

Numerische Lineare Algebra

Oliver Ernst

Professur Numerische Mathematik

Wintersemester 2017/18



Mathematik!
TU Chemnitz

① Einleitung

- 1.1 Lineare Gleichungssysteme
- 1.2 Matrixfunktionen
- 1.3 Modellreduktion
- 1.4 Eigenwertaufgaben

② Krylov-Unterraumverfahren

- 2.1 Projektionen
- 2.2 Orthogonale Projektionsverfahren
- 2.3 Krylov-Unterräume
- 2.4 Basen von Krylov-Räumen

③ Lineare Gleichungssysteme

- 3.1 Lösungsstrategien
- 3.2 Selbstadjungierte Probleme
- 3.3 Das Verfahren der konjugierten Gradienten
- 3.4 Krylov-Verfahren für lineare Ausgleichsprobleme
- 3.5 Vorkonditionierung

④ Matrixfunktionen

- 4.1 Erste Definition mithilfe der Jordanschen Normalform

- 4.2 Hermitesche Polynominterpolation
- 4.3 Alternative Darstellungen von Matrixfunktionen
- 4.4 Resolventenintegrale
- 4.5 Ein Beispiel

5 Krylov-Verfahren für Matrixfunktionen

- 5.1 Schranken für $\|f(\mathbf{A})\|$
- 5.2 Fehlerschranken für Krylov-Verfahren
- 5.3 Die Konvergenz des CG-Verfahrens

6 Das QR-Verfahren für Eigenwertaufgaben

- 6.1 Reduktion auf Hessenberg-Gestalt
- 6.2 Vektoriteration
- 6.3 QR-Iteration

- ① Einleitung
- ② Krylov-Unterraumverfahren
- ③ **Lineare Gleichungssysteme**
- ④ Matrixfunktionen
- ⑤ Krylov-Verfahren für Matrixfunktionen
- ⑥ Das QR-Verfahren für Eigenwertaufgaben

Wir betrachten lineare Gleichungssysteme

$$\mathbf{Ax} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{C}^{N \times N} \text{ invertierbar}, \quad \mathbf{b} \in \mathbb{C}^N, \quad N \text{ „groß“}. \quad (3.1)$$

- Erste iterative Lösungsverfahren: Ausgleichsrechnungen in Astronomie und Vermessungswesen, frühes 19. Jahrhundert (Gauß, Seidel, Jacobi, Nagel, ...)
- 1940er Jahre (Southwell, Young, Varga, Golub, ...) Anwendung dieser **Relaxationsverfahren** auf diskretisierte partielle Differentialgleichungen der mathematischen Physik, systematische Weiterentwicklung.
- 1952 **CG-Verfahren** von Hestenes und Stiefel veröffentlicht, gleichzeitig veröffentlichte Lanczos verwandte Methoden für Gleichungssysteme und Eigenwertaufgaben.
- In [Reid, 1971] sowie [Concus, Golub & O'Leary, 1976] wurde das Potential der Kombination des CG-Verfahrens mit **Vorkonditionierung** erkannt.
- Verwandte Verfahrensklasse: **semi-iterative Verfahren** [Golub & Varga, 1961], [Varga, 1962], [Eiermann, Niethammer & Varga, 1985].
- Seither wurde das CG-Verfahren systematisch verallgemeinert und analysiert, bleibt aber der prominenteste Vertreter der **Kylov-Unterraumverfahren**.

- 1 Einleitung
- 2 Krylov-Unterraumverfahren
- 3 Lineare Gleichungssysteme
 - 3.1 Lösungsstrategien
 - 3.2 Selbstadjungierte Probleme
 - 3.3 Das Verfahren der konjugierten Gradienten
 - 3.4 Krylov-Verfahren für lineare Ausgleichsprobleme
 - 3.5 Vorkonditionierung
- 4 Matrixfunktionen
- 5 Krylov-Verfahren für Matrixfunktionen
- 6 Das QR-Verfahren für Eigenwertaufgaben

Iterationsverfahren: sukzessive Verbesserung einer **Startnäherung** $\mathbf{x}_0 \approx \mathbf{A}^{-1}\mathbf{b}$ durch additive **Korrektur** $\mathbf{x}_0 + \mathbf{v}$. Diese ist „exakt“, falls

$$\mathbf{A}(\mathbf{x}_0 + \mathbf{v}) = \mathbf{b} \quad \Leftrightarrow \quad \mathbf{A}\mathbf{v} = \mathbf{b} - \mathbf{A}\mathbf{x}_0 =: \mathbf{r}_0.$$

Krylov-Unterraumverfahren bestimmen im m -ten Schritt eine Korrektur $\mathbf{v} \in \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$, d.h. eine Näherung

$$\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0), \quad m = 1, 2, \dots, d = d(\mathbf{A}, \mathbf{r}_0).$$

Lemma 3.1

Ist \mathbf{A} invertierbar, so gilt $d(\mathbf{A}, \mathbf{r}_0) = \min\{m \in \mathbb{N} : \mathbf{A}^{-1}\mathbf{r}_0 \in \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)\}$.

Fazit: Die exakte Lösung $\mathbf{A}^{-1}\mathbf{b}$ liegt in $\mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{x}_0)$ für $m = d(\mathbf{A}, \mathbf{r}_0)$, aber für kein kleineres m .

Frage: Wahl der Korrektur in $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$.

Bei **MR-Verfahren** (minimales Residuum, oder auch „Kleinste-Quadrate-Methoden“) bestimmt man $\mathbf{x}_m^{\text{MR}} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ so, dass

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}_m^{\text{MR}}\| = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|. \quad (3.2)$$

Dabei bezeichnet $\|\cdot\|$ eine Norm auf \mathbb{C}^N , die von einem Innenprodukt (\cdot, \cdot) induziert wird, also $\|\mathbf{v}\| = (\mathbf{v}, \mathbf{v})^{1/2}$.

Bei **OR-Verfahren** (orthogonales Residuum, oder auch Galerkin-Methoden) bestimmt man $\mathbf{x}_m^{\text{OR}} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ so, dass

$$\mathbf{b} - \mathbf{A}\mathbf{x}_m^{\text{OR}} \perp \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0). \quad (3.3)$$

Dabei bezeichnet \perp Orthogonalität bezüglich eines gegebenen Innenprodukts (\cdot, \cdot) auf \mathbb{C}^N .

Wir halten zunächst fest, dass für jedes Krylov-Unterraum-Verfahren

$$\begin{aligned} \mathbf{r}_m &:= \mathbf{b} - \mathbf{A}\mathbf{x}_m \in \mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)) \\ &= \mathbf{r}_0 + \mathbf{A}\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) \subset \mathcal{K}_{m+1}(\mathbf{A}, \mathbf{r}_0) \end{aligned}$$

gilt. Beide Verfahren (MR- und OR-) brechen nach endlich vielen Schritten ab, weil sie dann die exakte Lösung gefunden haben.

Lemma 3.2

Es sei $d = d(\mathbf{A}, \mathbf{r}_0)$ der Index, an dem die Folge der Krylov-Räume $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ stationär wird (vgl. (2.4)). Dann gelten

$$\mathbf{x}_d^{\text{OR}} = \mathbf{x}_d^{\text{MR}} = \mathbf{x}_* = \mathbf{A}^{-1}\mathbf{b}.$$

Für $m < d$ ist sowohl $\mathbf{x}_m^{\text{OR}} \neq \mathbf{x}_*$ als auch $\mathbf{x}_m^{\text{MR}} \neq \mathbf{x}_*$.

Beide Näherungen, MR- und OR-Approximationen, lassen sich mit Hilfe der Arnoldi-Zerlegung von \mathbf{A} bezüglich $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ bestimmen:

Satz 3.3

Es sei $\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\mathbf{H}_{m+}$ die Arnoldi-Zerlegung (2.6) von \mathbf{A} bezüglich $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$. Durch die Bedingung (3.2) wird eine eindeutig bestimmte MR-Approximation

$$\mathbf{x}_m^{\text{MR}} = \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m^{\text{MR}}$$

festgelegt, wobei der Koordinatenvektor $\mathbf{y}_m^{\text{MR}} \in \mathbb{C}^m$ die eindeutige Lösung des linearen Ausgleichsproblems

$$\|\beta \mathbf{e}_1 - \mathbf{H}_{m+} \mathbf{y}\|_2 \rightarrow \min_{\mathbf{y} \in \mathbb{C}^m}$$

ist. Dabei ist $\beta = \|\mathbf{r}_0\|$ und $\mathbf{e}_1 \in \mathbb{R}^{m+1}$ bezeichnet den ersten Koordinateneinheitsvektor. Mit anderen Worten:

$$\mathbf{x}_m^{\text{MR}} = \mathbf{x}_0 + \beta \mathbf{V}_m (\mathbf{H}_{m+}^H \mathbf{H}_{m+})^{-1} \mathbf{H}_{m+}^H \mathbf{e}_1.$$

Satz 3.4

Es sei $AV_m = V_m H_m + h_{m+1,m+1} v_{m+1} e_m^T$ die Arnoldi-Zerlegung (2.6) von A bezüglich $\mathcal{K}_m(A, r_0)$. Die OR-Approximation

$$x_m^{\text{OR}} = x_0 + V_m y_m^{\text{OR}}$$

existiert genau dann, wenn H_m invertierbar ist. In diesem Fall ist sie eindeutig bestimmt. Der Koordinatenvektor $y_m^{\text{OR}} \in \mathbb{C}^m$ löst das lineare Gleichungssystem

$$H_m y = \beta e_1.$$

Dabei ist $\beta = \|r_0\|$ und $e_1 \in \mathbb{R}^m$ bezeichnet den ersten Koordinateneinheitsvektor. Mit anderen Worten:

$$x_m^{\text{OR}} = x_0 + \beta V_m H_m^{-1} e_1.$$

Korollar 3.5

Existiert die OR-Approximation \mathbf{x}_m^{OR} , so gilt für ihr Residuum

$$\mathbf{r}_m^{\text{OR}} = \mathbf{b} - \mathbf{A}\mathbf{x}_m^{\text{OR}} = -\beta h_{m+1,m} (\mathbf{e}_m^{\text{T}} \mathbf{H}_m^{-1} \mathbf{e}_1) \mathbf{v}_{m+1}$$

und

$$\|\mathbf{r}_m^{\text{OR}}\| = \beta h_{m+1,m} |\mathbf{e}_m^{\text{T}} \mathbf{H}_m^{-1} \mathbf{e}_1|.$$

Wie beschreiben im Folgenden die praktische Berechnung der MR-Näherung \mathbf{x}_m^{MR} (und werden später sehen, dass man, daraus auch \mathbf{x}_m^{OR} gewinnen kann).

- Das lineare Ausgleichsproblem aus Satz 3.3 kann mit Hilfe einer QR-Zerlegung

$$\mathbf{Q}_m \mathbf{H}_{m+} = \begin{bmatrix} \mathbf{R}_m \\ \mathbf{0} \end{bmatrix} \quad (3.4)$$

der Hessenberg-Matrix \mathbf{H}_{m+} gelöst werden. Hier ist $\mathbf{Q}_m \in \mathbb{C}^{(m+1) \times (m+1)}$ unitär ($\mathbf{Q}_m^H \mathbf{Q}_m = \mathbf{I}_{m+1}$) und $\mathbf{R}_m \in \mathbb{C}^{m \times m}$ besitzt obere Dreiecksform.

- Setzt man diese QR-Zerlegung in das lineare Ausgleichsproblem aus Satz 3.3 ein, so ergibt sich

$$\begin{aligned} \min_{\mathbf{y} \in \mathbb{C}^m} \|\beta \mathbf{e}_1 - \mathbf{H}_{m+} \mathbf{y}\|_2 &= \min_{\mathbf{y} \in \mathbb{C}^m} \left\| \mathbf{Q}_m^H \left(\beta \mathbf{Q}_m \mathbf{e}_1 - \begin{bmatrix} \mathbf{R}_m \\ \mathbf{0} \end{bmatrix} \mathbf{y} \right) \right\|_2 \\ &= \min_{\mathbf{y} \in \mathbb{C}^m} \left\| \beta \mathbf{Q}_m \mathbf{e}_1 - \begin{bmatrix} \mathbf{R}_m \\ \mathbf{0} \end{bmatrix} \mathbf{y} \right\|_2 = \min_{\mathbf{y} \in \mathbb{C}^m} \left\| \begin{bmatrix} \beta \mathbf{q}_m - \mathbf{R}_m \mathbf{y} \\ \beta \eta_m \end{bmatrix} \right\|_2, \end{aligned}$$

wobei $[\mathbf{q}_m^T, \eta_m]^T := \mathbf{Q}_m \mathbf{e}_1 \in \mathbb{C}^{m+1}$ für die erste Spalte von \mathbf{Q}_m steht.

- Weil \mathbf{H}_{m+} vollen Rang besitzt, ist \mathbf{R}_m invertierbar und unser Ausgleichsproblem wird durch $\mathbf{y}_m^{\text{MR}} = \beta \mathbf{R}_m^{-1} \mathbf{q}_m$ gelöst.
- Der zugehörige kleinste-Quadrate-Fehler beträgt $\|\mathbf{r}_m^{\text{MR}}\| = \beta |\eta_m|$.

Lemma 3.6

Mit den obigen Bezeichnungen ist die MR-Approximation $\mathbf{x}_m^{\text{MR}} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ gegeben durch

$$\mathbf{x}_m^{\text{MR}} = \mathbf{x}_0 + \beta \mathbf{V}_m \mathbf{R}_m^{-1} \mathbf{q}_m.$$

Die zugehörige Residualnorm ist

$$\|\mathbf{r}_m^{\text{MR}}\| = \beta |\eta_m|.$$

- Die QR-Zerlegung (3.4) von \mathbf{H}_{m+} bestimmt man mittels **Givens-Rotationen**. Die unitäre Matrix \mathbf{Q}_m wird also als Produkt von Rotationsmatrizen dargestellt:

$$\mathbf{Q}_m = \mathbf{G}_m \begin{bmatrix} \mathbf{G}_{m-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{G}_{m-2} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_2 \end{bmatrix} \cdots \begin{bmatrix} \mathbf{G}_1 & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_{m-1} \end{bmatrix}. \quad (3.5)$$

- Dazu wählen wir die Faktoren

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & c_k & s_k e^{-i\varphi_k} \\ \mathbf{0} & -s_k e^{i\varphi_k} & c_k \end{bmatrix} \in \mathbb{C}^{(k+1) \times (k+1)} \quad (3.6)$$

mit $c_k, s_k \geq 0$ ($c_k^2 + s_k^2 = 1$), $\varphi_k \in \mathbb{R}$ ($k = 1, 2, \dots, m$)

so, dass bei der unitären Transformation von \mathbf{H}_{m+} auf obere Dreiecksgestalt der $(m+1, m)$ -Eintrag vernullt wird:

- Angenommen wir haben die Koeffizienten $h_{2,1}, \dots, h_{m-1,m-2}, h_{m,m-1}$ bereits vernullt, d.h. wir haben Rotationen $\mathbf{G}_1, \dots, \mathbf{G}_{m-2}, \mathbf{G}_{m-1}$ konstruiert mit

$$\mathbf{G}_{m-1} \begin{bmatrix} \mathbf{G}_{m-2} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \cdots \begin{bmatrix} \mathbf{G}_1 & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_{m-2} \end{bmatrix} \mathbf{H}_{(m-1)+} = \begin{bmatrix} \mathbf{R}_{m-1} \\ \mathbf{0} \end{bmatrix}$$

bzw.

$$\begin{bmatrix} \mathbf{G}_{m-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{G}_{m-2} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_2 \end{bmatrix} \cdots \begin{bmatrix} \mathbf{G}_1 & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_{m-1} \end{bmatrix} \mathbf{H}_{m+} = \begin{bmatrix} \mathbf{R}_{m-1} & \mathbf{r} \\ \mathbf{0} & \tau \\ \mathbf{0} & h_{m+1,m} \end{bmatrix}.$$

- Dann setzen wir

$$c_m := \frac{|\tau|}{\sqrt{|\tau|^2 + h_{m+1,m}^2}}, \quad s_m := \frac{h_{m+1,m}}{\sqrt{|\tau|^2 + h_{m+1,m}^2}}, \quad (3.7)$$

$$\varphi_m := \arg(h_{m+1,m}) - \arg(\tau) = -\arg(\tau)$$

(denn $h_{m+1,m} > 0$) und verifizieren durch einfaches Nachrechnen

$$\begin{bmatrix} \mathbf{I}_{m-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & c_m & s_m e^{-i\varphi_m} \\ \mathbf{0} & -s_m e^{i\varphi_m} & c_m \end{bmatrix} \begin{bmatrix} \mathbf{R}_{m-1} & \mathbf{r} \\ \mathbf{0} & \tau \\ \mathbf{0} & h_{m+1,m} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{m-1} & \mathbf{r} \\ \mathbf{0} & r_{m,m} \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_m \\ \mathbf{0} \end{bmatrix}$$

mit $r_{m,m} = \sqrt{|\tau|^2 + h_{m+1,m}^2} e^{-i\varphi_m}$.

Die erste Spalte von \mathbf{Q}_m ist von besonderer Bedeutung (vgl. Lemma 3.6). Sie lässt sich rekursiv berechnen:

Lemma 3.7

Zerlegt man die erste Spalte von \mathbf{Q}_m gemäß $\mathbf{Q}_m \mathbf{e}_1 = \begin{bmatrix} \mathbf{q}_m \\ \eta_m \end{bmatrix}$ mit $\mathbf{q}_m \in \mathbb{C}^m$, so gelten

$$\mathbf{q}_m = \begin{bmatrix} \mathbf{q}_{m-1} \\ c_m \eta_{m-1} \end{bmatrix} \quad \text{und} \quad \eta_m = -s_m e^{i\varphi_m} \eta_{m-1}, \quad m = 2, 3, \dots$$

mit den Anfangsbedingungen $\mathbf{q}_1 = c_1$ und $\eta_1 = -s_1 e^{i\varphi_1}$.

Um einen Algorithmus für ein Krylov-basiertes MR-Verfahren zu konstruieren, braucht man nur noch die Einzelteile zusammensetzen, nämlich

1. das Arnoldi-Verfahren zur Erzeugung der orthonormalen Basisvektoren v_1, v_2, \dots und der Hessenberg-Matrix H_{m+} ,
2. die QR-Zerlegung (3.4) von H_{m+} , aus der wir die obere Dreiecksmatrix R_m ablesen sowie den Vektor $\beta \begin{bmatrix} q_m \\ \eta_m \end{bmatrix} := \beta Q_m e_1 \in \mathbb{C}^{m+1}$, dessen letzte Komponente uns die Residualnorm $\|r_m^{\text{MR}}\|$ liefert, und
3. die Lösung des Dreieckssystems $R_m y_m^{\text{MR}} = q_m$, die auf die m -te Näherung $x_m^{\text{MR}} = x_0 + V_m y_m^{\text{MR}}$ führt, wobei dieser Schritt nur dann durchgeführt werden muss, wenn wir an der Näherung x_m^{MR} interessiert sind (also, wenn $\|r_m^{\text{MR}}\|$ genügend klein ist).

Der resultierende Algorithmus 5 ist als **Generalized Minimal Residual Method** (GMRES) bekannt [Saad & Schultz, 1986] und – verglichen etwa mit dem Verfahren der konjugierten Gradienten – erstaunlicherweise relativ neuen Datums.

Algorithmus 5: Generalized Minimal Residual Method (GMRES)

Gegeben: $A \in \mathbb{C}^{N \times N}$, $b \in \mathbb{C}^N$, $x_0 \approx A^{-1}b$, Innenprodukt (\cdot, \cdot) auf \mathbb{C}^N , $\|\cdot\| = \sqrt{(\cdot, \cdot)}$.

1 $r_0 \leftarrow b - Ax_0$, $\beta \leftarrow \|r_0\|$, $v_1 \leftarrow r_0/\beta$, $u_1 \leftarrow \beta$

2 **for** $m = 1, 2, \dots$ **do**

3 Konstruiere mit dem Arnoldi-Prozess eine Arnoldi-Zerlegung (2.6)

4 $\tau \leftarrow h_{1,m}$

5 **for** $j=1, 2, \dots, m-1$ **do**

6 $r_{j,m} \leftarrow c_j \tau + s_j \exp(-i\varphi_j) h_{j+1,m}$

7 $\tau \leftarrow -s_j \exp(i\varphi_j) \tau + c_j h_{j+1,m}$

8 $d \leftarrow \sqrt{|\tau|^2 + h_{m+1,m}^2}$

9 $\varphi_m \leftarrow -\arg(\tau)$, $c_m \leftarrow |\tau|/d$, $s_m \leftarrow h_{m+1,m}/d$, $r_{m,m} \leftarrow d \exp(-i\varphi_m)$,

10 $t \leftarrow u_m$, $u_m \leftarrow c_m t$, $u_{m+1} \leftarrow -s_m \exp(i\varphi_m) t$, $\|r_m\| \leftarrow |u_{m+1}|$

11 **if** $\|r_m\| \leq \varepsilon \|b\|$ **then**

12 $\text{con} = 1$, **break**

13 **if** $\text{con} = 1$ **then**

14 Löse das Dreieckssystem $Ry_m = [u_1, u_2, \dots, u_m]^T$

15 $x_m \leftarrow x_0 + [v_1 \ v_2 \ \dots \ v_m] y_m$

16 $\|r_m\| \leftarrow \|b - Ax_m\|$

Bemerkung 3.8

- (1) In Algorithmus 5 kann die Arnoldi-Zerlegung mit dem einfachen Arnoldi-Verfahren (Alg. 1), dem modifizierten Arnoldi-Verfahren (Alg. 2) oder dem Arnoldi-Verfahren mit Reorthogonalisierung (Alg. 3) konstruiert werden.
- (2) Es scheint überflüssig zu sein, nach der Berechnung von \mathbf{x}_m noch einmal die Residualnorm $\|\mathbf{r}_m\| = \|\mathbf{b} - \mathbf{A}\mathbf{x}_m\|$ zu bestimmen, weil sie vorher über die Formel $\|\mathbf{r}_m\| = |u_{m+1}|$ ermittelt wurde. In Gleitpunktarithmetik führt die Berechnung via $\|\mathbf{r}_m\| = |u_{m+1}|$ nicht immer zu korrekten Ergebnissen.
- (3) Wir halten fest, dass die Residualnorm wegen der Lemmata 3.6 und 3.7 auch durch $\|\mathbf{r}_m\| = s_m \|\mathbf{r}_{m-1}\|$ aufdatiert werden kann.
- (4) Dass die orthonormalen Vektoren $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{m+1}$ (das sind $m + 1$ „lange“, d.h. n -dimensionale, i.Allg. vollbesetzte Vektoren) alle bereit gehalten werden müssen, verursacht den hauptsächlichen Speicheraufwand von GMRES. Auch der Rechenaufwand resultiert i.W. aus dem Arnoldi-Teil: ein Matrix-Vektor-Produkt mit \mathbf{A} , m Vektoraufdatierungen von Vektoren der Dimension n sowie $m + 1$ Innenprodukte solcher Vektoren müssen im m -ten Schritt bestimmt werden.

Bemerkung 3.8 (Fortsetzung)

- (5) Im Gegensatz zu beispielsweise den klassischen Iterationsverfahren sind sowohl Speicher- als auch Rechenaufwand bei GMRES nicht konstant pro Iterationsschritt: Spätere Iterationsschritte sind wesentlich teurer als die ersten, so dass es in der Regel nicht möglich ist, sehr viele GMRES-Schritte durchzuführen.
- (6) Aus diesem Grund wird oft ein k so gewählt, dass der Aufwand von k GMRES-Schritten (gerade noch) vertretbar ist, um dann das Verfahren nach k Schritten abzubrechen. Ist die Qualität der Näherungslösung x_k nicht zufriedenstellend (was in der Regel der Fall sein wird), so startet man GMRES erneut mit der Startnäherung x_k und führt wieder k GMRES-Schritte durch. Dies kann mehrmals wiederholt werden. Man spricht vom **neugestarteten GMRES** (engl. restarted GMRES) und bezeichnet dieses Verfahren mit $\text{GMRES}(k)$.

Ein Krylov-basiertes OR-Verfahren lässt sich ähnlich implementieren:

- Das lineare Gleichungssystem $\mathbf{H}_m \mathbf{y}_m^{\text{OR}} = \beta \mathbf{e}_1$ kann mit Hilfe einer QR-Zerlegung von \mathbf{H}_m gelöst werden, die wie bei GMRES durch Givens-Rotationen aufdatiert wird.
- Man erhält auf diese Weise einen Algorithmus, der unter dem Namen **Full Orthogonalization Method** (FOM) bekannt ist [Saad, 1981].
- Die Residualnorm $\|\mathbf{r}_m^{\text{OR}}\| = \|\mathbf{b} - A\mathbf{x}_m^{\text{OR}}\|$ kann durch

$$\|\mathbf{r}_m^{\text{OR}}\| = h_{m+1,m} |\mathbf{e}_m^T \mathbf{y}_m^{\text{OR}}|$$

(vgl. Korollar 3.5) ohne explizite Kenntnis von \mathbf{x}_m^{OR} bestimmt werden.

- Wir gehen auf dieses Verfahren nicht näher ein, weil die FOM-Iterierten – wie wir später sehen werden – leicht aus den GMRES-Näherungen berechnet werden können.

- 1 Einleitung
- 2 Krylov-Unterraumverfahren
- 3 Lineare Gleichungssysteme
 - 3.1 Lösungsstrategien
 - 3.2 Selbstadjungierte Probleme
 - 3.3 Das Verfahren der konjugierten Gradienten
 - 3.4 Krylov-Verfahren für lineare Ausgleichsprobleme
 - 3.5 Vorkonditionierung
- 4 Matrixfunktionen
- 5 Krylov-Verfahren für Matrixfunktionen
- 6 Das QR-Verfahren für Eigenwertaufgaben

- In diesem Abschnitt untersuchen wir, wie sich GMRES vereinfacht, wenn die Koeffizientenmatrix \mathbf{A} selbstadjungiert bezüglich des Innenprodukts (\cdot, \cdot) ist, d.h. wenn $(\mathbf{A}\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{A}\mathbf{y})$ für alle $\mathbf{x}, \mathbf{y} \in \mathbb{C}^N$ gilt.
- Natürlich ist der Spezialfall hermitescher Matrizen, die bezüglich des Euklidischen Innenprodukts selbstadjungiert sind, von besonderer Bedeutung. Wir wissen bereits, dass sich dann das Arnoldi-Verfahren auf den hermiteschen Lanczos-Prozess (Alg. 4) reduziert, was insbesondere bedeutet, dass sich der Basisvektor \mathbf{v}_{m+1} aus seinen beiden Vorgängern \mathbf{v}_m und \mathbf{v}_{m-1} berechnen lässt (siehe (2.9)).
- Ersetzt man jetzt im GMRES-Algorithmus (Alg. 5) die Arnoldi- durch die entsprechende Lanczos-Orthogonalisierung, so reduziert sich der Rechenaufwand zwar beträchtlich, wegen $\mathbf{x}_m^{\text{MR}} = \mathbf{x}_0 + [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_m] \mathbf{y}_m^{\text{MR}}$ werden aber nach wie vor alle Basisvektoren benötigt, so dass der Speicheraufwand unverändert groß bleibt.

Wir betrachten zunächst die QR-Zerlegung der reellen symmetrischen Tridiagonalmatrix \mathbf{T}_{m+} aus (2.8):

$$\mathbf{Q}_m \mathbf{T}_{m+} = \mathbf{G}_m \begin{bmatrix} \mathbf{Q}_{m-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \gamma_1 & \delta_2 & 0 & \dots & 0 \\ \delta_2 & \gamma_2 & \delta_3 & \ddots & \vdots \\ 0 & \delta_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \delta_m \\ 0 & \dots & 0 & \delta_m & \gamma_m \\ 0 & \dots & & 0 & \delta_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_m \\ \mathbf{0} \end{bmatrix}.$$

Jetzt sind \mathbf{Q}_m und \mathbf{R}_m natürlich ebenfalls reell. Das ist einer der wenigen Unterschiede zum allgemeinen (komplexen) Fall, der im letzten Abschnitt beschrieben wurde (vgl. (3.5), (3.6) und (3.7)).

- Die orthogonale Matrix $\mathbf{Q}_m \in \mathbb{R}^{(m+1) \times (m+1)}$ ist ein Produkt von Givens-Rotationen (wie in (3.5)), jede Rotationsmatrix hat analog zu (3.6) die Form

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & c_k & s_k \\ \mathbf{0} & -s_k & c_k \end{bmatrix} \in \mathbb{R}^{(k+1) \times (k+1)} \quad (k = 1, 2, \dots, m)$$

mit $c_k, s_k \in \mathbb{R}$ (möglicherweise negativ), so dass $s_k^2 + c_k^2 = 1$ erfüllt ist.

- Damit

$$\begin{bmatrix} \mathbf{I}_{m-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & c_m & s_m \\ \mathbf{0} & -s_m & c_m \end{bmatrix} \begin{bmatrix} \mathbf{R}_{m-1} & \mathbf{r} \\ \mathbf{0} & \tau \\ \mathbf{0} & \delta_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{m-1} & \mathbf{r} \\ \mathbf{0} & r_{m,m} \\ \mathbf{0} & 0 \end{bmatrix}$$

gilt, setzen wir

$$c_m := \frac{\tau}{\sqrt{\tau^2 + \delta_{m+1}^2}} \quad \text{und} \quad s_m := \frac{\delta_{m+1}}{\sqrt{\tau^2 + \delta_{m+1}^2}}, \quad (3.8)$$

woraus sich $r_{m,m}$ zu

$$r_{m,m} = \sqrt{\tau^2 + \delta_{m+1}^2} \quad \left(> \delta_{m+1} > 0, \quad \text{falls } m < d(\mathbf{A}, \mathbf{r}_0) \right)$$

ergibt.

- Wir erinnern daran, dass sich die MR-Iterierten in der Form

$$\mathbf{x}_m^{\text{MR}} = \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m^{\text{MR}} \quad \text{mit} \quad \mathbf{R}_m \mathbf{y}_m^{\text{MR}} = \beta \mathbf{q}_m$$

darstellen lassen.

- Mit den Richtungsvektoren $\mathbf{P}_m = [\mathbf{p}_1 \ \mathbf{p}_2 \ \cdots \ \mathbf{p}_m] := \mathbf{V}_m \mathbf{R}_m^{-1}$ ergibt sich (vgl. Lemma 3.7)

$$\begin{aligned} \mathbf{x}_m^{\text{MR}} &= \mathbf{x}_0 + \beta \mathbf{P}_m \mathbf{q}_m = \mathbf{x}_0 + \beta [\mathbf{P}_{m-1} \ \mathbf{p}_m] \begin{bmatrix} \mathbf{q}_{m-1} \\ c_m \eta_{m-1} \end{bmatrix} \\ &= \mathbf{x}_0 + \beta \mathbf{P}_{m-1} \mathbf{q}_{m-1} + \beta c_m \eta_{m-1} \mathbf{p}_m = \mathbf{x}_{m-1}^{\text{MR}} + \beta c_m \eta_{m-1} \mathbf{p}_m. \end{aligned}$$

- Die Rekursionsformel für η_m aus Lemma 3.7 vereinfacht sich (weil \mathbf{Q}_m reell ist) zu $\eta_m = -s_m \eta_{m-1}$.
- Wir erinnern außerdem an $r_m^{\text{MR}} = \delta_1 |\eta_m|$ (vgl. Lemma 3.6) mit $\delta_1 = \|\mathbf{r}_0\|$.

Lemma 3.10

Die Richtungsvektoren genügen der Rekursionsformel

$$\mathbf{p}_1 = \mathbf{v}_1 / r_{1,1},$$

$$\mathbf{p}_2 = (\mathbf{v}_2 - r_{1,2}\mathbf{p}_1) / r_{2,2},$$

$$\mathbf{p}_m = (\mathbf{v}_m - r_{m-1,m}\mathbf{p}_{m-1} - r_{m-2,m}\mathbf{p}_{m-2}) / r_{m,m}, \quad m = 3, 4, \dots$$

Algorithmus 6: Minimal Residual Method (MINRES)

Gegeben: $A = A^* \in \mathbb{C}^{N \times N}$ bez. (\cdot, \cdot) auf \mathbb{C}^N , $\|\cdot\| = \sqrt{(\cdot, \cdot)}$, $b \in \mathbb{C}^N$, $x_0 \approx A^{-1}b$.

```
1  $r_0 \leftarrow b - Ax_0$ ,  $\delta_1 \leftarrow \|r_0\|$ ,  $v_1 \leftarrow r_0/\delta_1$ ,  $v_0 \leftarrow \mathbf{0}$ ,  $\eta \leftarrow \delta_1$ 
2 for  $m = 1, 2, \dots$  do
3    $w \leftarrow Av_m$ ,  $\gamma \leftarrow (w, v_m)$ ,  $w \leftarrow w - \gamma v_m - \delta_m v_{m-1}$ ,  $\delta_{m+1} \leftarrow \|w\|$ 
4   if  $\delta_{m+1} = 0$  then
5      $\text{con} = 1$ , break
6    $v_{m+1} \leftarrow w/\delta_{m+1}$ 
7   if  $m = 1$  then
8      $\tau_1 \leftarrow \gamma_1$ ,  $r_{1,1} \leftarrow \sqrt{\tau_1^2 + \delta_2^2}$ ,  $c_1 \leftarrow \tau_1/r_{1,1}$ ,  $s_1 \leftarrow \delta_2/r_{1,1}$ ,  $p_1 \leftarrow v_1/r_{1,1}$ 
9   if  $m = 2$  then
10     $\tau_2 \leftarrow c_1\gamma_2 - s_1\delta_2$ ,  $r_{2,2} \leftarrow \sqrt{\tau_2^2 + \delta_3^2}$ ,  $r_{1,2} \leftarrow c_1\delta_2 + s_1\gamma_2$ ,  $c_2 \leftarrow \tau_2/r_{2,2}$ ,
11     $s_2 \leftarrow \delta_3/r_{2,2}$ ,  $p_2 \leftarrow (v_2 - r_{1,2}p_1)/r_{2,2}$ 
12   if  $m \geq 3$  then
13     $\tau_m \leftarrow c_{m-1}\gamma_m - s_{m-1}c_{m-2}\delta_m$ ,  $r_{m-2,m} \leftarrow s_{m-2}\delta_m$ ;  $r_{m,m} \leftarrow \sqrt{\tau_m^2 + \delta_{m+1}^2}$ ,
14     $r_{m-1,m} \leftarrow c_{m-1}c_{m-2}\delta_m + s_{m-1}\gamma_m$ ,  $c_m \leftarrow \tau_m/r_{m,m}$ ,  $s_m \leftarrow \delta_{m+1}/r_{m,m}$ 
15     $p_m \leftarrow (v_m - r_{m-1,m}p_{m-1} - r_{m-2,m}p_{m-2})/r_{m,m}$ 
16    $x_m \leftarrow x_{m-1} + c_m\eta p_m$ ,  $\eta \leftarrow -s_m\eta$ ,  $\|r_m\| \leftarrow |\eta|$ 
```

- 1 Einleitung
- 2 Krylov-Unterraumverfahren
- 3 Lineare Gleichungssysteme
 - 3.1 Lösungsstrategien
 - 3.2 Selbstadjungierte Probleme
 - 3.3 Das Verfahren der konjugierten Gradienten
 - 3.4 Krylov-Verfahren für lineare Ausgleichsprobleme
 - 3.5 Vorkonditionierung
- 4 Matrixfunktionen
- 5 Krylov-Verfahren für Matrixfunktionen
- 6 Das QR-Verfahren für Eigenwertaufgaben

Lineare Gleichungssysteme

Das Verfahren der konjugierten Gradienten

Das **Verfahren der konjugierten Gradienten** (*engl.* conjugate gradient method) oder kürzer **CG-Verfahren** ist das Krylov-Raum basierte OR-Verfahren zur Lösung von $\mathbf{Ax} = \mathbf{b}$, im Spezialfall, dass \mathbf{A} **selbstadjungiert und positiv definit** ist d.h.

$$(\mathbf{Ax}, \mathbf{x}) > 0 \quad \text{für alle } \mathbf{x} \neq \mathbf{0}.$$

Wir wollen also $\mathbf{x}_m^{\text{CG}} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ bestimmen mit

$$\mathbf{r}_m^{\text{CG}} = \mathbf{b} - \mathbf{Ax}_m^{\text{CG}} \perp \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0).$$

Das **Verfahren der konjugierten Gradienten** (*engl.* conjugate gradient method) oder kürzer **CG-Verfahren** ist das Krylov-Raum basierte OR-Verfahren zur Lösung von $\mathbf{Ax} = \mathbf{b}$, im Spezialfall, dass \mathbf{A} **selbstadjungiert und positiv definit** ist d.h.

$$(\mathbf{Ax}, \mathbf{x}) > 0 \quad \text{für alle } \mathbf{x} \neq \mathbf{0}.$$

Wir wollen also $\mathbf{x}_m^{\text{CG}} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ bestimmen mit

$$\mathbf{r}_m^{\text{CG}} = \mathbf{b} - \mathbf{Ax}_m^{\text{CG}} \perp \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0).$$

Mögliches Vorgehen:

1. Berechne Zerlegung $\mathbf{AV}_m = \mathbf{V}_m \mathbf{T}_m + \delta_{m+1} \mathbf{v}_{m+1} \mathbf{e}_m^T$ mit dem hermiteschen Lanczos-Prozess (Alg. 4),
2. löse LGS $\mathbf{T}_m \mathbf{y}_m^{\text{CG}} = \beta \mathbf{e}_1$,
3. setze $\mathbf{x}_m^{\text{CG}} = \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m^{\text{CG}}$ (vgl. Satz 3.4).

In diesem Abschnitt wird ein effizienterer Algorithmus beschrieben, der (in runnungsfreier Rechnung) zu denselben Näherungen führt.

Zunächst: die symmetrische Tridiagonalmatrix

$$\mathbf{T}_m = \begin{bmatrix} \gamma_1 & \delta_2 & 0 & \dots & 0 \\ \delta_2 & \gamma_2 & \delta_3 & \ddots & \vdots \\ 0 & \delta_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \delta_m \\ 0 & \dots & 0 & \delta_m & \gamma_m \end{bmatrix} \in \mathbb{R}^{m \times m}$$

aus dem hermiteschen Lanczos-Prozess mit \mathbf{A} erbt positiv-Definitheit: wegen

$$[\mathbf{T}_m]_{i,j} = (\mathbf{A}\mathbf{v}_i, \mathbf{v}_j) \quad \text{für alle } 1 \leq i, j \leq m \quad (\text{vgl. (2.7)})$$

ergibt sich für $\mathbf{y} = [y_1, y_2, \dots, y_m]^T \in \mathbb{C}^m \setminus \{\mathbf{0}\}$

$$\mathbf{y}^H \mathbf{T}_m \mathbf{y} = \left(\mathbf{A} \sum_{k=1}^m y_k \mathbf{v}_k, \sum_{k=1}^m y_k \mathbf{v}_k \right) > 0.$$

Eine positiv definite Matrix ist invertierbar und damit ist $\mathbf{T}_m \mathbf{y}_m^{\text{CG}} = \beta \mathbf{e}_1$ (eindeutig) lösbar. Mit anderen Worten: die OR-Näherung \mathbf{x}_m^{CG} existiert für jedes m .

Lineare Gleichungssysteme

Das Verfahren der konjugierten Gradienten

Dass \mathbf{T}_m positiv definit ist hat eine weitere Konsequenz: \mathbf{T}_m und damit \mathbf{T}_{m+} besitzen (rationale) Cholesky-Zerlegungen (siehe z.B. [Golub & Van Loan, 1983, § 4.2.3])

$$\mathbf{T}_m = \mathbf{L}_m \mathbf{D}_m \mathbf{L}_m^\top \quad \text{bzw.} \quad \mathbf{T}_{m+} = \mathbf{L}_{m+} \mathbf{D}_m \mathbf{L}_m^\top \quad (3.9)$$

mit

$$\mathbf{L}_{m+} = \begin{bmatrix} 1 & & & & \\ \lambda_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & \lambda_m & 1 & \\ & & & \lambda_{m+1} & \end{bmatrix} \in \mathbb{R}^{(m+1) \times m}$$

(weil \mathbf{T}_{m+} tridiagonal ist, ist \mathbf{L}_{m+} bidiagonal), einer Diagonalmatrix

$$\mathbf{D}_m = \text{diag}(\mu_1, \mu_2, \dots, \mu_m)$$

und $\mathbf{L}_m = [\mathbf{I}_m \mathbf{0}] \mathbf{L}_{m+} \in \mathbb{R}^{m \times m}$.

Lineare Gleichungssysteme

Das Verfahren der konjugierten Gradienten

Die Einträge dieser Matrizen berechnet man durch einen simplen Koeffizientenvergleich in der Matrixidentität $\mathbf{T}_{m+} = \mathbf{L}_{m+} \mathbf{D}_m \mathbf{L}_m^T$: die erste Spalte liefert

$$\gamma_1 = \mu_1 \quad \text{sowie} \quad \delta_2 = \mu_1 \lambda_2$$

und die darauffolgenden

$$\gamma_m = \mu_{m-1} \lambda_m^2 + \mu_m, \quad \delta_{m+1} = \mu_m \lambda_{m+1}, \quad m = 2, 3, \dots,$$

was wir wie folgt festhalten:

Lemma 3.11

Für die Einträge von \mathbf{L}_{m+} und \mathbf{D}_m gelten

$$\begin{aligned} \mu_1 &= \gamma_1, & \mu_{m+1} &= \gamma_{m+1} - \mu_m \lambda_{m+1}^2, & m &= 1, 2, \dots, \\ \lambda_{m+1} &= \delta_{m+1} / \mu_m, & & & m &= 1, 2, \dots \end{aligned}$$

Die Spalten von

$$\hat{\mathbf{P}}_m = [\hat{\mathbf{p}}_1 \ \hat{\mathbf{p}}_2 \ \cdots \ \hat{\mathbf{p}}_m] := \mathbf{V}_m \mathbf{L}_m^{-\top}$$

(wegen $\det \mathbf{L}_m = 1$ ist \mathbf{L}_m^{\top} invertierbar) spannen offenbar denselben Unterraum auf wie die Spalten von \mathbf{V}_m , d.h.

$$\text{span}\{\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \dots, \hat{\mathbf{p}}_m\} = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\} = \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0).$$

Aus

$$\mathbf{T}_m = [(\mathbf{A}\mathbf{v}_i, \mathbf{v}_j)]_{1 \leq i, j \leq m} \quad (\text{vgl. (2.7)})$$

und der Cholesky-Zerlegung (3.9) von \mathbf{T}_m folgt

$$\begin{aligned} [(\mathbf{A}\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j)]_{1 \leq i, j \leq m} &= \mathbf{L}_m^{-1} [(\mathbf{A}\mathbf{v}_i, \mathbf{v}_j)]_{1 \leq i, j \leq m} \mathbf{L}_m^{-\top} \\ &= \mathbf{L}_m^{-1} \mathbf{T}_m \mathbf{L}_m^{-\top} \\ &= \mathbf{L}_m^{-1} \mathbf{L}_m \mathbf{D}_m \mathbf{L}_m^{\top} \mathbf{L}_m^{-\top} = \mathbf{D}_m. \end{aligned}$$

Insbesondere sind die Vektoren $\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \dots, \hat{\mathbf{p}}_m$ \mathbf{A} -orthogonal (**A-konjugiert**):

$$(\mathbf{A}\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j) = 0 \quad \text{für alle } i \neq j.$$

Außerdem gelten $\hat{\mathbf{P}}_m \mathbf{L}_m^\top = \mathbf{V}_m$ und

$$\begin{aligned} \mathbf{A} \hat{\mathbf{P}}_m &= \mathbf{A} \mathbf{V}_m \mathbf{L}_m^{-\top} = \mathbf{V}_{m+1} \mathbf{T}_{m+1} \mathbf{L}_m^{-\top} = \mathbf{V}_{m+1} (\mathbf{L}_m + \mathbf{D}_m \mathbf{L}_m^\top) \mathbf{L}_m^{-\top} \\ &= \mathbf{V}_{m+1} \mathbf{L}_{m+1} \mathbf{D}_m, \quad m = 1, 2, \dots \end{aligned}$$

Schreibt man diese beiden Matrixgleichungen spaltenweise auf, so ergibt sich (siehe auch Lemma 3.11)

$$\hat{\mathbf{p}}_1 = \mathbf{v}_1,$$

$$\lambda_k \hat{\mathbf{p}}_{k-1} + \hat{\mathbf{p}}_k = (\delta_k / \mu_{k-1}) \hat{\mathbf{p}}_{k-1} + \hat{\mathbf{p}}_k = \mathbf{v}_k \quad (k = 2, 3, \dots),$$

$$\mathbf{A} \hat{\mathbf{p}}_k = \mu_k \mathbf{v}_k + \lambda_{k+1} \mu_k \mathbf{v}_{k+1} = \mu_k \mathbf{v}_k + \delta_{k+1} \mathbf{v}_{k+1} \quad (k = 1, 2, \dots).$$

Beachtet man noch, dass $(\mathbf{A} \hat{\mathbf{p}}_m, \hat{\mathbf{p}}_m) = \mu_m$ und

$$\delta_{m+1} = \|\delta_{m+1} \mathbf{v}_{m+1}\| = \|\mathbf{A} \hat{\mathbf{p}}_m - \mu_m \mathbf{v}_m\| \quad (\text{denn } \delta_{m+1} > 0)$$

gelten, so erhält man eine Variante des hermiteschen Lanczos-Verfahrens zur Konstruktion einer ON-Basis von $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$, die auf zwei **gekoppelten zweistufigen Rekursionen** (anstelle der Drei-Term-Rekursion (2.9)) beruht:

Algorithmus 7: Hermitescher Lanczos–Prozess mit gekoppelten zweistufigen Rekursionen

Gegeben: $\mathbf{r}_0 \in \mathbb{C}^N$, $\mathbf{A} \in \mathbb{C}^{N \times N}$ selbstadjungiert und positiv definit bez. Innenprodukt (\cdot, \cdot) auf \mathbb{C}^N , $\|\cdot\| = \sqrt{(\cdot, \cdot)}$.

- 1 $\delta_1 \leftarrow \|\mathbf{r}_0\|$, $\mathbf{v}_1 \leftarrow \mathbf{r}_0/\delta_1$, $\hat{\mathbf{p}}_0 \leftarrow \mathbf{0}$, $\mu_0 \leftarrow 1$
 - 2 **for** $m = 1, 2, \dots$ **do**
 - 3 $\hat{\mathbf{p}}_m \leftarrow \mathbf{v}_m - \delta_m/\mu_{m-1}\hat{\mathbf{p}}_{m-1}$
 - 4 $\mathbf{w} \leftarrow \mathbf{A}\hat{\mathbf{p}}_m$
 - 5 $\mu_m \leftarrow (\mathbf{w}, \hat{\mathbf{p}}_m)$
 - 6 $\mathbf{w} \leftarrow \mathbf{w} - \mu_m\mathbf{v}_m$
 - 7 $\delta_{m+1} \leftarrow \|\mathbf{w}\|$
 - 8 $\mathbf{v}_{m+1} \leftarrow \mathbf{w}/\delta_{m+1}$
-

- Nach Korollar 3.5 sind die Residualvektoren \mathbf{r}_m eines OR-Verfahrens und die Basisvektoren \mathbf{v}_{m+1} kollinear (was übrigens auch sofort daraus folgt, dass beide Vektoren im eindimensionalen Raum $\mathcal{K}_{m+1}(\mathbf{A}, \mathbf{r}_0) \cap \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)^\perp$ liegen).
- Für $m = 1, 2, \dots, d(\mathbf{A}, \mathbf{r}_0)$ gibt es also $\theta_m \neq 0$ mit

$$\mathbf{r}_{m-1}^{\text{CG}} = \theta_m \mathbf{v}_m \quad (\theta_1 = \|\mathbf{r}_0^{\text{CG}}\|).$$

Um Speicherplatz zu sparen, wollen wir die Vektoren \mathbf{v}_m durch (Vielfache von) $\mathbf{r}_{m-1}^{\text{CG}}$ ersetzen.

- Aus

$$\hat{\mathbf{p}}_1 = \mathbf{v}_1 \quad \text{und} \quad \hat{\mathbf{p}}_m = \mathbf{v}_m - \frac{\delta_m}{\mu_{m-1}} \hat{\mathbf{p}}_{m-1} \quad (m = 2, 3, \dots)$$

ergeben sich durch Multiplikation mit θ_m die Gleichungen

$$\theta_1 \hat{\mathbf{p}}_1 = \mathbf{r}_0 \quad \text{und} \quad \theta_m \hat{\mathbf{p}}_m = \mathbf{r}_{m-1}^{\text{CG}} - \frac{\delta_m}{\mu_{m-1}} \frac{\theta_m}{\theta_{m-1}} \theta_{m-1} \hat{\mathbf{p}}_{m-1}.$$

- Setzt man jetzt noch

$$\mathbf{p}_m := \theta_m \hat{\mathbf{p}}_m \quad (m = 1, 2, \dots),$$

so liest sich das als

$$\mathbf{p}_1 = \mathbf{r}_0, \quad \mathbf{p}_m = \mathbf{r}_{m-1}^{\text{CG}} + \beta_{m-1} \mathbf{p}_{m-1} \quad (m = 2, 3, \dots) \quad (3.10)$$

mit $\beta_{m-1} = -(\delta_m / \mu_{m-1})(\theta_m / \theta_{m-1})$, was man aber anders berechnet.

- Weil der Vektor \mathbf{p}_m ein skalares Vielfaches des Vektors $\hat{\mathbf{p}}_m$ ist, gilt

$$\text{span}\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m\} = \text{span}\{\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \dots, \hat{\mathbf{p}}_m\} = \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0).$$

- Wegen

$$(\mathbf{A}\mathbf{p}_i, \mathbf{p}_j) = \theta_i \bar{\theta}_j (\mathbf{A}\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j) = 0 \quad \text{für } i \neq j$$

stellt auch $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m\}$ eine \mathbf{A} -orthogonale Basis von $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ dar.

- Insbesondere können wir die CG-Iterierte $\mathbf{x}_m^{\text{CG}} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ in der Form

$$\mathbf{x}_m^{\text{CG}} = \mathbf{x}_0 + \mathbf{P}_m \mathbf{z}_m^{\text{CG}} \quad \text{mit } \mathbf{P}_m := [\mathbf{p}_1 \ \mathbf{p}_2 \ \cdots \ \mathbf{p}_m] \text{ und } \mathbf{z}_m^{\text{CG}} \in \mathbb{C}^m$$

schreiben.

- Aus $\mathbf{r}_m^{\text{CG}} \perp \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ folgt

$$\begin{aligned} 0 &= \left(\mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{P}_m \mathbf{z}_m^{\text{CG}}), \mathbf{p}_j \right) = (\mathbf{r}_0 - \mathbf{A} \mathbf{P}_m \mathbf{z}_m^{\text{CG}}, \mathbf{p}_j) \\ &= (\mathbf{r}_0, \mathbf{p}_j) - (\mathbf{A} \mathbf{P}_m \mathbf{z}_m^{\text{CG}}, \mathbf{p}_j) \quad (j = 1, 2, \dots, m). \end{aligned}$$

- Der Koordinatenvektor \mathbf{z}_m^{CG} löst also das lineare Gleichungssystem

$$[(\mathbf{A} \mathbf{p}_i, \mathbf{p}_j)]_{1 \leq i, j \leq m} \mathbf{z}_m^{\text{CG}} = [(\mathbf{r}_0, \mathbf{p}_j)]_{1 \leq j \leq m},$$

dessen Matrix wegen der \mathbf{A} -Orthogonalität der \mathbf{p}_j Diagonalform besitzt.

- Das bedeutet

$$\mathbf{z}_m^{\text{CG}} = \begin{bmatrix} \mathbf{z}_{m-1}^{\text{CG}} \\ \alpha_m \end{bmatrix} \quad \text{mit} \quad \alpha_m = \frac{(\mathbf{r}_0, \mathbf{p}_m)}{(\mathbf{A}\mathbf{p}_m, \mathbf{p}_m)}.$$

- Jetzt sieht man, dass sich die CG-Näherungen durch

$$\begin{aligned} \mathbf{x}_m^{\text{CG}} &= \mathbf{x}_0 + \mathbf{P}_m \mathbf{z}_m^{\text{CG}} = \mathbf{x}_0 + [\mathbf{P}_{m-1} \ \mathbf{p}_m] \begin{bmatrix} \mathbf{z}_{m-1}^{\text{CG}} \\ \alpha_m \end{bmatrix} \\ &= \mathbf{x}_0 + \mathbf{P}_{m-1} \mathbf{z}_{m-1}^{\text{CG}} + \alpha_m \mathbf{p}_m = \mathbf{x}_{m-1}^{\text{CG}} + \alpha_m \mathbf{p}_m \end{aligned} \quad (3.11)$$

aufdatieren lassen.

- Daraus folgt noch

$$\mathbf{r}_m^{\text{CG}} = \mathbf{b} - \mathbf{A}(\mathbf{x}_{m-1}^{\text{CG}} + \alpha_m \mathbf{p}_m) = \mathbf{r}_{m-1}^{\text{CG}} - \alpha_m \mathbf{A}\mathbf{p}_m. \quad (3.12)$$

- Zusammen mit (3.10) stellen diese beiden Rekursionsformeln i.W. das Verfahren der konjugierten Gradienten dar. Wir müssen nur noch klären, wie die Koeffizienten α_m und β_m effizient berechnet werden.

- Wegen

$$\mathbf{r}_{m-1}^{\text{CG}} = \mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{P}_{m-1}\mathbf{z}_{m-1}^{\text{CG}}) = \mathbf{r}_0 - \mathbf{A}\mathbf{P}_{m-1}\mathbf{z}_{m-1}^{\text{CG}}$$

und $\mathbf{p}_m \perp_A \mathcal{K}_{m-1}(\mathbf{A}, \mathbf{r}_0)$ gilt

$$(\mathbf{r}_{m-1}^{\text{CG}}, \mathbf{p}_m) = (\mathbf{r}_0 - \mathbf{A}\mathbf{P}_{m-1}\mathbf{z}_{m-1}^{\text{CG}}, \mathbf{p}_m) = (\mathbf{r}_0, \mathbf{p}_m).$$

- Mit Hilfe von (3.10) und $\mathbf{r}_{m-1}^{\text{CG}} \perp \mathcal{K}_{m-1}(\mathbf{A}, \mathbf{r}_0) \ni \mathbf{p}_{m-1}$ ergibt sich

$$(\mathbf{r}_{m-1}^{\text{CG}}, \mathbf{p}_m) = (\mathbf{r}_{m-1}^{\text{CG}}, \mathbf{r}_{m-1}^{\text{CG}} + \beta_{m-1}\mathbf{p}_{m-1}) = (\mathbf{r}_{m-1}^{\text{CG}}, \mathbf{r}_{m-1}^{\text{CG}}) \quad (3.13)$$

also

$$\alpha_m = \frac{(\mathbf{r}_0, \mathbf{p}_m)}{(\mathbf{A}\mathbf{p}_m, \mathbf{p}_m)} = \frac{(\mathbf{r}_{m-1}^{\text{CG}}, \mathbf{p}_m)}{(\mathbf{A}\mathbf{p}_m, \mathbf{p}_m)} = \frac{(\mathbf{r}_{m-1}^{\text{CG}}, \mathbf{r}_{m-1}^{\text{CG}})}{(\mathbf{A}\mathbf{p}_m, \mathbf{p}_m)} \quad (m = 1, 2, \dots),$$

was das Speichern des Vektors \mathbf{r}_0 überflüssig macht
(und, en passant, $\alpha_m > 0$ für $m = 1, 2, \dots, d(\mathbf{A}, \mathbf{r}_0)$ beweist).

- Die \mathbf{A} -Orthogonalität der Richtungsvektoren \mathbf{p}_m impliziert mit (3.10) außerdem

$$0 = (\mathbf{A}\mathbf{p}_{m+1}, \mathbf{p}_m) = (\mathbf{A}\mathbf{r}_m^{\text{CG}}, \mathbf{p}_m) + \beta_m(\mathbf{A}\mathbf{p}_m, \mathbf{p}_m),$$

also $\beta_m = -(\mathbf{A}\mathbf{r}_m^{\text{CG}}, \mathbf{p}_m) / (\mathbf{A}\mathbf{p}_m, \mathbf{p}_m)$.

- Nun ist mit (3.12)

$$(\mathbf{A}\mathbf{r}_m^{\text{CG}}, \mathbf{p}_m) = (\mathbf{r}_m^{\text{CG}}, \mathbf{A}\mathbf{p}_m) = (\mathbf{r}_m^{\text{CG}}, \frac{1}{\alpha_m}(\mathbf{r}_{m-1}^{\text{CG}} - \mathbf{r}_m^{\text{CG}})) = -\frac{1}{\alpha_m}(\mathbf{r}_m^{\text{CG}}, \mathbf{r}_m^{\text{CG}})$$

weil $\mathbf{r}_m^{\text{CG}} \perp \mathbf{r}_{m-1}^{\text{CG}}$.

- Mit derselben Argumentation ergibt sich

$$(\mathbf{A}\mathbf{p}_m, \mathbf{p}_m) = (\mathbf{p}_m, \mathbf{A}\mathbf{p}_m) = (\mathbf{p}_m, \frac{1}{\alpha_m}(\mathbf{r}_{m-1}^{\text{CG}} - \mathbf{r}_m^{\text{CG}})) = \frac{1}{\alpha_m}(\mathbf{p}_m, \mathbf{r}_{m-1}^{\text{CG}}).$$

- Setzt man jetzt noch (3.13) ein, so folgt

$$(\mathbf{A}\mathbf{p}_m, \mathbf{p}_m) = \frac{1}{\alpha_m}(\mathbf{p}_m, \mathbf{r}_{m-1}^{\text{CG}}) = \frac{1}{\alpha_m}(\mathbf{r}_{m-1}^{\text{CG}}, \mathbf{r}_{m-1}^{\text{CG}}).$$

Wir haben damit

$$\beta_m = \frac{(\mathbf{r}_m^{\text{CG}}, \mathbf{r}_m^{\text{CG}})}{(\mathbf{r}_{m-1}^{\text{CG}}, \mathbf{r}_{m-1}^{\text{CG}})}$$

bewiesen.

Algorithmus 8: Verfahren der konjugierten Gradienten

Gegeben: $\mathbf{A} \in \mathbb{C}^{N \times N}$ selbstadjungiert und positiv definit bez.

Innenprodukt (\cdot, \cdot) auf \mathbb{C}^N , $\|\cdot\| = \sqrt{(\cdot, \cdot)}$, $\mathbf{b} \in \mathbb{C}^N$, $\mathbf{x}_0 \approx \mathbf{A}^{-1}\mathbf{b}$

- 1 $\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{p}_1 \leftarrow \mathbf{r}_0$
- 2 **for** $m = 1, 2, \dots$ **do**
- 3 $\mathbf{w} \leftarrow \mathbf{A}\mathbf{p}_m$
- 4 $\alpha_m \leftarrow (\mathbf{r}_{m-1}, \mathbf{r}_{m-1}) / (\mathbf{w}, \mathbf{p}_m)$
- 5 $\mathbf{x}_m \leftarrow \mathbf{x}_{m-1} + \alpha_m \mathbf{p}_m$
- 6 $\mathbf{r}_m \leftarrow \mathbf{r}_{m-1} - \alpha_m \mathbf{w}$
- 7 $\beta_m \leftarrow (\mathbf{r}_m, \mathbf{r}_m) / (\mathbf{r}_{m-1}, \mathbf{r}_{m-1})$
- 8 $\mathbf{p}_{m+1} \leftarrow \mathbf{r}_m + \beta_m \mathbf{p}_m$

Pro Schritt: Berechnung von

- einem Matrix-Vektor-Produkt $(\mathbf{A}\mathbf{p}_m)$,
- zwei Innenprodukten $((\mathbf{w}, \mathbf{p}_m), (\mathbf{r}_m, \mathbf{r}_m))$, denn $(\mathbf{r}_{m-1}, \mathbf{r}_{m-1})$ bereits aus vorherigen Schritt bekannt) sowie
- drei Vektor-Aufdatierungen (für \mathbf{x}_m , \mathbf{r}_m sowie \mathbf{p}_{m+1}).

Es reicht aus, vier Vektoren $(\mathbf{w}, \mathbf{x}_m, \mathbf{r}_m, \mathbf{p}_{m+1})$ zu speichern.

Satz 3.12

Es seien $\mathbf{x}_0, \mathbf{x}_1^{\text{CG}}, \dots, \mathbf{x}_d^{\text{CG}} = \mathbf{x}_*$ die Iterierten, $\mathbf{r}_0, \mathbf{r}_1^{\text{CG}}, \dots, \mathbf{r}_d^{\text{CG}} = \mathbf{0}$ die zugehörigen Residualvektoren und $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_d$ die Suchrichtungen des CG-Verfahrens zur Lösung von $\mathbf{A}\mathbf{x} = \mathbf{b}$. Dann gelten für alle $m = 1, 2, \dots, d$ die Identitäten

$$\text{span} \{ \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m \} = \text{span} \{ \mathbf{r}_0, \mathbf{r}_1^{\text{CG}}, \dots, \mathbf{r}_{m-1}^{\text{CG}} \} = \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0),$$

die Orthogonalitätsrelationen

$$\begin{aligned} \mathbf{p}_m &\perp_{\mathbf{A}} \mathbf{p}_k \quad (1 \leq k, m \leq d, m \neq k) \\ \text{und} \quad \mathbf{r}_m^{\text{CG}} &\perp \mathbf{r}_k^{\text{CG}} \quad (0 \leq k, m \leq d, m \neq k) \end{aligned}$$

sowie die Minimierungseigenschaften

$$\begin{aligned} \|\mathbf{x}_* - \mathbf{x}_m^{\text{CG}}\|_{\mathbf{A}} &= \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)} \|\mathbf{x}_* - \mathbf{x}\|_{\mathbf{A}} \\ \text{und} \quad \|\mathbf{b} - \mathbf{A}\mathbf{x}_m^{\text{CG}}\|_{\mathbf{A}^{-1}} &= \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_{\mathbf{A}^{-1}}. \end{aligned}$$

Lineare Gleichungssysteme

Das Verfahren der konjugierten Gradienten

- Bisher: $\mathbf{A} \in \mathbb{C}^{N \times N}$ invertierbar.
- Wir zeigen nun: das CG-Verfahren löst auch Systeme, deren Matrix selbstadjungiert und (lediglich) positiv semidefinit, d.h. $(\mathbf{A}\mathbf{x}, \mathbf{x}) \geq 0$ für alle \mathbf{x} , also möglicherweise singulär ist.
- Sei hierzu $\mathbf{A}\mathbf{x} = \mathbf{b}$ ein **konsistentes** LGS (d.h. $\mathbf{b} \in \mathcal{R}(\mathbf{A})$) mit \mathbf{A} positiv semidefinit und $\mathcal{N}(\mathbf{A}) \neq \{\mathbf{0}\}$.
- Die Lösungsmenge von $\mathbf{A}\mathbf{x} = \mathbf{b}$ ist durch $\mathbf{A}^\dagger \mathbf{b} + \mathcal{N}(\mathbf{A})$ gegeben. Dabei bezeichnet \mathbf{A}^\dagger die (Moore–Penrose) Pseudoinverse⁶ von \mathbf{A} .
- Die spezielle Lösung $\mathbf{A}^\dagger \mathbf{b}$ ist einzige Lösung in $\mathcal{R}(\mathbf{A})$. Unter allen Lösungen besitzt sie minimale Norm.

⁶ \mathbf{A} kann folgendermaßen zerlegt werden: $\mathbf{A} = \mathbf{U} \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{B} \end{bmatrix} \mathbf{U}^*$ mit $\mathbf{U}^* \mathbf{U} = \mathbf{I}_n$ und \mathbf{B} selbstadjungiert sowie positiv definit. Dann ist

$$\mathbf{A}^\dagger = \mathbf{U}^* \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{B}^{-1} \end{bmatrix} \mathbf{U}.$$

Diese Konstruktion von \mathbf{A}^\dagger ist nur in Spezialfällen korrekt. Einzelheiten findet man bei Campbell und Meyer (1991).

- Unter den gegebenen Voraussetzungen ist

$$\mathbb{C}^N = \mathcal{N}(\mathbf{A}) \oplus \mathcal{R}(\mathbf{A}),$$

wobei die Summanden zusätzlich orthogonal sind.

- Schränkt man \mathbf{A} auf $\mathcal{R}(\mathbf{A})$ ein, so resultiert eine invertierbare lineare Abbildung

$$\mathbf{A}^R := \mathbf{A}|_{\mathcal{R}(\mathbf{A})} : \mathcal{R}(\mathbf{A}) \rightarrow \mathcal{R}(\mathbf{A}), \quad \mathbf{x} \mapsto \mathbf{A}\mathbf{x},$$

denn \mathbf{A}^R ist per Definition surjektiv und, wegen $\mathcal{N}(\mathbf{A}) \cap \mathcal{R}(\mathbf{A}) = \{\mathbf{0}\}$, auch injektiv.

- Die Inverse von \mathbf{A}^R ist $\mathbf{A}^\dagger|_{\mathcal{R}(\mathbf{A})}$ und die eindeutige Lösung von $\mathbf{A}^R\mathbf{x} = \mathbf{b}$ ist $\mathbf{A}^\dagger\mathbf{b}$.

- Wir wollen das CG-Verfahren mit (beliebigem) Startvektor \mathbf{x}_0 untersuchen.
- Zunächst zerlegen wir \mathbf{x}_0 gemäß

$$\mathbf{x}_0 = \mathbf{x}_0^{\text{N}} + \mathbf{x}_0^{\text{R}} \quad \text{mit } \mathbf{x}_0^{\text{N}} = P_{\mathcal{N}(\mathbf{A})}\mathbf{x}_0 \in \mathcal{N}(\mathbf{A})$$
$$\text{und } \mathbf{x}_0^{\text{R}} = P_{\mathcal{R}(\mathbf{A})}\mathbf{x}_0 \in \mathcal{R}(\mathbf{A}),$$

wobei $P_{\mathcal{N}(\mathbf{A})}$ und $P_{\mathcal{R}(\mathbf{A})}$ die Orthogonalprojektionen auf $\mathcal{N}(\mathbf{A})$ bzw. $\mathcal{R}(\mathbf{A})$ bezeichnen.

- Dann wenden das CG-Verfahren

zum einen mit dem Startvektor \mathbf{x}_0 auf $\mathbf{A}\mathbf{x} = \mathbf{b}$,

zum anderen mit dem Startvektor \mathbf{x}_0^{R} auf $\mathbf{A}^{\text{R}}\mathbf{x} = \mathbf{b}$

an. Man erkennt, dass Richtungs- und Residualvektoren beider Verfahren identisch sind (vgl. Alg. 8), während sich die Näherungen um \mathbf{x}_0^{N} unterscheiden.

Korollar 3.13

Sei $\mathbf{A} \in \mathbb{C}^{N \times N}$ selbstadjungiert und positiv semidefinit und $\mathbf{b} \in \mathcal{R}(\mathbf{A})$. Außerdem seien \mathbf{x}_0 ein beliebiger Startvektor und \mathbf{x}_*^0 die durch

$$\mathbf{x}_*^0 := \mathbf{A}^\dagger \mathbf{b} + P_{\mathcal{N}(\mathbf{A})} \mathbf{x}_0$$

definierte spezielle Lösung von $\mathbf{A}\mathbf{x} = \mathbf{b}$. Dann besitzen die Näherungen \mathbf{x}_m^{CG} des CG-Verfahrens die Minimierungseigenschaften^a

$$\begin{aligned} \|\mathbf{x}_*^0 - \mathbf{x}_m^{\text{CG}}\|_{\mathbf{A}} &= \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, r_0)} \|\mathbf{x}_*^0 - \mathbf{x}\|_{\mathbf{A}} \\ \text{und} \quad \|\mathbf{b} - \mathbf{A}\mathbf{x}_m^{\text{CG}}\|_{\mathbf{A}^\dagger} &= \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, r_0)} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_{\mathbf{A}^\dagger}. \end{aligned}$$

Insbesondere bricht das CG-Verfahren nach endlich vielen Schritten mit der Näherung $\mathbf{x}_d = \mathbf{x}_*^0$ ab. Für $\mathbf{x}_0 \in \mathcal{R}(\mathbf{A})$, z.B. für $\mathbf{x}_0 = \mathbf{0}$, ergibt sich $\mathbf{x}_*^0 = \mathbf{A}^\dagger \mathbf{b}$.

^aDabei ist zu beachten, dass $\|\cdot\|_{\mathbf{A}}$ und $\|\cdot\|_{\mathbf{A}^\dagger}$ nur Halbnormen auf \mathbb{C}^N sind; schränkt man sie auf $\mathcal{R}(\mathbf{A})$ ein, so stellen sie aber Normen dar.

- 1 Einleitung
- 2 Krylov-Unterraumverfahren
- 3 Lineare Gleichungssysteme
 - 3.1 Lösungsstrategien
 - 3.2 Selbstadjungierte Probleme
 - 3.3 Das Verfahren der konjugierten Gradienten
 - 3.4 Krylov-Verfahren für lineare Ausgleichsprobleme
 - 3.5 Vorkonditionierung
- 4 Matrixfunktionen
- 5 Krylov-Verfahren für Matrixfunktionen
- 6 Das QR-Verfahren für Eigenwertaufgaben

Abschließend sollen Iterationsverfahren besprochen werden, mit denen **lineare Ausgleichsprobleme**

$$\| \mathbf{b} - \mathbf{A} \mathbf{x} \| \longrightarrow \min_{\mathbf{x} \in \mathbb{C}^N}, \quad \text{mit } \mathbf{A} \in \mathbb{C}^{M \times N}, \mathbf{b} \in \mathbb{C}^M \quad (3.14)$$

gelöst werden können.

Dabei bezeichnet

- $\| \cdot \|$ eine Norm, die durch ein Innenprodukt $(\cdot, \cdot)_1$ in \mathbb{C}^M induziert wird.
- Auch \mathbb{C}^N ist mit einem Innenprodukt $(\cdot, \cdot)_2$ versehen (die davon induzierte Norm wird auch mit $\| \cdot \|$ bezeichnet).

Lineare Gleichungssysteme

Die Normalgleichungen für lineare Ausgleichsprobleme

- Bekannt: $\mathbf{x} \in \mathbb{C}^N$ löst (3.14) genau dann, wenn \mathbf{x} die Normalgleichungen

$$\mathbf{A}^* \mathbf{A} \mathbf{x} = \mathbf{A}^* \mathbf{b} \quad (3.15)$$

erfüllt.

- $\mathbf{A}^* \in \mathbb{C}^{N \times M}$ ist die **Adjungierte** von \mathbf{A} , d.h.

$$(\mathbf{A} \mathbf{x}, \mathbf{r})_1 = (\mathbf{x}, \mathbf{A}^* \mathbf{r})_2 \quad \text{für alle } \mathbf{x} \in \mathbb{C}^N \text{ und alle } \mathbf{r} \in \mathbb{C}^M.$$

- Die Koeffizientenmatrix $\mathbf{A}^* \mathbf{A}$ von (3.15) ist selbstadjungiert und positiv semidefinit; außerdem sind die Normalgleichungen wegen $\mathcal{R}(\mathbf{A}^* \mathbf{A}) = \mathcal{R}(\mathbf{A}^*)$ immer konsistent.
- Es ist daher naheliegend, (3.15) mit dem CG-Verfahren zu lösen.

Lineare Gleichungssysteme

Anwendung des CG-Verfahrens auf die Normalgleichungen

Ersetzt man in Algorithmus 8 A durch A^*A und b durch A^*b , so erhält man

$$s_0 \leftarrow A^*b - A^*Ax_0, p_1 \leftarrow s_0$$

for $m = 1, 2, \dots$

$$w \leftarrow A^*Ap_m$$

$$\alpha_m \leftarrow (s_{m-1}, s_{m-1})_2 / (w, p_m)_2$$

$$x_m \leftarrow x_{m-1} + \alpha_m p_m$$

$$s_m \leftarrow s_{m-1} - \alpha_m w$$

$$\beta_m \leftarrow (s_m, s_m)_2 / (s_{m-1}, s_{m-1})_2$$

$$p_{m+1} = s_m + \beta_m p_m$$

- Wir führen hier den Vektor $s_m = A^*b - A^*Ax_m$ ein, weil wir die Bezeichnung $r_m = b - Ax_m$ für das Residuum von x_m bez. des Ausgangssystems $Ax = b$ reservieren wollen.
- Offenbar gilt $s_m = A^*r_m$ und, mit $z_m := Ap_m$, ist $(w, p_m)_2 = (A^*Ap_m, p_m)_2 = (Ap_m, Ap_m)_1 = (z_m, z_m)_1$ sowie $r_m = b - A(x_{m-1} - \alpha_m p_m) = r_{m-1} - \alpha_m z_m$.

Lineare Gleichungssysteme

Anwendung des CG-Verfahrens auf die Normalgleichungen

Nach diesen (eher kosmetischen) Eingriffen erhalten wir das sogenannte **CGNR-Verfahren** (NR steht für „normal residuals“).

[Paige und Saunders, 1982] sowie [Björck, 1996] nennen es **CGLS**.

Algorithmus 9: CGNR-Verfahren (auch CGLS)

Gegeben: Innenprodukte $(\cdot, \cdot)_1$ und $(\cdot, \cdot)_2$ in \mathbb{C}^M bzw. \mathbb{C}^N , $\mathbf{A} \in \mathbb{C}^{M \times N}$, $\mathbf{b} \in \mathbb{C}^M$ und eine Startnäherung $\mathbf{x}_0 \in \mathbb{C}^N$ für eine Kleinste-Quadrate-Lösung von $\|\mathbf{b} - \mathbf{A}\mathbf{x}\| \rightarrow \min$.

```
1  $\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$ ,  $\mathbf{s}_0 \leftarrow \mathbf{A}^* \mathbf{r}_0$ ,  $\mathbf{p}_1 \leftarrow \mathbf{s}_0$ 
2 for  $m = 1, 2, \dots$  do
3    $\mathbf{z} \leftarrow \mathbf{A}\mathbf{p}_m$ 
4    $\alpha_m \leftarrow (\mathbf{s}_{m-1}, \mathbf{s}_{m-1})_2 / (\mathbf{z}, \mathbf{z})_1$ 
5    $\mathbf{x}_m \leftarrow \mathbf{x}_{m-1} + \alpha_m \mathbf{p}_m$ 
6    $\mathbf{r}_m \leftarrow \mathbf{r}_{m-1} - \alpha_m \mathbf{z}$ 
7    $\mathbf{s}_m \leftarrow \mathbf{A}^* \mathbf{r}_m$ 
8    $\beta_m \leftarrow (\mathbf{s}_m, \mathbf{s}_m)_2 / (\mathbf{s}_{m-1}, \mathbf{s}_{m-1})_2$ 
9    $\mathbf{p}_{m+1} \leftarrow \mathbf{s}_m + \beta_m \mathbf{p}_m$ 
```

Lineare Gleichungssysteme

Anwendung des CG-Verfahrens auf die Normalgleichungen

Aufwand des CGNR-Verfahrens:

- 2 Matrix-Vektorprodukte (mit \mathbf{A} bzw. \mathbf{A}^*)
- 2 Innenprodukte
- 3 Vektoraufdatierungen
- Speicherung: $\mathbf{x}_m, \mathbf{r}_m, \mathbf{s}_m, \mathbf{p}_m, \mathbf{z}$

Korollar 3.14

Die CGNR-Iterierte $\mathbf{x}_m^{\text{CGNR}} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{r}_0)$ ist durch

$$\|\mathbf{b} - \mathbf{A} \mathbf{x}_m^{\text{CGNR}}\| = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{r}_0)} \|\mathbf{b} - \mathbf{A} \mathbf{x}\|$$

charakterisiert. Das CGNR-Verfahren bricht nach endlich vielen Schritten mit der Kleinsten-Quadrate-Lösung

$$\mathbf{x}_*^0 = P_{\mathcal{N}(\mathbf{A})} \mathbf{x}_0 + \mathbf{A}^\dagger \mathbf{b}$$

ab. Für $\mathbf{x}_0 \in \mathcal{R}(\mathbf{A}^*)$, z.B. für $\mathbf{x}_0 = \mathbf{0}$ ergibt sich $\mathbf{x}_*^0 = \mathbf{A}^\dagger \mathbf{b}$, die Kleinsten-Quadrate-Lösung mit minimaler Norm.

Das **LSQR-Verfahren** (Least Squares QR) von [Paige und Saunders, 1982] liefert in exakter Arithmetik dieselben Näherungen wie CGNR, ist aber numerische stabiler, wenn \mathbf{A} schlecht konditioniert ist. Wir formulieren zunächst das Ausgleichsproblem um.

Lemma 3.15

Jede Lösung $\mathbf{x} \in \mathbb{C}^N$ des Ausgleichsproblems (3.14) bzw. der Normalgleichungen (3.15) mit Residualvektor $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} \in \mathbb{R}^M$ erfüllt auch das **augmentierte System**

$$\begin{bmatrix} \mathbf{I}_M & \mathbf{A} \\ \mathbf{A}^* & \mathbf{O}_N \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}. \quad (3.16)$$

Genügt umgekehrt der Vektor $\begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix}$ ($\mathbf{r} \in \mathbb{C}^M$, $\mathbf{x} \in \mathbb{C}^N$) den Gleichungen (3.16), so löst \mathbf{x} das Ausgleichsproblem (3.14) bzw. die Normalgleichungen (3.15) und \mathbf{r} ist der zugehörige Residualvektor.

Die Matrix

$$\widehat{\mathbf{A}} := \begin{bmatrix} \mathbf{I}_M & \mathbf{A} \\ \mathbf{A}^* & \mathbf{O}_N \end{bmatrix} \in \mathbb{C}^{(M+N) \times (M+N)}$$

aus (3.16) ist selbstadjungiert bezüglich des Innenprodukts

$$\left(\begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix}, \begin{bmatrix} \mathbf{s} \\ \mathbf{y} \end{bmatrix} \right) := (\mathbf{r}, \mathbf{s})_1 + (\mathbf{x}, \mathbf{y})_2 \quad (\mathbf{r}, \mathbf{s} \in \mathbb{C}^M, \mathbf{x}, \mathbf{y} \in \mathbb{C}^N) \quad (3.17)$$

auf \mathbb{C}^{M+N} .

Sei jetzt \mathbf{x}_0 ein beliebiger Startvektor mit Residuum $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$. Wir setzen

$$\widehat{\mathbf{r}}_0 = \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{M+N}.$$

Lemma 3.16

$$\mathcal{K}_{2m}(\widehat{\mathbf{A}}, \widehat{\mathbf{r}}_0) = \left\{ \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} : \mathbf{u} \in \mathcal{K}_m(\mathbf{A}\mathbf{A}^*, \mathbf{r}_0), \mathbf{v} \in \mathcal{K}_m(\mathbf{A}^*\mathbf{A}, \mathbf{A}^*\mathbf{r}_0) \right\}.$$

Wir wenden den Hermiteschen Lanczos-Prozess (siehe Alg. 4) auf \hat{A} und \hat{r}_0 an:

$$\hat{w}_1 \leftarrow \hat{r}_0, \delta_1 \leftarrow \|\hat{w}_1\|, \hat{v}_1 \leftarrow \hat{w}_1/\delta_1, \hat{v}_0 \leftarrow \mathbf{0}$$

for $m = 1, 2, \dots$

$$\hat{w}_{m+1} \leftarrow \hat{A}\hat{v}_m$$

$$\gamma_m \leftarrow (\hat{w}_{m+1}, \hat{v}_m)$$

$$\hat{w}_{m+1} \leftarrow \hat{w}_{m+1} - \gamma_m \hat{v}_m - \delta_m \hat{v}_{m-1}$$

$$\delta_{m+1} \leftarrow \|\hat{w}_{m+1}\|$$

$$\hat{v}_{m+1} \leftarrow \hat{w}_{m+1}/\delta_{m+1}.$$

Im nächsten Schritt beachten wir die spezielle Struktur von \hat{A} und \hat{r}_0 und schreiben den Lanczos-Prozess blockweise mit

$$\hat{w}_m = \begin{bmatrix} \mathbf{w}_m \\ \mathbf{z}_m \end{bmatrix} \quad \text{und} \quad \hat{v}_m = \begin{bmatrix} \mathbf{u}_m \\ \mathbf{v}_m \end{bmatrix} :$$

$$\mathbf{w}_1 \leftarrow \mathbf{r}_0, \mathbf{z}_1 \leftarrow \mathbf{0}, \delta_1 \leftarrow \|\mathbf{w}_1\|$$

$$\mathbf{u}_1 \leftarrow \mathbf{w}_1/\delta_1, \mathbf{v}_1 \leftarrow \mathbf{z}_1/\delta_1, \mathbf{u}_0 \leftarrow \mathbf{0}, \mathbf{v}_0 \leftarrow \mathbf{0}$$

for $m = 1, 2, \dots$

$$\mathbf{w}_{m+1} \leftarrow \mathbf{u}_m + \mathbf{A}\mathbf{v}_m, \mathbf{z}_{m+1} \leftarrow \mathbf{A}^* \mathbf{u}_m$$

$$\gamma_m \leftarrow (\mathbf{w}_{m+1}, \mathbf{u}_m)_1 + (\mathbf{z}_{m+1}, \mathbf{v}_m)_2$$

$$\mathbf{w}_{m+1} \leftarrow \mathbf{w}_{m+1} - \gamma_m \mathbf{u}_m - \delta_m \mathbf{u}_{m-1}$$

$$\mathbf{z}_{m+1} \leftarrow \mathbf{z}_{m+1} - \gamma_m \mathbf{v}_m - \delta_m \mathbf{v}_{m-1}$$

$$\delta_{m+1} \leftarrow [\|\mathbf{w}_{m+1}\|^2 + \|\mathbf{z}_{m+1}\|^2]^{1/2}$$

$$\mathbf{u}_{m+1} \leftarrow \mathbf{w}_{m+1}/\delta_{m+1}, \mathbf{v}_{m+1} \leftarrow \mathbf{z}_{m+1}/\delta_{m+1}$$

Jetzt stellt sich heraus, dass einige dieser Größen nicht (oder zumindest nicht so kompliziert) berechnet werden müssen:

Lemma 3.17

Für $m = 1, 2, \dots$ gelten

$$\begin{aligned}z_2 &= A^* u_1, \\z_{2m} &= A^* u_{2m-1} - \delta_{2m-1} v_{2m-2}, \\ \gamma_{2m-1} &= 1, \quad \gamma_{2m} = 0, \\ \mathbf{w}_{2m} &= \mathbf{u}_{2m} = \mathbf{0}, \quad \delta_{2m} = \|z_{2m}\|, \\ \mathbf{w}_{2m+1} &= A v_{2m} - \delta_{2m} u_{2m-1}, \\ z_{2m+1} &= v_{2m+1} = \mathbf{0}, \quad \delta_{2m+1} = \|\mathbf{w}_{2m+1}\|.\end{aligned}$$

Mit den Bezeichnungen

$$\begin{aligned} \mathbf{w}_m^{\text{GK}} &:= \mathbf{w}_{2m-1}, & \mathbf{u}_m^{\text{GK}} &:= \mathbf{u}_{2m-1}, \\ \mathbf{z}_m^{\text{GK}} &:= \mathbf{z}_{2m}, & \mathbf{v}_m^{\text{GK}} &:= \mathbf{v}_{2m}, \\ \alpha_m^{\text{GK}} &:= \delta_{2m} = \|\mathbf{z}_m^{\text{GK}}\|, & \beta_m^{\text{GK}} &:= \delta_{2m-1} = \|\mathbf{w}_m^{\text{GK}}\| \end{aligned}$$

erhält man aus dem Lanczos-Prozess einen Algorithmus, der \mathbf{A} und \mathbf{A}^* simultan bidiagonalisiert⁷.

⁷vgl. G. Golub und W. Kahan: Calculating the singular values and pseudo-inverse of a matrix. *SIAM J. Numer. Anal.* **2**, 205–224 (1965)

Algorithmus 10: Golub–Kahan Bidiagonalisierung

Gegeben: Innenprodukt-induzierte Normen $\|\cdot\|$ in \mathbb{C}^M bzw. \mathbb{C}^N ,
 $A \in \mathbb{C}^{M \times N}$ und $r_0 \in \mathbb{C}^M$.

- 1 $w_1^{\text{GK}} \leftarrow r_0, \beta_1^{\text{GK}} \leftarrow \|w_1^{\text{GK}}\|, u_1^{\text{GK}} \leftarrow w_1^{\text{GK}}/\beta_1^{\text{GK}}$
 - 2 $z_1^{\text{GK}} \leftarrow A^* u_1^{\text{GK}}, \alpha_1^{\text{GK}} \leftarrow \|z_1^{\text{GK}}\|, v_1^{\text{GK}} \leftarrow z_1^{\text{GK}}/\alpha_1^{\text{GK}}$.
 - 3 **for** $m = 1, 2, \dots$ **do**
 - 4 $w_{m+1}^{\text{GK}} \leftarrow A v_m^{\text{GK}} - \alpha_m^{\text{GK}} u_m^{\text{GK}}$
 - 5 $\beta_{m+1}^{\text{GK}} \leftarrow \|w_{m+1}^{\text{GK}}\|$
 - 6 $u_{m+1}^{\text{GK}} \leftarrow w_{m+1}^{\text{GK}}/\beta_{m+1}^{\text{GK}}$
 - 7 $z_{m+1}^{\text{GK}} \leftarrow A^* u_{m+1}^{\text{GK}} - \beta_{m+1}^{\text{GK}} v_m^{\text{GK}}$
 - 8 $\alpha_{m+1}^{\text{GK}} \leftarrow \|z_{m+1}^{\text{GK}}\|$
 - 9 $v_{m+1}^{\text{GK}} \leftarrow z_{m+1}^{\text{GK}}/\alpha_{m+1}^{\text{GK}}$
-

Die entscheidenden Gleichungen sind hier

$$\begin{aligned} \mathbf{A} \mathbf{v}_m &= \beta_{m+1} \mathbf{u}_{m+1} + \alpha_m \mathbf{u}_m, \\ \mathbf{A}^* \mathbf{u}_{m+1} &= \alpha_{m+1} \mathbf{v}_{m+1} + \beta_{m+1} \mathbf{v}_m, \end{aligned}$$

(ab sofort wird der hochgestellte Zusatz „GK“ weggelassen) oder, mit

$$\mathbf{U}_m := [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_m], \quad \mathbf{V}_m := [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_m]$$

(es gelten $(\mathbf{u}_k, \mathbf{u}_\ell)_1 = \delta_{k,\ell}$ und $(\mathbf{v}_k, \mathbf{v}_\ell)_2 = \delta_{k,\ell}$) und den reellen Matrizen

$$\mathbf{B}_m = \begin{bmatrix} \alpha_1 & 0 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \beta_m & \alpha_m \end{bmatrix}, \quad \mathbf{B}_{m+} = \begin{bmatrix} \mathbf{B}_m \\ \beta_{m+1} \mathbf{e}_m^\top \end{bmatrix},$$

$$\begin{aligned} \mathbf{A} \mathbf{V}_m &= \mathbf{U}_m \mathbf{B}_m + \beta_{m+1} \mathbf{u}_{m+1} \mathbf{e}_m^\top = \mathbf{U}_{m+1} \mathbf{B}_{m+}, & \mathbf{r}_0 &= \beta_1 \mathbf{U}_{m+1} \mathbf{e}_1, \\ \mathbf{A}^* \mathbf{U}_{m+1} &= \mathbf{V}_m \mathbf{B}_{m+}^\top + \alpha_{m+1} \mathbf{v}_{m+1} \mathbf{e}_{m+1}^\top = \mathbf{V}_{m+1} \mathbf{B}_{m+1}^\top, \end{aligned}$$

Lineare Gleichungssysteme

Das LSQR-Verfahren für lineare Ausgleichsprobleme

Wir kehren zu unserem ursprünglichen Problem zurück, nämlich der Berechnung von $\mathbf{x}_m^{\text{LSQR}} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{r}_0)$ mit

$$\|\mathbf{r}_m^{\text{LSQR}}\| = \|\mathbf{b} - \mathbf{A} \mathbf{x}_m^{\text{LSQR}}\|_2 = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{r}_0)} \|\mathbf{b} - \mathbf{A} \mathbf{x}\|_2.$$

Weil die Spalten von \mathbf{V}_m eine Basis des Krylov-Raums $\mathcal{K}_m(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{r}_0)$ bilden, ist $\mathbf{x}_m^{\text{LSQR}} = \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m^{\text{LSQR}}$ (mit einem Koordinatenvektor $\mathbf{y}_m^{\text{LSQR}} \in \mathbb{C}^m$) und daher, da die Spalten von \mathbf{U}_{m+1} orthonormal sind,

$$\begin{aligned} \|\mathbf{r}_m^{\text{LSQR}}\| &= \|\mathbf{b} - \mathbf{A} [\mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m^{\text{LSQR}}]\| \\ &= \|\mathbf{r}_0 - \mathbf{A} \mathbf{V}_m \mathbf{y}_m^{\text{LSQR}}\| \\ &= \|\beta_1 \mathbf{U}_{m+1} \mathbf{e}_1 - \mathbf{U}_{m+1} \mathbf{B}_{m+} \mathbf{y}_m^{\text{LSQR}}\| \\ &= \|\beta_1 \mathbf{e}_1 - \mathbf{B}_{m+} \mathbf{y}_m^{\text{LSQR}}\|_2. \end{aligned} \tag{3.18}$$

Wieder löst der Koordinatenvektor $\mathbf{y}_m^{\text{LSQR}}$ ein klassisches lineares Ausgleichsproblem, jetzt mit einer bidiagonalen Matrix \mathbf{B}_{m+} .

Lineare Gleichungssysteme

Das LSQR-Verfahren für lineare Ausgleichsprobleme

Auch dieses Mal bestimmen wir eine QR-Zerlegung von B_{m+} mit Hilfe von Givens-Rotationen:

$$Q_m B_{m+} = \begin{bmatrix} R_m \\ \mathbf{0} \end{bmatrix} \text{ mit } R_m = \begin{bmatrix} r_{1,1} & r_{1,2} & 0 & 0 & \cdots & 0 \\ 0 & r_{2,2} & r_{2,3} & \ddots & \ddots & \vdots \\ \vdots & \ddots & r_{3,3} & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & r_{m-1,m} \\ 0 & \cdots & \cdots & \cdots & 0 & r_{m,m} \end{bmatrix},$$

mit der orthogonalen Matrix

$$Q_m = G_m \begin{bmatrix} G_{m-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} G_{m-2} & \mathbf{0} \\ \mathbf{0} & I_2 \end{bmatrix} \cdots \begin{bmatrix} G_1 & \mathbf{0} \\ \mathbf{0} & I_{m-1} \end{bmatrix} \in \mathbb{R}^{(m+1) \times (m+1)},$$

und den Rotationsmatrizen

$$G_j = \begin{bmatrix} I_{j-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & c_j & s_j \\ \mathbf{0} & -s_j & c_j \end{bmatrix} \in \mathbb{R}^{(j+1) \times (j+1)}, \quad s_j^2 + c_j^2 = 1.$$

Lineare Gleichungssysteme

Das LSQR-Verfahren für lineare Ausgleichsprobleme

Weil wir bereits zweimal erläutert haben (bei der Herleitung von GMRES und MINRES), wie die Größen $r_{m,m}$, $r_{m-1,m}$, c_m und s_m bestimmt werden, fassen wir uns jetzt kurz:

$$r_{m-1,m} = s_{m-1}\alpha_m, \quad m = 2, 3, \dots,$$

$$r_{m,m} = [\tau_m^2 + \beta_{m+1}^2]^{1/2}, \quad m = 1, 2, \dots,$$

$$\text{mit } \tau_m = \begin{cases} \alpha_1, & m = 1, \\ c_{m-1}\alpha_m, & m = 2, 3, \dots, \end{cases}$$

$$c_m = \tau_m/r_{m,m}, \quad s_m = \beta_{m+1}/r_{m,m}.$$

Mit Hilfe dieser QR-Zerlegung erhalten wir aus (3.18)

$$\|\mathbf{r}_m^{\text{LSQR}}\| = \left\| \beta_1 \mathbf{Q}_m \mathbf{e}_1 - \begin{bmatrix} \mathbf{R}_m \\ \mathbf{0} \end{bmatrix} \mathbf{y}_m^{\text{LSQR}} \right\|$$

und damit

$$\mathbf{y}_m^{\text{LSQR}} = \beta_1 \mathbf{R}_m^{-1} \mathbf{q}_m$$

sowie

$$\|\mathbf{r}_m^{\text{LSQR}}\| = \beta_1 |\eta_m|,$$

wenn wir die erste Spalte von \mathbf{Q}_m wieder in der Form $[\mathbf{q}_m^T \ \eta_m]^T$ schreiben. Es gelten (vgl. Lemma 3.7)

$$\mathbf{q}_1 = c_1, \ \eta_1 = -s_1, \ \mathbf{q}_m = \begin{bmatrix} \mathbf{q}_{m-1} \\ c_m \eta_{m-1} \end{bmatrix}, \ \eta_m = -s_m \eta_{m-1} \quad (m = 2, 3, \dots).$$

Lineare Gleichungssysteme

Das LSQR-Verfahren für lineare Ausgleichsprobleme

Führt man jetzt noch die Richtungsvektoren

$$\mathbf{P}_m [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_m] := \mathbf{V}_m \mathbf{R}_m^{-1} \text{diag}(\mathbf{R}_m)$$

ein, die – wie ein Spaltenvergleich in $\mathbf{P}_m \text{diag}(\mathbf{R}_m)^{-1} \mathbf{R}_m = \mathbf{V}_m$ zeigt – mittels

$$\mathbf{p}_1 = \mathbf{v}_1, \quad \mathbf{p}_m = \mathbf{v}_m - (r_{m-1,m}/r_{m-1,m-1})\mathbf{p}_{m-1} \quad (m = 2, 3, \dots)$$

berechnet werden können, so erhalten wir schließlich eine Update-Formel für die LSQR-Näherungen

$$\begin{aligned} \mathbf{x}_m^{\text{LSQR}} &= \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m^{\text{LSQR}} = \mathbf{x}_0 + \beta_1 \mathbf{V}_m \mathbf{R}_m^{-1} \mathbf{q}_m = \mathbf{x}_0 + \beta_1 \mathbf{P}_m \text{diag}(\mathbf{R}_m)^{-1} \mathbf{q}_m \\ &= \mathbf{x}_0 + \beta_1 [\mathbf{P}_{m-1} \quad \mathbf{p}_m] \begin{bmatrix} \text{diag}(\mathbf{R}_{m-1}) & \mathbf{0} \\ \mathbf{0} & r_{m,m} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{q}_{m-1} \\ \eta_{m-1} \end{bmatrix} \\ &= \mathbf{x}_{m-1}^{\text{LSQR}} + (\beta_1 \eta_{m-1} / r_{m,m}) \mathbf{p}_m. \end{aligned}$$

Algorithmus 11: LSQR-Verfahren

Gegeben: Von Innenprodukten induzierte Normen $\|\cdot\|$ in \mathbb{C}^M bzw. \mathbb{C}^N ,
 $\mathbf{A} \in \mathbb{C}^{M \times N}$, $\mathbf{b} \in \mathbb{C}^M$ und eine Startnäherung $\mathbf{x}_0 \in \mathbb{C}^N$ für eine
Kleinste-Quadrate-Lösung von $\|\mathbf{b} - \mathbf{A}\mathbf{x}\| \rightarrow \min$

- 1 $\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$
- 2 $\mathbf{u}_1 \leftarrow \mathbf{r}_0$, $\beta_1 \leftarrow \|\mathbf{u}_1\|$, $\mathbf{u}_1 \leftarrow \mathbf{u}_1/\beta_1$
- 3 $\mathbf{v}_1 \leftarrow \mathbf{A}^* \mathbf{u}_1$, $\alpha_1 \leftarrow \|\mathbf{v}_1\|$, $\mathbf{v}_1 \leftarrow \mathbf{v}_1/\alpha_1$
- 4 $\tau_1 \leftarrow \alpha_1$, $\psi_1 \leftarrow \beta_1$, $\mathbf{p}_1 \leftarrow \mathbf{v}_1$
- 5 **for** $m = 1, 2, \dots$ **do**
- 6 $\mathbf{u}_{m+1} \leftarrow \mathbf{A}\mathbf{v}_m - \alpha_m \mathbf{u}_m$, $\beta_{m+1} \leftarrow \|\mathbf{u}_{m+1}\|$, $\mathbf{u}_{m+1} \leftarrow \mathbf{u}_{m+1}/\beta_{m+1}$
- 7 $\mathbf{v}_{m+1} \leftarrow \mathbf{A}^* \mathbf{u}_{m+1} - \beta_{m+1} \mathbf{v}_m$, $\alpha_{m+1} \leftarrow \|\mathbf{v}_{m+1}\|$, $\mathbf{v}_{m+1} \leftarrow \mathbf{v}_{m+1}/\alpha_{m+1}$
- 8 $r_{m,m} \leftarrow \sqrt{\tau_m^2 + \beta_{m+1}^2}$, $c_m \leftarrow \tau_m/r_{m,m}$, $s_m \leftarrow \beta_{m+1}/r_{m,m}$
- 9 $r_{m,m+1} \leftarrow s_m \alpha_{m+1}$, $\tau_{m+1} \leftarrow c_m \alpha_{m+1}$
- 10 $\varphi_m \leftarrow c_m \psi_m$, $\psi_{m+1} \leftarrow -s_m \psi_m$
- 11 $\mathbf{x}_m \leftarrow \mathbf{x}_{m-1} + (\varphi_m/r_{m,m}) \mathbf{p}_m$
- 12 $\|\mathbf{r}_m\| \leftarrow |\psi_{m+1}|$
- 13 $\mathbf{p}_{m+1} \leftarrow \mathbf{v}_{m+1} - (r_{m,m+1}/r_{m,m}) \mathbf{p}_m$

Wir befassen uns noch speziell mit der Lösung **unterbestimmter** LGS

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{b} \in \mathbb{C}^M, \quad \mathbf{A} \in \mathbb{C}^{M \times N}, \quad \text{Rang } \mathbf{A} = M < N. \quad (3.19)$$

- Das LGS (3.19) ist lösbar, die Lösung $\mathbf{x}^\dagger \in \mathbb{C}^N$ minimaler Norm durch die Forderung $\mathbf{x}^\dagger \perp \mathcal{N}(\mathbf{A})$ charakterisiert.
- Wegen $\mathcal{N}(\mathbf{A})^\perp = \mathcal{R}(\mathbf{A}^*)$ ist $\mathbf{x}^\dagger = \mathbf{A}^* \mathbf{y}$ für ein $\mathbf{y} \in \mathbb{C}^M$.
- Man kann damit \mathbf{x}^\dagger bestimmen durch Lösung der **Normalgleichungen 2. Art**

$$\mathbf{A}\mathbf{A}^* \mathbf{y} = \mathbf{b}, \quad \mathbf{x}^\dagger = \mathbf{A}^* \mathbf{y}.$$

- \mathbf{x}^\dagger ist (auch im Fall $\text{Rang } \mathbf{A} < M$) auch gegeben durch $\mathbf{A}^\dagger \mathbf{b}$, was man leicht mit Hilfe der Singulärwertzerlegung von \mathbf{A} bestätigt.
- Da unter den getroffenen Annahmen $\mathbf{A}\mathbf{A}^* \in \mathbb{C}^{M \times M}$ selbstadjungiert positiv definit ist, können wir zur Lösung der Normalgleichungen 2. Art das CG-Verfahren anwenden.

Algorithmus 12: CGNE-Verfahren (auch Craig-Verfahren)

Gegeben: Innenprodukte $(\cdot, \cdot)_1$ und $(\cdot, \cdot)_2$ in \mathbb{C}^M bzw. \mathbb{C}^N , $\mathbf{A} \in \mathbb{C}^{M \times N}$, $\mathbf{b} \in \mathbb{C}^M$ und eine Startnäherung $\mathbf{x}_0 = \mathbf{A}^* \mathbf{y}_0 \in \mathbb{C}^N$ für eine Kleinste-Quadrate-Lösung von $\|\mathbf{b} - \mathbf{A}\mathbf{x}\| \rightarrow \min$.

- 1 $\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{p}_1 \leftarrow \mathbf{A}^* \mathbf{r}_0$
- 2 **for** $m = 1, 2, \dots$ **do**
- 3 $\alpha_m \leftarrow (\mathbf{r}_{m-1}, \mathbf{r}_{m-1})_1 / (\mathbf{p}_m, \mathbf{p}_m)_2$
- 4 $\mathbf{x}_m \leftarrow \mathbf{x}_{m-1} + \alpha_m \mathbf{p}_m$
- 5 $\mathbf{r}_m \leftarrow \mathbf{r}_{m-1} - \alpha_m \mathbf{A}\mathbf{p}_m$
- 6 $\beta_m \leftarrow (\mathbf{r}_m, \mathbf{r}_m)_1 / (\mathbf{r}_{m-1}, \mathbf{r}_{m-1})_1$
- 7 $\mathbf{p}_{m+1} \leftarrow \mathbf{A}^* \mathbf{r}_m + \beta_m \mathbf{p}_m$

Aufwand beim CGNE-Verfahren:

- 2 Matrix-Vektor-Multiplikationen ($\mathbf{A}\mathbf{p}_m$, $\mathbf{A}^* \mathbf{r}_m$)
- 3 Vektor-Aufdatierungen
- 2 Innenprodukte
- Speicher: 5 Vektoren (\mathbf{x}_m , \mathbf{r}_m , \mathbf{p}_m , $\mathbf{A}\mathbf{p}_m$, $\mathbf{A}^* \mathbf{r}_m$)

Satz 3.18

Das CGNE-Verfahren mit Startnäherung $\mathbf{x}_0 = \mathbf{A}^* \mathbf{y}_0$, $\mathbf{y}_0 \in \mathbb{C}^M$ und zugehörigem Residuum $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$ konstruiert Näherungslösungen an die Minimum-Norm-Lösung $\mathbf{x}^\dagger = \mathbf{A}^\dagger \mathbf{b}$ des LGS (3.19) in dem verschobenen Krylov-Raum $\mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{r}_0)$.

Die Näherungen \mathbf{x}_m besitzen die Minimierungseigenschaft

$$\|\mathbf{x}_m - \mathbf{x}^\dagger\| = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{r}_0)} \|\mathbf{x} - \mathbf{x}^\dagger\|.$$

- Im Gegensatz zu CGLS und LSQR fällt die Residualnorm $\|\mathbf{r}_m\|$ bei CGNE nicht monoton.
- Im neueren LSMR-Verfahren von [Fong & Saunders, 2011] wird mittels GK-Bidiagonalisierung anstelle von $\|\mathbf{r}_m\|_2$ (wie bei LSQR) stattdessen in jedem Schritt $\|\mathbf{A}^* \mathbf{r}_m\|$ minimiert. Beide Größen sind monoton fallend mit m .
- [Estrin & al., 2017] schlagen ein weiteres fehlerminimierendes Krylov-Verfahren LSLQ vor.

- 1 Einleitung
- 2 Krylov-Unterraumverfahren
- 3 Lineare Gleichungssysteme
 - 3.1 Lösungsstrategien
 - 3.2 Selbstadjungierte Probleme
 - 3.3 Das Verfahren der konjugierten Gradienten
 - 3.4 Krylov-Verfahren für lineare Ausgleichsprobleme
 - 3.5 Vorkonditionierung
- 4 Matrixfunktionen
- 5 Krylov-Verfahren für Matrixfunktionen
- 6 Das QR-Verfahren für Eigenwertaufgaben

- Obwohl wir erst später und in größerer Allgemeinheit über die “Konvergenz”-Rate⁸ von Krylov-Verfahren sprechen werden, sei hier bemerkt, dass dies von Eigenschaften der Koeffizientenmatrix abhängt.
- Diese können das Spektralintervall, die Eigenwertverteilung oder der Wertebereich sein.
- Der Begriff **Vorkonditionierung** (engl. **preconditioning**) bezeichnet den Übergang von einem LGS $\mathbf{Ax} = \mathbf{b}$ zu einem äquivalenten LGS

$$\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b} \quad \text{bzw.} \quad \mathbf{AM}^{-1}\mathbf{y} = \mathbf{b}, \quad \mathbf{y} = \mathbf{Mx},$$

mit neuer Koeffizientenmatrix $\mathbf{M}^{-1}\mathbf{A}$ bzw. \mathbf{AM}^{-1} .

- Dabei ist die (invertierbare) Matrix \mathbf{M} eine Approximation an \mathbf{A} , bzw. \mathbf{M}^{-1} eine Approximation an \mathbf{A}^{-1} , wobei LGS mit \mathbf{M} ‘leicht’ zu lösen sind.
- Wir beschreiben hier die algorithmische Anbindung an bereits besprochene Krylov-Verfahren.

⁸Bei einem Verfahren, welches nach endlich vielen Schritten die exakte Lösung liefert, von Konvergenz zu sprechen ergibt zunächst keinen Sinn.

- Bei Krylov-Verfahren für nichtsymmetrische Systeme können beliebige Vorkonditionierer M verwendet werden; man unterscheidet

Links-Vorkonditionierung

$$M^{-1}Ax = M^{-1}b,$$

Rechts-Vorkonditionierung

$$AM^{-1}y = b, \quad y = Mx.$$

- Bei der Links-Vorkonditionierung entsteht neben der Folge der Residuen $r_m = b - Ax_m$ die der vorkonditionierten Residuen $z_m := M^{-1}r_m$.
- An die Stelle der Matrix-Vektor-Multiplikation $w \leftarrow Av$ tritt nun die Operation $w \leftarrow M^{-1}Av$, die in zwei Schritten erfolgen kann.
- Der Arnoldi-Prozess konstruiert dann eine ONB des Krylov-Raumes

$$\mathcal{K}_m^L := \mathcal{K}_m(M^{-1}A, M^{-1}r_0) = M^{-1}\mathcal{K}_m(AM^{-1}, r_0).$$

- Die Approximation x_m minimiert dann $\|M^{-1}(b - Ax)\|$ unter allen Vektoren

$$x \in x_0 + \mathcal{K}_m^L.$$

- Bei der Rechts-Vorkonditionierung muss die Hilfsvariable $\mathbf{y} = \mathbf{M}\mathbf{x}$ nie explizit berechnet werden.
- Beim Startresiduum $\mathbf{b} = \mathbf{A}\mathbf{M}^{-1}\mathbf{y}_0$ kann $\mathbf{M}^{-1}\mathbf{y}_0 = \mathbf{x}_0$ verwendet werden.
- Die generierte Krylov-Raum-Folge ist $\mathcal{K}_m^R = \mathcal{K}_m(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0) = \mathbf{M}\mathcal{K}_m^L$. Die Approximation $\mathbf{y}_m \in \mathbf{y}_0 + \mathcal{K}_m^R$ minimiert $\|\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\mathbf{y}\|$ über diesen affinen Unterraum.
- Die rücksubstituierte Variable $\mathbf{x}_m = \mathbf{M}^{-1}\mathbf{y}_m$ liegt dann im affinen Raum

$$\mathbf{M}^{-1}\mathbf{y}_0 + \mathbf{M}^{-1}\mathcal{K}_m^R = \mathbf{x}_0 + \mathcal{K}_m^L,$$

also im selben Raum wie bei der Links-Vorkonditionierung. (Um diese zu erhalten ist allerdings noch eine Anwendung von \mathbf{M}^{-1} erforderlich.)

- **Fazit:** in beiden Fällen wird die Näherung aus demselben Raum gewählt, bei Links-Vorkonditionierung wird $\|\mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x})\|$, bei Rechtsvorkonditionierung hingegen $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|$ minimiert.

- Ist M selbstadjungiert und positiv definit, so kann man den vorkonditionierten Algorithmus auch als standard-GMRES/FOM, aber bezüglich des Innenprodukts $(M\cdot, \cdot)$ interpretieren.
- Eine Variante, bei dem der Vorkonditionierer in jedem Schritt ein anderer sein kann ist als **FGMRES** (flexible GMRES) bekannt. Hier muss eine weitere Vektorfolge gespeichert werden.

- Auch wenn neben A auch M selbstadjungiert und positiv definit ist, gilt dies i.A. nicht für $M^{-1}A$ oder AM^{-1} zu gelten.
- Allerdings gilt für $(x, y)_M := (Mx, y) = (x, My)$

$$\begin{aligned}(M^{-1}Ax, y)_M &= (Ax, y) = (x, Ay) = (M^{-1}Mx, Ay) \\ &= (Mx, M^{-1}Ay) = (x, M^{-1}Ay)_M,\end{aligned}$$

d.h. $M^{-1}A$ ist selbstadjungiert (und positiv definit) bez. des Innenprodukts $(\cdot, \cdot)_M$.

- Dies legt die Idee nahe, im vorkonditionierte CG-Verfahren das durch M induzierte Innenprodukt zu verwenden.

Lineare Gleichungssysteme

Vorkonditionierung bei CG

Mit den Bezeichnungen $\mathbf{r}_m = \mathbf{b} - \mathbf{A}\mathbf{x}_m$ und $\mathbf{z}_m = \mathbf{M}^{-1}\mathbf{r}_m$ erhalten wir für CG angewandt auf $\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$

$$\mathbf{w} \leftarrow \mathbf{M}^{-1}\mathbf{A}\mathbf{p}_m$$

$$\alpha_m \leftarrow (\mathbf{M}\mathbf{z}_{m-1}, \mathbf{z}_{m-1}) / (\mathbf{M}\mathbf{w}, \mathbf{p}_m)$$

$$\mathbf{x}_m \leftarrow \mathbf{x}_{m-1} + \alpha_m \mathbf{p}_m$$

$$\mathbf{r}_m \leftarrow \mathbf{r}_{m-1} - \alpha_m \mathbf{A}\mathbf{p}_m, \quad \mathbf{z}_m \leftarrow \mathbf{M}^{-1}\mathbf{r}_m$$

$$\beta_m \leftarrow (\mathbf{M}\mathbf{z}_m, \mathbf{z}_m) / (\mathbf{M}\mathbf{z}_{m-1}, \mathbf{z}_{m-1})$$

$$\mathbf{p}_{m+1} \leftarrow \mathbf{z}_m + \beta_m \mathbf{p}_m$$

Wegen $(\mathbf{M}\mathbf{z}_m, \mathbf{z}_m) = (\mathbf{r}_m, \mathbf{z}_m)$ und $(\mathbf{M}\mathbf{w}, \mathbf{p}_m) = (\mathbf{A}\mathbf{p}_m, \mathbf{p}_m)$ müssen die \mathbf{M} -Innenprodukte nie explizit berechnet werden.

Algorithmus 13: Vorkonditioniertes CG-Verfahren

Gegeben: $A, M \in \mathbb{C}^{n \times n}$ selbstadjungiert und positiv definit bez.

Innenprodukt (\cdot, \cdot) auf \mathbb{C}^n , $\|\cdot\| = \sqrt{(\cdot, \cdot)}$, $\mathbf{b} \in \mathbb{C}^n$, $\mathbf{x}_0 \approx A^{-1}\mathbf{b}$

1 $\mathbf{r}_0 \leftarrow \mathbf{b} - A\mathbf{x}_0$, $\mathbf{z}_0 \leftarrow M^{-1}\mathbf{r}_0$, $\mathbf{p}_1 \leftarrow \mathbf{z}_0$

2 **for** $m = 1, 2, \dots$ **do**

3 $\mathbf{w} \leftarrow A\mathbf{p}_m$

4 $\alpha_m \leftarrow (\mathbf{r}_{m-1}, \mathbf{z}_{m-1}) / (\mathbf{w}, \mathbf{p}_m)$

5 $\mathbf{x}_m \leftarrow \mathbf{x}_{m-1} + \alpha_m \mathbf{p}_m$

6 $\mathbf{r}_m \leftarrow \mathbf{r}_{m-1} - \alpha_m \mathbf{w}$

7 $\mathbf{z}_m \leftarrow M^{-1}\mathbf{r}_m$

8 $\beta_m \leftarrow (\mathbf{r}_m, \mathbf{z}_m) / (\mathbf{r}_{m-1}, \mathbf{z}_{m-1})$

9 $\mathbf{p}_{m+1} \leftarrow \mathbf{z}_m + \beta_m \mathbf{p}_m$

- Ähnliche Überlegungen gibt es im selbstadjungierten indefiniten Fall für MINRES.
- Möchte man Krylov-Verfahren auf lineare Variationsgleichungen in Hilberträumen anwenden ($A : V \rightarrow V'$), so ist ein Vorkonditionierer erforderlich (um zurück nach V zu gelangen).
- Bei inversen (schlecht gestellten) Problemen ist die Vorkonditionierung deutlich trickreicher.
- Eine neuere Übersicht findet man bei [Wathen, 2015]⁹

⁹A. J. Wathen. Preconditioning. Acta Numerica, Volume 24, May 2015, pp. 329–376.