidiagonalmatrix

$$\hat{H}_m = \begin{pmatrix} ||w^{(0)}||/\overline{\alpha_0} & & & & \\ -||w^{(1)}||/\overline{\alpha_0} & ||w^{(1)}||/\overline{\alpha_1} & & & \\ & -||w^{(2)}||/\overline{\alpha_1} & ||w^{(2)}||/\overline{\alpha_2} & & \\ & & \dots & & \\ & & & -||w^{(m-1)}||/\overline{\alpha_{m-2}} & ||w^{(m-1)}||/\overline{\alpha_{m-1}} \\ & & & & -||w^{(m)}||/\overline{\alpha_{m-1}} \end{pmatrix}$$

auf obere Dreiecksgestalt zu überführen.

Wir setzen also wieder

$$\Omega_0 := I_{m+1}, \;\; c_0 := 1, \;\; s_0 := 0,$$

und definieren für $i \in \{1, ..., m\}$ zunächst

$$t_i := (\Omega_{i-1}\Omega_{i-2}...\Omega_0\hat{H}_m)_{ii}.$$

Dann defnieren wir die i-te Givens-Rotation $\Omega_i \in \mathbb{C}^{(m+1)\times(m+1)}$ durch

$$\Omega_i := \begin{array}{c} \\ i \\ i+1 \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccccc} & & & i & i+1 & & & \end{array} \\ \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & c_i & \overline{s_i} & & & \\ & & & -s_i & \overline{c_i} & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix} \end{array}, c_i := \frac{\overline{t_i}}{\sqrt{|t_i|^2+|h_{i+1,i}|^2}}, s_i := \frac{h_{i+1,i}}{\sqrt{|t_i|^2+|h_{i+1,i}|^2}}.$$

Auf diese Weise erhalten wir analog zum GMRES- und QMR-Verfahren vermöge

$$Q_m := \Omega_m\Omega_{m-1}...\Omega_1 \in \mathbb{C}^{(m+1)\times(m+1)}$$

eine unitäre Matrix, die $\hat{H}_m$ durch Multiplikation in eine obere Dreiecksmatrix

$$\hat{R}_m := (r_{ij})_{i\in\{1,...,m+1\},j\in\{1,...,m\}} := Q_m\hat{H}_m \in \mathbb{C}^{(m+1)\times m}$$

transformiert. Genauso erhalten wir durch Multiplikation der rechten Seite $||r^{(0)}||e_1$ des linearen Gleichungssystems $\hat{H}_m y = ||r^{(0)}||e_1$ mit Q_m wieder den vollbesetzten Vektor

$$\hat{g}^{(m)} := Q_m||r^{(0)}||e_1 \in \mathbb{C}^{(m+1)}.$$

Notation 3.6 *Im Folgenden definieren wir:*

- $R_m \in \mathbb{C}^{m\times m}$ *als die Matrix* $\hat{R}_m$ *ohne dessen letzte Zeile,*
- $g^{(m)} \in \mathbb{C}^m$ *als den Vektor* $\hat{g}^{(m)}$ *ohne dessen letzte Komponente* $\hat{g}^{(m)}_{m+1}$,
- $\gamma_{m+1} := \hat{g}^{(m)}_{m+1}$,
- $\hat{g}^{(0)} := (||r^{(0)}||) \in \mathbb{C}^1, \gamma_1 := ||r^{(0)}||$,
- $Q_0 := I_1 \in \mathbb{C}^{1\times 1}$.

Wie zuvor gilt nun

$$\hat{g}^{(m)} = \Omega_m \begin{pmatrix} \hat{g}^{(m-1)} \\ 0 \end{pmatrix} = \Omega_m \begin{pmatrix} g^{(m-1)} \\ \gamma_m \\ 0 \end{pmatrix} = \begin{pmatrix} g^{(m-1)} \\ c_m\gamma_m \\ -s_m\gamma_m \end{pmatrix}$$ [1]

[1] $g^{(0)}$ interpretieren wir hierbei als nichts.

und somit insbesondere

$$g^{(m)} = \begin{pmatrix} g^{(m-1)} \\ c_m \gamma_m \end{pmatrix}, \quad \gamma_{m+1} = -s_m \gamma_m. \tag{3.21}$$

Um unsere m-te TFQMR-Iterierte $x^{(m)} = x^{(0)} + U_m y^{(m)}$ mit (3.19) zu konstruieren, könnten wir nun also analog zum QMR-Verfahren vorgehen. Stattdessen wollen wir nun jedoch eine praktische, rekursive Darstellung von $x^{(m)}$ herleiten.
Hierzu bemerken wir zunächst, dass für alle $i \in \{1, ..., m\}$ wegen $||w_i|| \neq 0$ ebenfalls $h_{i+1,i} \neq 0$ ist und somit

$$(R_m)_{ii} = \sqrt{|t_i|^2 + |h_{i+1,i}|^2} > 0$$

gilt. Die obere Dreiecksmatrix R_m ist also regulär und für unser $y^{(m)}$ mit (3.19) folgt ebenso wie bei dem GMRES- und dem QMR-Verfahren

$$y^{(m)} = R_m^{-1} g^{(m)}. \tag{3.22}$$

Bezeichnen wir im Folgenden die m-te CGS-Iterierte mit $\tilde{x}^{(m)} \in x^{(0)} + K_m(A, r^{(0)})$, so liefert (3.13) außerdem die Darstellung

$$\tilde{x}^{(m)} = x^{(0)} + U_m \tilde{y}^{(m)} \tag{3.23}$$

mit

$$\tilde{y}^{(m)} := (\overline{\alpha_0}, \overline{\alpha_1}, ..., \overline{\alpha_{m-1}})^T \in \mathbb{C}^m. \tag{3.24}$$

Weiter definieren wir $H_m \in \mathbb{C}^{m \times m}$ als die Matrix $\hat{H}_m$ ohne dessen letzte Zeile, womit folglich

$$H_m \tilde{y}^{(m)} = ||w^{(0)}|| e_1 = ||r^{(0)}|| e_1 \tag{3.25}$$

gilt.
Damit können wir nun eine alternative Darstellung für $x^{(m)}$ formulieren.

Satz 3.7 *Für die m-te TFQMR-Iterierte $x^{(m)}$ und die m-te CGS-Iterierte $\tilde{x}^{(m)}$ gilt*

$$x^{(m)} = x^{(m-1)} + |c_m|^2 (\tilde{x}^{(m)} - x^{(m-1)}).$$

Außerdem ist $c_m \neq 0$. Für $m \geq 2$ gilt zudem

$$R_m \tilde{y}^{(m)} = \begin{pmatrix} g^{(m-1)} \\ \overline{c_m}^{-1} \gamma_m \end{pmatrix}.$$

Beweis. Dieser Beweis basiert auf den Beweisen von Satz 4.101 und 4.102 aus [5, S.213-218].
Unter Verwendung von (3.22) und (3.24) erhalten wir durch direktes Nachrechnen

$$\tilde{y}^{(1)} = \overline{\alpha_0}, \quad c_1 = \frac{|\alpha_0|}{\alpha_0} \frac{||w^{(0)}||}{\sqrt{||w^{(0)}||^2 + ||w^{(1)}||^2}}, \quad y^{(1)} = \overline{\alpha_0} \frac{||w^{(0)}||^2}{||w^{(0)}||^2 + ||w^{(1)}||^2}.$$

Wegen $u^{(0)} = r^{(0)} = w^{(0)}$ gilt nach (3.20) und (3.23) also

$$\tilde{x}^{(1)} = x^{(0)} + U_1\tilde{y}^{(1)} = x^{(0)} + \tilde{y}^{(1)}u^{(0)} = x^{(0)} + \overline{\alpha_0}r^{(0)},$$
$$x^{(1)} = x^{(0)} + U_1 y^{(1)} = x^{(0)} + y^{(1)}u^{(0)} = x^{(0)} + \overline{\alpha_0}\frac{||w^{(0)}||^2}{||w^{(0)}||^2 + ||w^{(1)}||^2}r^{(0)}$$

und damit

$$x^{(0)} + |c_1|^2(\tilde{x}^{(1)} - x^{(0)}) = x^{(0)} + |c_1|^2\overline{\alpha_0}r^{(0)} = x^{(0)} + \frac{||w^{(0)}||^2}{||w^{(0)}||^2 + ||w^{(1)}||^2}\overline{\alpha_0}r^{(0)} = x^{(1)}.$$

Die Behauptung ist also für den Fall $m = 1$ erfüllt. Im Folgenden sei daher $m \geq 2$ angenommen. Es gilt

$$\hat{R}_m = \begin{pmatrix} R_m \\ 0 \end{pmatrix} = \Omega_m \begin{pmatrix} Q_{m-1}H_m \\ 0...0\ h_{m+1,m} \end{pmatrix} = \begin{pmatrix} I_{m-1} & & \\ & c_m & \overline{s_m} \\ & -s_m & \overline{c_m} \end{pmatrix} \begin{pmatrix} Q_{m-1}H_m \\ 0...0\ h_{m+1,m} \end{pmatrix},$$

also

$$R_m = \begin{pmatrix} I_{m-1} & & \\ & c_m & \overline{s_m} \end{pmatrix} \begin{pmatrix} Q_{m-1}H_m \\ 0...0\ h_{m+1,m} \end{pmatrix}. \tag{3.26}$$

Wegen

$$(Q_{m-1}H_m)_{mm} = t_m = \frac{|t_m|^2 t_m + t_m|h_{m+1,m}|^2}{|t_m|^2 + |h_{m+1,m}|^2} = |c_m|^2 t_m + \overline{c_m s_m}h_{m+1,m}$$

und weil $Q_{m-1}H_m \in \mathbb{C}^{m\times m}$ eine obere Dreiecksmatrix ist, folgt also mit (3.26)

$$\begin{pmatrix} I_{m-1} & \\ & \overline{c_m} \end{pmatrix} R_m = \begin{pmatrix} I_{m-1} & & \\ & |c_m|^2 & \overline{c_m s_m} \end{pmatrix} \begin{pmatrix} Q_{m-1}H_m \\ 0...0\ h_{m+1,m} \end{pmatrix} = Q_{m-1}H_m. \tag{3.27}$$

Da die untere Dreiecksmatrix H_m wegen $||w^{(i)}|| \neq 0\ \forall i \in \{0, ..., m\}$ regulär ist und Q_{m-1} unitär ist, muss auch $Q_{m-1}H_m = \begin{pmatrix} I_{m-1} & \\ & \overline{c_m} \end{pmatrix} R_m$ regulär sein. Damit ist $c_m \neq 0$.
Wegen (3.25) gilt außerdem $\tilde{y}^{(m)} = ||r^{(0)}||H_m^{-1}e_1$. Damit folgt

$$\begin{aligned}
\tilde{y}^{(m)} &= H_m^{-1}||r^{(0)}||e_1 = H_m^{-1}Q_{m-1}^{-1}Q_{m-1}||r^{(0)}||e_1 = (Q_{m-1}H_m)^{-1}\hat{g}^{(m-1)} \\
&\overset{(3.27)}{=} \left[\begin{pmatrix} I_{m-1} & \\ & \overline{c_m} \end{pmatrix} R_m\right]^{-1}\hat{g}^{(m-1)} = R_m^{-1}\begin{pmatrix} I_{m-1} & \\ & \overline{c_m}^{-1} \end{pmatrix}\hat{g}^{(m-1)} \\
&= R_m^{-1}\begin{pmatrix} I_{m-1} & \\ & \overline{c_m}^{-1} \end{pmatrix}\begin{pmatrix} g^{(m-1)} \\ \gamma_m \end{pmatrix} = R_m^{-1}\begin{pmatrix} g^{(m-1)} \\ \overline{c_m}^{-1}\gamma_m \end{pmatrix} \\
&\overset{(3.22)}{=} R_m^{-1}\begin{pmatrix} R_{m-1}y^{(m-1)} \\ \overline{c_m}^{-1}\gamma_m \end{pmatrix}.
\end{aligned} \tag{3.28}$$

Wegen $R_m = \begin{pmatrix} R_{m-1} & * \\ & r_{mm} \end{pmatrix}$ folgt damit

$$\begin{aligned}
y^{(m)} &\overset{(3.22)}{=} R_m^{-1} g^{(m)} \overset{(3.21)}{=} R_m^{-1} \begin{pmatrix} g^{(m-1)} \\ c_m \gamma_m \end{pmatrix} \overset{(3.22)}{=} R_m^{-1} \begin{pmatrix} R_{m-1} y^{(m-1)} \\ c_m \gamma_m \end{pmatrix} \\
&= R_m^{-1} \Big[(1 - |c_m|^2) \begin{pmatrix} R_{m-1} y^{(m-1)} \\ 0 \end{pmatrix} + |c_m|^2 \begin{pmatrix} R_{m-1} y^{(m-1)} \\ \gamma_m \overline{c_m}^{-1} \end{pmatrix} \Big] \\
&= R_m^{-1} \Big[(1 - |c_m|^2) R_m \begin{pmatrix} y^{(m-1)} \\ 0 \end{pmatrix} + |c_m|^2 \begin{pmatrix} R_{m-1} y^{(m-1)} \\ \overline{c_m}^{-1} \gamma_m \end{pmatrix} \Big] \\
&= (1 - |c_m|^2) \begin{pmatrix} y^{(m-1)} \\ 0 \end{pmatrix} + |c_m|^2 R_m^{-1} \begin{pmatrix} R_{m-1} y^{(m-1)} \\ \overline{c_m}^{-1} \gamma_m \end{pmatrix} \\
&\overset{(3.28)}{=} (1 - |c_m|^2) \begin{pmatrix} y^{(m-1)} \\ 0 \end{pmatrix} + |c_m|^2 \tilde{y}^{(m)}. \qquad (3.29)
\end{aligned}$$

Also gilt für die m-te TFQMR-Iterierte

$$\begin{aligned}
x^{(m)} &\overset{(3.20)}{=} x^{(0)} + U_m y^{(m)} \overset{(3.29)}{=} x^{(0)} + U_m \Big[(1 - |c_m|^2) \begin{pmatrix} y^{(m-1)} \\ 0 \end{pmatrix} + |c_m|^2 \tilde{y}^{(m)} \Big] \\
&= (1 - |c_m|^2) x^{(0)} + |c_m|^2 x^{(0)} + U_m \Big[(1 - |c_m|^2) \begin{pmatrix} y^{(m-1)} \\ 0 \end{pmatrix} + |c_m|^2 \tilde{y}^{(m)} \Big] \\
&= (1 - |c_m|^2) \Big[x^{(0)} + U_m \begin{pmatrix} y^{(m-1)} \\ 0 \end{pmatrix} \Big] + |c_m|^2 (x^{(0)} + U_m \tilde{y}^{(m)}) \\
&\overset{(3.23)}{=} (1 - |c_m|^2) x^{(m-1)} + |c_m|^2 \tilde{x}^{(m)} = x^{(m-1)} + |c_m|^2 (\tilde{x}^{(m)} - x^{(m-1)}).
\end{aligned}$$

□

Die m-te Iterierte $x^{(m)}$ des TFQMR-Verfahrens steht zu der m-ten Iterierten $\tilde{x}^{(m)}$ des CGS-Verfahrens also in einer engen Beziehung, welche wir nun verwenden können, um eine rekursive Darstellung für $x^{(m)}$ herzuleiten. Wir definieren

$$d^{(m)} := \frac{1}{\overline{\alpha_{m-1}}} (\tilde{x}^{(m)} - x^{(m-1)}), \quad d^{(0)} := 0,$$

$$\eta_m := |c_m|^2 \overline{\alpha_{m-1}}, \quad \eta_0 := 0.$$

Da wegen Satz 3.7 gerade $(\tilde{x}^{(m)} - x^{(m-1)}) = \frac{1}{|c_m|^2} (x^{(m)} - x^{(m-1)})$ gilt, erhalten wir somit

$$\begin{aligned}
x^{(m-1)} + \eta_m d^{(m)} &= x^{(m-1)} + \frac{|c_m|^2 \overline{\alpha_{m-1}}}{\overline{\alpha_{m-1}}} (\tilde{x}^{(m)} - x^{(m-1)}) \\
&= x^{(m-1)} + \frac{|c_m|^2 \overline{\alpha_{m-1}}}{|c_m|^2 \overline{\alpha_{m-1}}} (x^{(m)} - x^{(m-1)}) = x^{(m)}.
\end{aligned}$$

Unser nächstes Ziel besteht nun darin, eine rekursive Darstellung für $d^{(m)}$ zu finden.

Satz 3.8 *Es gilt:*

$$d^{(m)} = u^{(m-1)} + \frac{(1-|c_{m-1}|^2)\eta_{m-1}}{|c_{m-1}|^2\overline{\alpha_{m-1}}} d^{(m-1)}.$$

Beweis. Dieser Beweis basiert auf [6, S.250].
Es ist nach (3.23) und (3.24) $\tilde{x}^{(1)} = x^{(0)} + \overline{\alpha_0} u^{(0)}$, also folgt wegen $d^{(0)} = 0$

$$d^{(1)} = \frac{1}{\overline{\alpha_0}}(\tilde{x}^{(1)} - x^{(0)}) = \frac{1}{\overline{\alpha_0}}(\overline{\alpha_0} u^{(0)}) = u^{(0)} = u^{(0)} + \frac{(1-|c_0|^2)\eta_0}{|c_0|^2\overline{\alpha_0}} d^{(0)}.$$

Die Behauptung gilt also für den Fall $m = 1$. Im Folgenden sei also angenommen, dass $m \geq 2$ ist. Nach (3.13) bzw. (3.23) und (3.24) gilt $\tilde{x}^{(m)} = \tilde{x}^{(m-1)} + \overline{\alpha_{m-1}} u^{(m-1)}$. Also folgt mit der Definition von $d^{(m)}$

$$\begin{aligned}
d^{(m)} &= \frac{1}{\overline{\alpha_{m-1}}}(\tilde{x}^{(m)} - x^{(m-1)}) \\
&= \frac{1}{\overline{\alpha_{m-1}}}(\tilde{x}^{(m)} - \tilde{x}^{(m-1)} + \tilde{x}^{(m-1)} - x^{(m-1)}) \\
&= \frac{1}{\overline{\alpha_{m-1}}}(\overline{\alpha_{m-1}} u^{(m-1)} + \tilde{x}^{(m-1)} - x^{(m-1)}) \\
&= u^{(m-1)} + \frac{1}{\overline{\alpha_{m-1}}}(\tilde{x}^{(m-1)} - x^{(m-1)}) \\
&= u^{(m-1)} + \frac{1}{\overline{\alpha_{m-1}}}(\tilde{x}^{(m-1)} - x^{(m-1)} - x^{(m-2)} + x^{(m-2)}) \\
&= u^{(m-1)} + \frac{1}{\overline{\alpha_{m-1}}}(\tilde{x}^{(m-1)} - x^{(m-2)} - [x^{(m-1)} - x^{(m-2)}]) \\
&\overset{\text{Satz 3.7}}{=} u^{(m-1)} + \frac{1}{\overline{\alpha_{m-1}}}(\tilde{x}^{(m-1)} - x^{(m-2)} - |c_{m-1}|^2[\tilde{x}^{(m-1)} - x^{(m-2)}]) \\
&= u^{(m-1)} + \frac{1-|c_{m-1}|^2}{\overline{\alpha_{m-1}}}(\tilde{x}^{(m-1)} - x^{(m-2)}).
\end{aligned}$$

Außerdem gilt mit der Definition von $d^{(m-1)}$ und η_{m-1}

$$\begin{aligned}
\frac{(1-|c_{m-1}|^2)\eta_{m-1}}{|c_{m-1}|^2\overline{\alpha_{m-1}}} d^{(m-1)} &= \frac{(1-|c_{m-1}|^2)|c_{m-1}|^2\overline{\alpha_{m-2}}}{|c_{m-1}|^2\overline{\alpha_{m-1}}} \frac{1}{\overline{\alpha_{m-2}}}(\tilde{x}^{(m-1)} - x^{(m-2)}) \\
&= \frac{1-|c_{m-1}|^2}{\overline{\alpha_{m-1}}}(\tilde{x}^{(m-1)} - x^{(m-2)}).
\end{aligned}$$

□

Mit Hilfe von Satz 3.7 können wir außerdem das folgende Hilfsresultat beweisen.

Satz 3.9 *Für das m-te CGS-Residuum $w^{(m)} = b - A\tilde{x}^{(m)}$ gilt:*

$$||w^{(m)}|| = |\gamma_m| \frac{|s_m|}{|c_m|}.$$

Beweis. Dieser Beweis basiert auf dem Beweis von Satz 4.101 aus [5, S.213-217].
Durch direktes Nachrechnen erhalten wir

$$c_1 = \frac{|\alpha_0|}{\alpha_0}\frac{||w^{(0)}||}{\sqrt{||w^{(0)}||^2 + ||w^{(1)}||^2}}, \quad s_1 = -\frac{|\alpha_0|}{\overline{\alpha_0}}\frac{||w^{(1)}||}{\sqrt{||w^{(0)}||^2 + ||w^{(1)}||^2}}.$$

Damit folgt sofort wegen $r^{(0)} = w^{(0)}$ und $\gamma_1 = ||r^{(0)}||$

$$|\gamma_1|^2\frac{|s_1|^2}{|c_1|^2} = ||r^{(0)}||^2\frac{||w^{(1)}||^2}{||w^{(0)}||^2} = ||r^{(0)}||^2\frac{||w^{(1)}||^2}{||r^{(0)}||^2} = ||w^{(1)}||^2.$$

Die Behauptung ist also für den Fall $m = 1$ erfüllt. Daher nehmen wir im Folgenden an, dass $m \geq 2$ ist. Wie in (3.25) bereits eingesehen, gilt $H_m\tilde{y}^{(m)} = ||r^{(0)}||e_1$. Also folgt

$$\left|\left|||r^{(0)}||e_1 - \hat{H}_m\tilde{y}^{(m)}\right|\right| = \left|\left|(0, ..., 0, ||w^{(m)}||)^T\right|\right| = ||w^{(m)}||.$$

Mit Satz 3.7 gilt also

$$\begin{aligned}||w^{(m)}||^2 = \left|\left|||r^{(0)}||e_1 - \hat{H}_m\tilde{y}^{(m)}\right|\right|^2 = \left|\left|Q_m\left[||r^{(0)}||e_1 - \hat{H}_m\tilde{y}^{(m)}\right]\right|\right|^2 = \left|\left|\hat{g}^{(m)} - \hat{R}_m\tilde{y}^{(m)}\right|\right|^2 \\ \overset{(3.21)}{=} \left|\left|\begin{pmatrix} g^{(m-1)} \\ c_m\gamma_m \\ -s_m\gamma_m \end{pmatrix} - \begin{pmatrix} R_m\tilde{y}^{(m)} \\ 0 \end{pmatrix}\right|\right|^2 \overset{\text{Satz 3.7}}{=} \left|\left|\begin{pmatrix} g^{(m-1)} \\ c_m\gamma_m \\ -s_m\gamma_m \end{pmatrix} - \begin{pmatrix} g^{(m-1)} \\ \overline{c_m}^{-1}\gamma_m \\ 0 \end{pmatrix}\right|\right|^2 \\ = |\gamma_m(c_m - \overline{c_m}^{-1})|^2 + |-s_m\gamma_m|^2 = |\gamma_m|^2(|c_m - \overline{c_m}^{-1}|^2 + |s_m|^2).\end{aligned}$$

Außerdem gilt

$$\begin{aligned}|c_m - \overline{c_m}^{-1}|^2 = (c_m - \overline{c_m}^{-1})\overline{(c_m - \overline{c_m}^{-1})} = (c_m - \overline{c_m}^{-1})(\overline{c_m} - c_m^{-1}) = |c_m|^2 - 1 - 1 + |c_m^{-1}|^2 \\ = |c_m|^2 + |c_m|^{-2} - 2. \qquad (3.30)\end{aligned}$$

Wegen $|c_m|^2 + |s_m|^2 = 1$ folgt also

$$\begin{aligned}|\gamma_m|^2(|c_m - \overline{c_m}^{-1}|^2 + |s_m|^2) \overset{(3.30)}{=} |\gamma_m|^2(|c_m|^2 + |c_m|^{-2} - 2 + |s_m|^2) = |\gamma_m|^2(|c_m|^{-2} - 1) \\ = |\gamma_m|^2\Big(\frac{1}{|c_m|^2} - 1\Big) = |\gamma_m|^2\Big(\frac{1}{|c_m|^2} - \frac{|c_m|^2}{|c_m|^2}\Big) = |\gamma_m|^2\Big(\frac{1 - |c_m|^2}{|c_m|^2}\Big) = |\gamma_m|^2\frac{|s_m|^2}{|c_m|^2}.\end{aligned}$$

□

Wir definieren nun die Norm des aktuellen Quasi-Residuums des TFQMR-Verfahrens durch

$$\tau_m := \left|\left|||r^{(0)}||e_1 - \hat{H}_m y^{(m)}\right|\right|, \quad \tau_0 := ||r^{(0)}||.$$

Darauf basierend definieren wir außerdem die Hilfsgröße

$$\theta_m := \frac{||w^{(m)}||}{\tau_{m-1}}, \quad \theta_0 := 0.$$

Lemma 3.10 *Es gilt:*

$$\tau_m = \tau_{m-1}\theta_m|c_m|.$$

Beweis. Dieser Beweis basiert auf dem Beweis von Satz 4.101 aus [5, S.213-217].
Mit der gleichen Argumentation wie bei dem GMRES-Verfahren und dem QMR-Verfahren gilt gerade

$$\tau_m = |\gamma_{m+1}| \text{ und } \tau_{m-1} = |\gamma_m|.$$

Damit folgt direkt mit Satz 3.9 und (3.21)

$$\tau_m = |\gamma_{m+1}| = |-s_m\gamma_m| = |s_m||\gamma_m| = |s_m|\tau_{m-1} \overset{\text{Satz 3.9}}{=} \frac{||w^{(m)}||}{|\gamma_m|}|c_m|\tau_{m-1} = \theta_m|c_m|\tau_{m-1}.$$

□

Lemma 3.11 *Es gilt:*

$$\theta_m^2 = \frac{1-|c_m|^2}{|c_m|^2} = \frac{1}{|c_m|^2} - 1.$$

Beweis. Dieser Beweis basiert auf dem Beweis von Satz 7.18 aus [4, S.293-296].
Wir verwenden erneut Satz 3.9 und die Identität $\tau_{m-1} = |\gamma_m|$ und erhalten

$$\theta_m^2 = \frac{||w^{(m)}||^2}{\tau_{m-1}^2} \overset{\text{Satz 3.9}}{=} \frac{|\gamma_m|^2|s_m|^2}{|c_m|^2|\gamma_m|^2} = \frac{|s_m|^2}{|c_m|^2} = \frac{1-|c_m|^2}{|c_m|^2} = \frac{1}{|c_m|^2} - 1.$$

□

Mit dieser Hilfsaussage können wir nun neue Darstellungen von $|c_m|$ und $d^{(m)}$ gewinnen.

$$|c_m| \overset{\text{Lemma 3.11}}{=} \frac{1}{\sqrt{\theta_m^2+1}},$$

$$d^{(m)} \overset{\text{Satz 3.8}}{=} u^{(m-1)} + \frac{1-|c_{m-1}|^2}{|c_{m-1}|^2}\frac{\eta_{m-1}}{\overline{\alpha_{m-1}}}d^{(m-1)} \overset{\text{Lemma 3.11}}{=} u^{(m-1)} + \theta_{m-1}^2\frac{\eta_{m-1}}{\overline{\alpha_{m-1}}}d^{(m-1)}.$$

Wir fassen nun die bisher gewonnenen Darstellungen für die Aktualisierungen der benötigten Skalare und Vektoren zusammen.

Bemerkung 3.12 (Rekursionsformeln) *Wir haben bisher gezeigt:*

- $x^{(m)} = x^{(m-1)} + \eta_m d^{(m)}$
- $\eta_m = |c_m|^2\overline{\alpha_{m-1}}$
- $d^{(m)} = u^{(m-1)} + \theta_{m-1}^2\frac{\eta_{m-1}}{\overline{\alpha_{m-1}}}d^{(m-1)}$
- $|c_m| = \frac{1}{\sqrt{\theta_m^2+1}}$
- $\theta_m = \frac{||w^{(m)}||}{\tau_{m-1}}$
- $\tau_m = \tau_{m-1}\theta_m|c_m|$
- $w^{(m)} = w^{(m-1)} - \overline{\alpha_{m-1}}Au^{(m-1)}$

Sofern uns $x^{(m-1)}, \eta_{m-1}, d^{(m-1)}, \tau_{m-1}, \theta_{m-1}, w^{(m-1)}, u^{(m-1)}$ und α_{m-1} zur Verfügung stehen, können wir also $w^{(m)}, \theta_m, |c_m|, \tau_m, \eta_m, d^{(m)}$ und damit insbesondere die m-te TFQMR-Näherungslösung $x^{(m)}$ berechnen.

Für eine vollständige, rekursive Konstruktion von $x^{(m)}$ müssen wir also zu guter Letzt noch diskutieren, wie α_m und $u^{(m)}$ berechnet werden können.
Dazu verwenden wir erneut die Darstellungen (3.4) - (3.12) für die Skalare und Vektoren des CGS-Algorithmus. Für $j \in \mathbb{N}_0$ definieren wir den Hilfsvektor

$$v^{(2j)} := Ap^{(2j)}.$$

Dann gilt wegen $p^{(0)} = u^{(0)}$ gerade $v^{(0)} = Ap^{(0)} = Au^{(0)}$ und für $j \geq 1$ erhalten wir

$$v^{(2j)} = Ap^{(2j)} \overset{(3.11)}{=} Au^{(2j)} + \overline{\beta_{2j-2}}(Aq^{(2j-2)} + \overline{\beta_{2j-2}}Ap^{(2j-2)}),$$

also wegen $u^{(2j-1)} = q^{(2j-2)}$ nach (3.12)

$$v^{(2j)} = Au^{(2j)} + \overline{\beta_{2j-2}}(Au^{(2j-1)} + \overline{\beta_{2j-2}}v^{(2j-2)}).$$

Weiter folgt nun mit (3.5), (3.10) und (3.12) für $j \in \mathbb{N}_0$

$$u^{(2j+1)} = q^{(2j)} = u^{(2j)} - \overline{\alpha_{2j}}v^{(2j)},$$

$$u^{(2j+2)} = w^{(2j+2)} + \overline{\beta_{2j}}u^{(2j+1)}.$$

Damit erhalten wir nun insgesamt folgende Berechnungsvorschrift für $u^{(m)}$:

$$u^{(m)} := \begin{cases} u^{(m-1)} - \overline{\alpha_{m-1}}v^{(m-1)} & \text{, falls m ungerade} \\ w^{(m)} + \overline{\beta_{m-2}}u^{(m-1)} & \text{, falls m gerade} \end{cases}$$

mit $u^{(0)} = p^{(0)} = r^{(0)}$ und $v^{(0)} = Au^{(0)}$.
Dabei berechnen wir β_{m-2} wie zuvor in (3.9) wegen $\tilde{r}^{(0)} = r^{(0)}$ mittels

$$\beta_{m-2} = \langle w^{(m)}, r^{(0)} \rangle / \langle w^{(m-2)}, r^{(0)} \rangle.$$

Außerdem können wir α_m nach (3.4) und (3.12) wegen $\tilde{r}^{(0)} = r^{(0)}$ nun vermöge

$$\alpha_m := \begin{cases} \langle w^{(m)}, r^{(0)} \rangle / \langle v^{(m)}, r^{(0)} \rangle & \text{, falls m gerade} \\ \alpha_{m-1} & \text{, falls m ungerade} \end{cases} \tag{3.31}$$

mit $\alpha_0 = \langle r^{(0)}, r^{(0)} \rangle / \langle v^{(0)}, r^{(0)} \rangle$ berechnen.
Damit steht uns nun alles zur Verfügung, um den TFQMR-Algorithmus zu formulieren. Für das Abbruchkriterium wählen wir wie bei dem QMR-Verfahren auch eine Fehlerschranke $\epsilon \in \mathbb{R}_{>0}$ und brechen ab, sobald die relative Quasi-Residuennorm diese unterschreitet, d.h sobald

$$\frac{\tau_m}{\tau_0} = \frac{|\gamma_{m+1}|}{||r^{(0)}||} \leq \epsilon$$

ist.
Da c_m nicht direkt für den Algorithmus benötigt wird, sondern lediglich dessen Betrag $|c_m|$, schreiben wir im Algorithmus c_m für $|c_m|$.

Damit haben wir unser Ziel erreicht: Genau wie bei dem QMR-Algorithmus sind auch bei dem TFQMR-Algorithmus sowohl der Speicherbedarf als auch der pro Schritt erforderliche Rechenaufwand unabhängig von der Anzahl der durchgeführten Iterationen und da $Au^{(m)}$ in einem Hilfsvektor gespeichert werden kann, ist die einzige Matrix-Vektor-Multiplikation, die in einem TFQMR-Schritt benötigt wird, eine Multiplikation mit A. Wir haben also eine Variante des QMR-Verfahrens konstruiert, die ohne Zugriff auf die Adjungierte A^* auskommt.

Algorithmus 3.13 ***(TFQMR)***

$r^{(0)} \leftarrow b - Ax^{(0)}; w^{(0)} \leftarrow r^{(0)}; u^{(0)} \leftarrow r^{(0)}; v^{(0)} \leftarrow Au^{(0)}; d^{(0)} \leftarrow 0;$
$\theta_0 \leftarrow 0; \eta_0 \leftarrow 0; \tau_0 \leftarrow ||r^{(0)}||; \rho_0 \leftarrow \langle r^{(0)}, r^{(0)} \rangle; \alpha_0 \leftarrow \rho_0 / \langle v^{(0)}, r^{(0)} \rangle;$
$m \leftarrow 0;$
while $\tau_m > \tau_0 \epsilon$ **do**
    $m \leftarrow m + 1;$
    $w^{(m)} \leftarrow w^{(m-1)} - \overline{\alpha_{m-1}} Au^{(m-1)};$
    $d^{(m)} \leftarrow u^{(m-1)} + \theta_{m-1}^2 (\eta_{m-1} / \overline{\alpha_{m-1}}) d^{(m-1)};$
    $\theta_m \leftarrow ||w^{(m)}|| / \tau_{m-1};$
    $c_m \leftarrow 1/\sqrt{\theta_m^2 + 1};$
    $\tau_m \leftarrow \tau_{m-1} \theta_m c_m;$
    $\eta_m \leftarrow c_m^2 \overline{\alpha_{m-1}};$
    $x^{(m)} \leftarrow x^{(m-1)} + \eta_m d^{(m)};$
    if m ungerade **then**
        $\alpha_m \leftarrow \alpha_{m-1};$
        $u^{(m)} \leftarrow u^{(m-1)} - \overline{\alpha_{m-1}} v^{(m-1)};$
    end if
    if m gerade **then**
        $\rho_m \leftarrow \langle w^{(m)}, r^{(0)} \rangle;$
        $\beta_{m-2} \leftarrow \rho_m / \rho_{m-2};$
        $u^{(m)} \leftarrow w^{(m)} + \overline{\beta_{m-2}} u^{(m-1)};$
        $v^{(m)} \leftarrow Au^{(m)} + \overline{\beta_{m-2}} (Au^{(m-1)} + \overline{\beta_{m-2}} v^{(m-2)});$
        $\alpha_m \leftarrow \rho_m / \langle v^{(m)}, r^{(0)} \rangle;$
    end if
end while

3.4 Breakdown des TFQMR-Verfahrens

Für die Durchführung des m-ten Schrittes des TFQMR-Verfahrens haben wir bisher vorausgesetzt, dass der CGS-Algorithmus nicht vor der Berechnung der benötigten Skalare und Vektoren abbricht, sodass $\alpha_0, ..., \alpha_{m-1} \neq 0$ sind. Außerdem haben wir vorausgesetzt, dass $w^{(0)}, ..., w^{(m)} \neq 0$ sind.

Wir untersuchen nun zunächst den Fall, dass das aktuelle CGS-Residuum $w^{(m)}$ gleich 0 ist, während alle anderen Voraussetzungen weiterhin erfüllt sind. Dann können wir O.B.d.A. annehmen, dass $w^{(0)}, ..., w^{(m-1)} \neq 0$ gilt, sodass die Matrix

$$\Lambda_m := \begin{pmatrix} ||w^{(0)}|| & & & \\ & ||w^{(1)}|| & & \\ & & \ddots & \\ & & & ||w^{(m-1)}|| \end{pmatrix} \in \mathbb{C}^{m \times m}$$

invertierbar ist.
Außerdem definieren wir nun $B_m \in \mathbb{C}^{m \times m}$ als die Matrix $\hat{B}_m$ ohne dessen letzte Zeile und

$$W_m := (w^{(0)} w^{(1)} ... w^{(m-1)}) \in \mathbb{C}^{n \times m}.$$

Damit erhalten wir wegen $w^{(m)} = 0$ nun sowohl

$$W_{m+1} \hat{B}_m = (W_m 0) \hat{B}_m = W_m B_m$$

als auch

$$\begin{pmatrix} H_m \\ 0 \end{pmatrix} = \hat{H}_m = \Lambda_{m+1} \hat{B}_m = \begin{pmatrix} \Lambda_m & \\ & 0 \end{pmatrix} \hat{B}_m = \begin{pmatrix} \Lambda_m B_m \\ 0 \end{pmatrix}.$$

Für ein beliebiges $x = x^{(0)} + U_m y \in x^{(0)} + K_m(A, r^{(0)})$ gilt also wegen $r^{(0)} = w^{(0)}$

$$\begin{aligned} b - Ax = b - Ax^{(0)} - AU_m y &\overset{\text{Satz 3.4}}{=} r^{(0)} - W_{m+1} \hat{B}_m y = r^{(0)} - W_m B_m y \\ = W_m [e_1 - B_m y] = W_m \Lambda_m^{-1} \Lambda_m [e_1 - B_m y] &= W_m \Lambda_m^{-1} \Big[||r^{(0)}|| e_1 - \Lambda_m B_m y \Big] \\ &= W_m \Lambda_m^{-1} \Big[||r^{(0)}|| e_1 - H_m y \Big] \end{aligned} \tag{3.32}$$

und damit

$$||b - Ax|| = \Big|\Big| W_m \Lambda_m^{-1} \Big[||r^{(0)}|| e_1 - H_m y \Big] \Big|\Big| \leq \Big|\Big| W_m \Lambda_m^{-1} \Big|\Big| \ \Big|\Big| ||r^{(0)}|| e_1 - H_m y \Big|\Big|.$$

Wegen $\hat{H}_m = \begin{pmatrix} H_m \\ 0 \end{pmatrix}$ gilt außerdem für alle $y \in \mathbb{C}^m$ gerade

$$\Big|\Big| ||r^{(0)}|| e_1 - \hat{H}_m y \Big|\Big| = \Big|\Big| ||r^{(0)}|| e_1 - H_m y \Big|\Big|,$$

also folgt für die m-te TFQMR-Iterierte $x^{(m)} = x^{(0)} + U_m y^{(m)}$ mit

$$min \Big\{ \Big|\Big| ||r^{(0)}|| e_1 - \hat{H}_m y \Big|\Big| \mid y \in \mathbb{C}^m \Big\} = \Big|\Big| ||r^{(0)}|| e_1 - \hat{H}_m y^{(m)} \Big|\Big| = \tau_m$$

somit

$$\begin{aligned} ||b - Ax^{(m)}|| &\leq \Big|\Big| W_m \Lambda_m^{-1} \Big|\Big| \ \Big|\Big| ||r^{(0)}|| e_1 - H_m y^{(m)} \Big|\Big| \\ &= \Big|\Big| W_m \Lambda_m^{-1} \Big|\Big| \ \Big|\Big| ||r^{(0)}|| e_1 - \hat{H}_m y^{(m)} \Big|\Big| = \Big|\Big| W_m \Lambda_m^{-1} \Big|\Big| \tau_m. \end{aligned} \tag{3.33}$$

Wegen $w^{(0)}, ..., w^{(m-1)} \neq 0$ ist die untere Dreiecksmatrix

$$H_m = \begin{pmatrix} ||w^{(0)}||/\overline{\alpha_0} & & & & \\ -||w^{(1)}||/\overline{\alpha_0} & ||w^{(1)}||/\overline{\alpha_1} & & & \\ & -||w^{(2)}||/\overline{\alpha_1} & ||w^{(2)}||/\overline{\alpha_2} & & \\ & & \cdots & & \\ & & & -||w^{(m-1)}||/\overline{\alpha_{m-2}} & ||w^{(m-1)}||/\overline{\alpha_{m-1}} \end{pmatrix}$$

außerdem weiterhin regulär. Damit ist auch die obere Dreiecksmatrix $Q_{m-1}H_m$ regulär und somit $t_m = (Q_{m-1}H_m)_{mm} \neq 0$. Damit ist auch R_m wegen

$$(R_m)_{mm} = \sqrt{|t_m|^2 + |h_{m+1,m}|^2} = \sqrt{|t_m|^2} = |t_m| > 0$$

regulär. Da H_m und R_m also auch in diesem Fall invertierbar sind, gelten Satz 3.7, Satz 3.8, Satz 3.9, Lemma 3.10, Lemma 3.11 und die daraus resultierenden Rekursionsformeln mit analoger Beweisführung weiterhin. Insbesondere haben wir wegen

$$\theta_m = \frac{||w^{(m)}||}{\tau_{m-1}} = 0$$

mit Lemma 3.10 also

$$\tau_m = \tau_{m-1}\theta_m|c_m| = 0.$$

Damit folgt nun mit (3.33)

$$||b - Ax^{(m)}|| \leq ||W_m\Lambda_m^{-1}||\tau_m = 0.$$

Die m-te TFQMR-Näherungslösung $x^{(m)}$ ist also die exakte Lösung. Wir halten fest:

Satz 3.14 (Lucky Breakdown) *Angenommen, der CGS-Algorithmus breche nicht vorzeitig ab. Ist $w^{(m)} = 0$, so ist die m-te TFQMR-Iterierte $x^{(m)}$ die exakte Lösung.*

Gilt hingegen $\alpha_{m-1} = 0$, so ist der m-te Schritt des TFQMR-Verfahrens gar nicht durchführbar. Falls $m-1$ ungerade ist, gilt nach (3.31) $\alpha_{m-1} = \alpha_{m-2}$, sodass wir uns im Folgenden auf den Fall, dass $m-1$ gerade ist, beschränken können. Dann gilt ebenso nach (3.31)

$$0 = \alpha_{m-1} = \langle w^{(m-1)}, r^{(0)}\rangle / \langle v^{(m-1)}, r^{(0)}\rangle \iff \langle w^{(m-1)}, r^{(0)}\rangle = 0.$$

Es gilt $\langle w^{(m-1)}, r^{(0)}\rangle = 0$ unter anderem dann, wenn $w^{(m-1)} = 0$ ist. Dieser Fall ist jedoch völlig unproblematisch:
Falls $m - 1 = 0$ ist, ist unser Startresiduum $r^{(0)} = w^{(0)}$ bereits 0, sodass unser Startvektor $x^{(0)}$ bereits die exakte Lösung ist und das TFQMR-Verfahren bereits vor der Durchführung des 1. Schrittes abbricht. Gilt hingegen $m - 1 \geq 1$, so folgt mit Satz 3.14, dass die zuletzt berechnete Näherungslösung $x^{(m-1)}$ bereits die exakte Lösung ist.
Falls $w^{(m-1)} = 0$ ist, liegt also ein Lucky Breakdown vor. Natürlich kann es aber auch vorkommen, dass $\langle w^{(m-1)}, r^{(0)}\rangle = 0$ und $w^{(m-1)} \neq 0$ gilt. In diesem Fall lässt sich leider keine vergleichbare Aussage über die Genauigkeit der aktuellen Näherungslösungen treffen und der m-te TFQMR-Schritt lässt sich dennoch nicht durchführen, sodass ein Serious Breakdown entsteht. Genauso verhält es sich, falls $\alpha_{m-1} = \langle w^{(m-1)}, r^{(0)}\rangle / \langle v^{(m-1)}, r^{(0)}\rangle$ wegen $\langle v^{(m-1)}, r^{(0)}\rangle = 0$ nicht berechnet werden kann.

3.5 Konvergenzeigenschaften des TFQMR-Verfahrens

Im Folgenden nehmen wir wieder an, dass das CGS-Verfahren nicht vor der Konstruktion der im m-ten TFQMR-Schritt benötigten Skalare und Vektoren abbricht bzw. $\alpha_0, ..., \alpha_{m-1} \neq 0$ sind und dass $w^{(0)}, ..., w^{(m)} \neq 0$ sind.

Lemma 3.15 *Für das m-te TFQMR-Residuum $r^{(m)} = b - Ax^{(m)}$ gilt*

$$||r^{(m)}|| \leq \sqrt{m+1}\ \tau_m = \sqrt{m+1}\ |s_1 s_2 ... s_m|\ ||r^{(0)}||.$$

Beweis. Dieser Beweis ist eine Kombination der Beweise von Satz 2.14 und Satz 2.15. Nach der Definition von Λ_{m+1} und W_{m+1} sind die Spalten von $W_{m+1}\Lambda_{m+1}^{-1}$ normiert. Analog zu Satz 2.15 gilt also

$$||W_{m+1}\Lambda_{m+1}^{-1}|| \leq \sqrt{m+1}.$$

Außerdem gilt nach (3.18) für $x^{(m)} = x^{(0)} + U_m y^{(m)}$

$$||b - Ax^{(m)}|| \leq \left|\left|W_{m+1}\Lambda_{m+1}^{-1}\right|\right|\ \left|\left|||r^{(0)}||e_1 - \hat{H}_m y^{(m)}\right|\right| = \left|\left|W_{m+1}\Lambda_{m+1}^{-1}\right|\right|\ \tau_m.$$

Damit folgt also insgesamt

$$||b - Ax^{(m)}|| \leq \sqrt{m+1}\ \tau_m.$$

Weiter gilt nach (3.21) für alle $i \in \{1, ..., m\}$ gerade $\gamma_{i+1} = -s_i\gamma_i$. Wegen $\gamma_1 = ||r^{(0)}||$ folgt damit

$$\tau_m = |\gamma_{m+1}| = |(-1)^m s_1 s_2 ... s_m \gamma_1| = |s_1 s_2 ... s_m||r^{(0)}||\ | = |s_1 s_2 ... s_m|\ ||r^{(0)}||.$$

□

Kombinieren wir (3.2) mit den durch die Verdopplung der Indizes gewählten Schreibweisen (3.5),(3.10) und (3.12), so bemerken wir, dass für alle $i \in \{0, ..., m\}$ ein Polynom s_i des Grades i existiert, welches folgende Bedingung erfüllt:

$$u^{(i)} = s_i(A)r^{(0)}.$$

Da für alle $i \in \{1, ..., m\}$ nach (3.15) wiederum

$$w^{(i)} = w^{(i-1)} - \overline{\alpha_{i-1}}Au^{(i-1)}$$

und $w^{(0)} = r^{(0)} = u^{(0)}$ gilt, folgt selbiges induktiv damit auch für $w^{(0)}, ..., w^{(m)}$.Wir können also analog zu Satz 3.3 folgendes festhalten:

Lemma 3.16 *Es gilt*

$$K_{m+1}(A, r^{(0)}) = span\{w^{(0)}, ..., w^{(m)}\}.$$

Insbesondere ist $\{w^{(0)}, ..., w^{(m)}\}$ eine Basis von $K_{m+1}(A, r^{(0)})$, falls $dim(K_{m+1}(A, r^{(0)})) = m + 1$ ist.

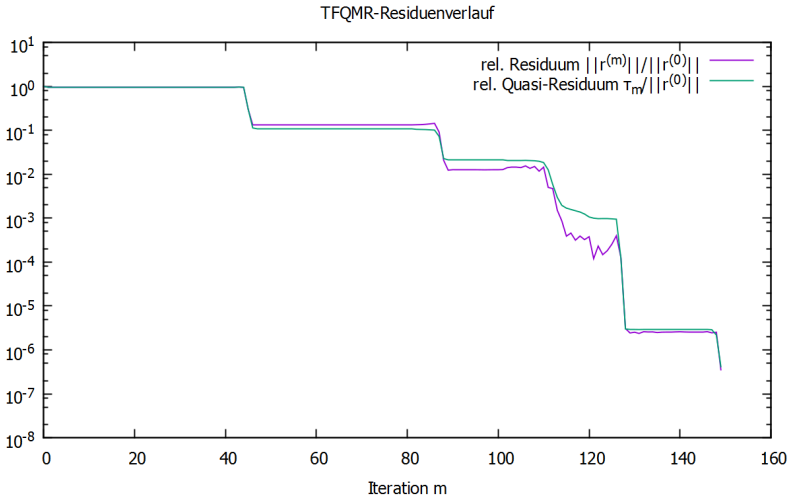
Solange wir die Krylow-Räume noch nicht ausgeschöpft haben, d.h solange $m+1 \leq m_0$ ist, sind $w^{(0)}, ..., w^{(m)}$ also linear unabhängig.
Damit können wir nun völlig analog zum QMR-Verfahren die Normen der TFQMR-Residuen mit den Normen der GMRES-Residuen vergleichen.

Satz 3.17 *Seien $x_T^{(m)}$ die m-te Näherungslösung des TFQMR-Verfahrens und $x_G^{(m)}$ die m-te Näherungslösung des GMRES-Verfahrens. Solange $w^{(0)}, ..., w^{(m)}$ linear unabhängig sind, gilt:*

$$||b - Ax_T^{(m)}|| \leq \kappa(W_{m+1}\Lambda_{m+1}^{-1}) \, ||b - Ax_G^{(m)}||.$$

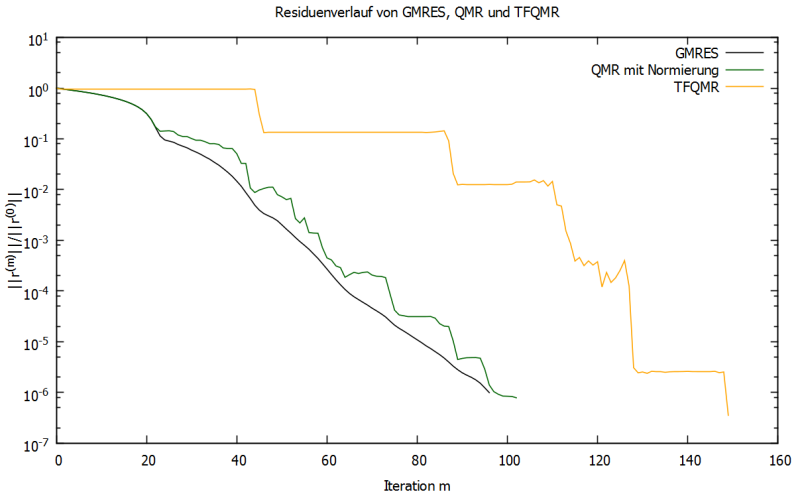
Beweis. Dieser Beweis ist eine Abwandlung des Beweises von Satz 2.17.
Man ersetze im Beweis von Satz 2.17 $x_Q^{(m)}$ durch $x_T^{(m)}$, $r_Q^{(m)}$ durch $r_T^{(m)}$, $v_1, ..., v_{m+1}$ durch $\frac{w^{(0)}}{||w^{(0)}||}, ..., \frac{w^{(m)}}{||w^{(m)}||}$, V_{m+1} durch $W_{m+1}\Lambda_{m+1}^{-1}$ und $\hat{T}_m$ durch $\hat{H}_m$.

□



Wie wir in Satz 3.7 nachgewiesen haben, hängen die TFQMR-Näherungslösungen eng mit den Iterierten des CGS-Verfahrens zusammen. Das TFQMR-Verfahren kombiniert also das stark oszillierende Verhalten des CGS-Verfahrens mit dem Glättungseffekt, der durch die Quasi-Minimierung entsteht. Der resultierende Residuenverlauf weist daher lange Stagnationsphasen auf, stets gefolgt von einem rasanten Abstieg.
Auffällig ist außerdem, dass das TFQMR-Verfahren zum Erreichen des von uns gewählten Abbruchkriteriums wesentlich mehr Iterationsschritte ausführt als das QMR-Verfahren und etwa doppelt so viele Schritte wie das CGS-Verfahren. Dies ist allerdings schlicht darauf zurückzuführen, dass wir für die Konstruktion des TFQMR-Verfahrens die ursprünglichen CGS-Schritte jeweils in zwei Schritte aufgeteilt haben. Eine TFQMR-Iteration entspricht in dieser Hinsicht sozusagen einem halben CGS-Schritt. Dafür erfordert ein Schritt des TFQMR-Verfahrens im Gegensatz zu QMR und CGS nicht zwei, sondern lediglich eine aufwändige Matrix-Vektor-Multiplikation.

4 Fazit



Das QMR-Verfahren liefert im Vergleich zum GMRES-Verfahren durchaus zufriedenstellende Ergebnisse. Wird eine Normierung der benötigten Basisvektoren vorgenommen, können die durch die Quasi-Minimierung entstehenden Oszillationen im Residuenverlauf eingeschränkt werden, sodass trotz Verzicht auf Orthonormalbasen ein verhältnismäßig glatter Verlauf entsteht. In unserem Modellproblem erreicht das QMR-Verfahren dabei eine sehr ähnliche Genauigkeit wie das GMRES-Verfahren und benötigt dabei nur wenige Iterationen mehr. Das QMR-Verfahren liefert also offenbar qualitativ ähnlich gute Resultate wie das GMRES-Verfahren. Problematisch ist neben dem möglichen Serious Breakdown vor allem, dass in jeder Iteration neben einer Matrix-Vektor-Multiplikation mit A auch eine solche mit A^* durchgeführt werden muss.
Das TFQMR-Verfahren erfordert je Iteration wie GMRES hingegen nur eine Multiplikation mit A und erreicht bis zum Abbruch ebenfalls das Genauigkeitsniveau von GMRES. Hierfür werden allerdings deutlich mehr Iterationen und somit auch mehr Matrix-Vektor-Multiplikationen benötigt. Außerdem erbt das TFQMR-Verfahren von

Tabelle 4.1: Ergebnisse der Verfahren nach Erreichen des Abbruchkriteriums

	Iterationen bis zum Abbruch	**relative Residuennorm**
GMRES	96	$9.86549 \cdot 10^{-7}$
QMR(normiert)	102	$7.82374 \cdot 10^{-7}$
TFQMR	149	$3.52758 \cdot 10^{-7}$

dem CGS-Verfahren mehrere Szenarien für einen Serious Breakdown und einen ziemlich unvorteilhaften Residuenverlauf mit langen Stagnationsphasen.
Dieses Problem kann vermindert werden, indem statt dem CGS-Verfahren das sogenannte *BiCGSTAB-Verfahren (BiCG stabilized)* verwendet wird. Hierbei handelt es sich um eine Variante des CGS-Verfahrens, die einen wesentlich glatteren Residuenverlauf aufweist, siehe [5, S.204-210]. Auf Grundlage des BiCGSTAB-Verfahrens kann auf sehr ähnliche Weise wie bei dem TFQMR-Verfahren das *QMRCGSTAB-Verfahren* hergeleitet werden, siehe [5, S.219-222].
Trotz jener Nachteile haben sowohl das QMR-Verfahren als auch das TFQMR-Verfahren dem GMRES-Verfahren wie erhofft zwei entscheidende Vorteile voraus: Der Rechenaufwand ist in jeder Iteration gleich und der Speicherbedarf wächst nicht mit der Anzahl der Iterationen. Während ersteres aufgrund der zusätzlich anfallenden Matrix-Vektor-Multiplikationen in erster Linie für Gleichungssysteme mit schwach besetzten Matrizen interessant sein dürfte, ist letzteres in jedem Fall die Lösung für ein wesentliches Problem des GMRES-Verfahrens bei großen Problemdimensionen. Damit sind beide Verfahren durchaus eine sinnvolle Alternative für die Behandlung großer Gleichungssysteme.

Literaturverzeichnis

[1] Börm, Steffen: *Iterative Lösungsverfahren für große lineare Gleichungssysteme.* Vorlesungsskript für die Vorlesung „Iterative Verfahren für große Gleichungssysteme“ an der Christian-Albrechts-Universität zu Kiel, Januar 2017. – Version vom 31.01.2017

[2] Clason, Christian: *Iterative Verfahren für lineare Gleichungssysteme und Eigenwertprobleme.* Vorlesungsskript für die Vorlesung „Iterative Verfahren für lineare Gleichungssysteme und Eigenwertprobleme“ an der Universität Duisburg-Essen, Juli 2016. – Version vom 20.07.2016

[3] Freund, Roland W.: A Transpose-Free Quasi-Minimal Residual Algorithm for Non-Hermitian Linear Systems. In: *SIAM Journal on Scientific Computing* 14 (1993), Nr. 2, 470–482. `http://dx.doi.org/10.1137/0914029`. – DOI `10.1137/0914029`

[4] Kanzow, Christian: *Numerik linearer Gleichungssysteme: Direkte und iterative Verfahren.* Springer, 2005

[5] Meister, Andreas: *Numerik linearer Gleichungssysteme.* Springer Spektrum, 2015

[6] Saad, Yousef: *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, 2003

[7] Sabine Le Borne: *Ein Vergleich CG - ähnlicher Verfahren zur Lösung indefiniter Probleme.* Kiel, Christian Albrechts Universität zu Kiel, Diplomarbeit, 1994

Alle Programme, die für die Durchführung der numerischen Experimente verwendet wurden, wurden auf Grundlage der Software-Bibliothek „H2Lib“ der Arbeitsgruppe „Scientific Computing“ der Christian-Albrechts-Universität zu Kiel erstellt. Siehe hierfür http://www.h2lib.org/ .

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Weiterhin versichere ich, dass diese Arbeit noch nicht als Abschlussarbeit an anderer Stelle vorgelegen hat.

09.04.2018,

Janne Henningsen