Kapitel 4

Iterative Verfahren zur Lösung von Linearen Gleichungssystemen

Situation: $A \in \mathbb{C}^{n\times n}$ schwach besetzt, n groß, $b \in \mathbb{C}^n$.

Ziel: Bestimme $x \in \mathbb{C}^n$ mit $Ax = b$.

4.1 Spliting-Methoden

Die Grundidee ist hier die Matrix in zwei Summanden aufzuteilen: $A = M - N$, so dass man das Problem in ein Fixpunktproblem umwandeln kann:

$$Mx = Nx + b$$

Dadurch ergibt sich ein Iterationsverfahren vermöge der Rekursion

$$Mx^{(k+1)} = Nx^{(k)} + b$$

Bemerkung: Sind A, M nichtsingulär und $\varrho\left(M^{-1}N\right) < 1$, wobei

$$\varrho\left(B\right) := \max_{\lambda\in\sigma(B)} |\lambda|,$$

dann konvergiert die Iteration für jeden Startwert x_0 gegen $A^{-1}b$ (siehe Übung).

Beispiel: Wir zerlegen $A = \underbrace{\begin{bmatrix} 0 & \dots & 0 \\ * & \ddots & \vdots \\ * & * & 0 \end{bmatrix}}_{=:L} + \underbrace{\begin{bmatrix} * & & 0 \\ & \ddots & \\ 0 & & * \end{bmatrix}}_{=:D} + \underbrace{\begin{bmatrix} 0 & * & * \\ \vdots & \ddots & * \\ 0 & \dots & 0 \end{bmatrix}}_{=:R}$.

(a) Setze $M = D$ und $N = -L - R$. DIes entspricht der *Jacobi-Iteration.*

(b) Setze $M = L + D$ und $N = -R$ entsprechend. Dies entspricht dem *Gauß-Seidel-Verfahren.*

4.2 Die Methode der konjugierten Gradienten (CG - „Conjugate Gradients“)

Spezialfall: $A \in \mathbb{R}^{n\times n}$ symmetrischen und positiv definit, $b \in \mathbb{R}^n$.

Grundidee: Betrachte

$$\varphi : \mathbb{R}^n \to \mathbb{R}, x \mapsto \varphi(x) = \frac{1}{2}x^T Ax - x^T b.$$

Dann gilt:

$$\nabla\varphi(x) = Ax - b, \quad \operatorname{Hess}\varphi(x) = A,$$

d.h. unser Lineares Gleichungssystem entspricht einem Extremwertproblem, denn in $\hat{x} = A^{-1}b$ ist das eindeutig bestimmte globale Minimum von φ:

$$\varphi(\hat{x}) = -\frac{1}{2}b^T A^{-1}b$$

Wir minimieren also φ einfach schrittweise und hoffen, dass wir uns dadurch auch der Lösung des linearen Gleichungssystems annähern.

4.2.1 Steilster Abstieg, Gradientensuchrichtung

Idee: φ fällt am stärksten in Richtung des negativen Gradienten ab:

$$-\nabla\varphi(x) = b - Ax$$

Definition: *Seien $A \in \mathbb{C}^{n\times n}$ und $x, b \in \mathbb{C}^n$. Dann heißt*

$$r = b - Ax$$

das Residuum von x bzgl. A und b.

Für $r \neq 0$ ist $\varphi(x + \alpha r) < \varphi(x)$ für ein $\alpha > 0$. Daher können wir φ verkleinern, indem wir so einen Parameter α bestimmen. Dabei minimieren wir in jedem Schritt über α:

Lemma 4.1 *Das Minimum von $\alpha \mapsto \varphi(x + \alpha r)$ ist in*

$$\alpha = \frac{r^T r}{r^T Ar}.$$

Beweis: Übung. □

Damit erhalten wir den folgenden Algorithmus.

Algorithmus („Steepest Descent“ bzw. „Steilster Abstieg“)
Berechnet für $A \in \mathbb{R}^{n\times n}$ symmetrisch positiv definit und $b \in \mathbb{R}^n$ die Lösung $x = A^{-1}b$ des linearen Gleichungssystems $Ax = b$.

1) Start: Wähle $x_0 \in \mathbb{R}^n$

2) Iteriere für $k = 1, 2, \ldots$ bis Konvergenz:

   (a) $r_{k-1} = b - Ax_{k-1}$

   (b) Falls $r_{k-1} = 0$ STOP! ($x_{k-1} = A^{-1}b$). Andernfalls setze $\alpha_k = \dfrac{r_{k-1}^T r_{k-1}}{r_{k-1}^T A r_{k-1}}$.

   (c) $x_k = x_{k-1} + \alpha_k r_{k-1}$

Bemerkung: Man kann zeigen, dass:

$$\varphi(x_{k+1}) + \frac{1}{2}b^T A^{-1} b \leq \left(1 - \frac{1}{\kappa_2(A)}\right)\left(\varphi(x_k) + \frac{1}{2}b^T A^{-1} b\right)$$

Wir erhalten also globale Konvergenz für alle Startwerte. Dabei müssen wir aber folgende Nachteile in Kauf nehmen:

(a) Die Konvergenz ist sehr langsam, falls $\kappa_2(A)$ groß ist.

(b) Die Konvergenzaussage bezieht sich auf φ, aber wenn φ schnell klein wird, muss dies nicht auch automatisch für das Residuum gelten.

Diese Nachteile entstehen uns im wesentlichen aus folgenden Gründen:

1) Wir minimieren nur über eine Suchrichtung r_k. Wir haben aber mehr Richtungen zur Verfügung (nämlich $r_0, \ldots, r_k$).

2) Die Suchrichtungen sind „nicht verschieden genug“.

4.2.2 A-konjugierte Suchrichtungen

Wir versuchen nun das Konvergenzverhalten des Algorithmus aus Abschnitt 4.2.1 durch eine kleine Modifikation zu verbessern. Die Grundidee dabei ist: Wählen wir in jedem Schritt statt des negativen Gradienten als Suchrichtung ein $p \in \mathbb{R}^n$, mit $p \not\perp r$, so finden wir auch in dieser Richtung (oder der Gegenrichtung) einen Abfall von φ. Wir wählen nun also in jedem Schritt statt r_k eine Suchrichtung p_k mit $p_k^T r_k \neq 0$.
Dazu stellen wir folgende **Forderungen** an die Wahl von p_{k+1} und x_{k+1}:

1) $p_1, \ldots, p_{k+1}$ sind linear unabhängig.

2) $\varphi(x_{k+1}) = \min\limits_{x \in \mathcal{R}_{k+1}} \varphi(x)$, wobei $\mathcal{R}_{k+1} := x_0 + \operatorname{Span}\{p_1, \ldots, p_{k+1}\}$.

3) x_{k+1} kann „leicht“ aus x_k berechnet werden.

Die Bedingungen 1) und 2) garantieren zusammen Konvergenz nach spätestens n Schritten, denn dann minimieren wir φ über den gesamten Raum $\mathbb{R}^n$.

Wir diskutieren im folgenden die Berechnung von p_{k+1} und x_{k+1}. Dazu seien die Suchrichtungen $p_1, \ldots, p_k \in \mathbb{R}^n$ und x_k mit $\varphi(x_k) = \min\limits_{x \in \mathcal{R}_k} \varphi(x)$ bereits bestimmt.

Gesucht: p_{k+1} und x_{k+1} mit $\varphi(x_{k+1}) = \min\limits_{x \in \mathcal{R}_{k+1}} \varphi(x)$, so dass 1)–3) erfüllt sind.

Dazu schreiben wir $x_k = x_0 + P_k y_k$ mit $P_k = [p_1, \ldots, p_k]$ und $y_k \in \mathbb{R}^k$ und machen den Ansatz

$$x_{k+1} = x_0 + P_k y + \alpha p_{k+1}$$

für $y \in \mathbb{R}^k, \alpha \in \mathbb{R}$. Unser Ziel ist dann die Bestimmung der Parameter y und α. Nun gilt:

$$\begin{aligned}
\varphi(x_{k+1}) &= \frac{1}{2}(x_0 + P_k y + \alpha p_{k+1})^T A (x_0 + P_k y + \alpha p_{k+1}) - \left(x_0 + P_k y + \alpha p_{k+1}^T\right) b \\
&= \varphi(x_0 + P_k y) + \alpha p_{k+1}^T A (x_0 + P_k y) - \alpha p_{k+1}^T b + \frac{1}{2}\alpha^2 p_{k+1}^T A p_{k+1} \\
&= \underbrace{\varphi(x_0 + P_k y)}_{\text{nur } y} + \alpha p_{k+1}^T A P_k y + \underbrace{\frac{1}{2}\alpha^2 p_{k+1}^T A p_{k+1} - \alpha p_{k+1}^T r_0}_{\text{nur } \alpha}
\end{aligned}$$

Wäre der störende Mischterm nicht, dann könnten wir getrennt über die beiden Variablen minimieren. Also wählen wir p_{k+1} so, dass gilt:

$$p_{k+1}^T A P_k = 0.$$

Damit erhalten wir:

$$\min_{x \in \mathcal{R}_{k+1}} \varphi(x) = \underbrace{\min_{y \in \mathbb{R}^k} \varphi(x_0 + P_k y)}_{\text{Lsg: } y = y_k} + \underbrace{\min_{\alpha \in \mathbb{R}} \left(\frac{1}{2}\alpha^2 p_{k+1}^T A p_{k+1}^T - \alpha p_{k+1}^T r_0\right)}_{\text{Lsg: } \alpha_{k+1} = \frac{p_{k+1}^T r_0}{p_{k+1}^T A p_{k+1}}}$$

Die erste Minimierungsaufgabe wird durch $y = y_k$ gelöst, denn $x_k = x_0 + P_k y_k$ erfüllt ja gerade

$$\varphi(x_k) = \min_{x \in \mathcal{R}_k} \varphi(x).$$

Die zweite Minimierungsaufgabe ist eine Minimierungsaufgabe über den reellen Zahlen und wird durch $\alpha_{k+1} = \frac{p_{k+1}^T r_0}{p_{k+1}^T A p_{k+1}}$ gelöst. Durch diese Vorgehensweise haben wir die Forderungen 2) und 3) erfüllt.

Fazit: Wähle A-konjugierte Suchrichtungen p_k, d.h. wähle

$$p_{k+1} \in \operatorname{Span}\{Ap_1, \ldots, Ap_k\}^\perp, k = 1, 2, \ldots$$

Dann folgt:

$$p_i^T A p_j = 0, \quad i \neq j, \quad i, j = 1, \ldots, k$$

d.h. $p_1, \ldots, p_k$ sind orthogonal bzgl. des Skalarprodukts:

$$\langle x, y \rangle_A := y^T A x$$

Es stellt sich nun die Frage, ob sich auch immer A-konjugierte Suchrichtungen finden lassen. Die Antwort erhalten wir aus dem folgenden Lemma.

Lemma 4.2 *Ist $r_k = b - Ax_k \neq 0$, so gibt es $p_{k+1} \in Span\{Ap_1, \ldots, Ap_k\}^\perp$ mit $p_{k+1}^T r_k \neq 0$.*

Beweis: Für $k = 0$ ist dies klar (wähle z.B. $p_1 = r_0$). Für $k \geq 1$ folgt wegen $r_k \neq 0$:

$$A^{-1}b \notin \mathcal{R}_k = x_0 + \text{Span}\{p_1, \ldots, p_k\},$$

d.h. insbesondere ist das Minimum von φ noch nicht erreicht. Somit ist dann auch

$$b \notin Ax_0 + \text{Span}\{Ap_1, \ldots, Ap_k\}$$

bzw.

$$r_0 = b - Ax_0 \notin \text{Span}\{Ap_1, \ldots, Ap_k\}.$$

Also gibt es $p_{k+1} \in \text{Span}\{Ap_1, \ldots, Ap_k\}^\perp$ mit $p_{k+1}^T r_0 \neq 0$. Wegen $x_k \in x_0 + \text{Span}\{p_1, \ldots, p_k\}$ gilt:

$$r_k = b - Ax_k \in r_0 + \text{Span}\{Ap_1, \ldots, Ap_k\}$$

also ist auch

$$p_{k+1}^T r_k = p_{k+1}^T r_0 \neq 0. \quad \square$$

Bemerkung: Wir halten folgende Beobachtung aus dem obigen Beweis fest: Wegen $p^T r_k = p^T r_0$ für $p \in \text{Span}\{Ap_1, \ldots, Ap_k\}^T$ gilt speziell $p_{k+1}^T r_k = p_{k+1}^T r_0$, also auch

$$\alpha_{k+1} = \frac{p_{k+1}^T r_0}{p_{k+1}^T A p_{k+1}} = \frac{p_{k+1}^T r_k}{p_{k+1}^T A p_{k+1}}$$

Wir zeigen nun, dass durch unsere Vorgehensweise auch die Forderung 1) erfüllt ist:

Lemma 4.3 *Die Suchrichtungen $p_1, \ldots, p_k$ sind linear unabhängig.*

Beweis: $P_k^T A P_k = \text{diag}\left(p_1^T A p_1, \ldots, p_k^T A p_k\right)$ ist insbesondere invertierbar (da A pos. def.). Also hat P_k vollen Rang, d.h. die Spalten $p_1, \ldots, p_k$ sind linear unabhängig. $\square$

Zusammenfassend erhalten wir folgenden Algorithmus:

Algorithmus (A-konjugierte Suchrichtungen)
Berechnet für $A \in \mathbb{R}^{n \times n}$ symmetrisch positiv definit und $b \in \mathbb{R}^n$ die Lösung $x = A^{-1}b$ des linearen Gleichungssystems $Ax = b$.

1) Start: Wähle $x_0 \in \mathbb{R}^n$

2) Iteriere für $k = 1, 2, \ldots$ bis Konvergenz:

   (a) $r_k = b - Ax_k$

   (b) Falls $r_k = 0$ STOP! ($x_k = A^{-1}b$). Andernfalls wähle $p_{k+1} \in \text{Span}\{Ap_1, \ldots, Ap_k\}^\perp$ mit $p_{k+1}^T r_k \neq 0$ und berechne

   $$\alpha_{k+1} = \frac{p_{k+1}^T r_k}{p_{k+1}^T A p_{k+1}}.$$

   (c) $x_{k+1} = x_k + \alpha_{k+1} p_{k+1}$

Man beachte, dass wir noch Freiheit in der Wahl von p_{k+1} haben.

4.2.3 „CG = steilster Abstieg + A-konjugierte Suchrichtungen“

Wie wir gesehen haben, bietet die Wahl von A-konjugierten Suchrichtungen einige Vorteile (einfache Berechnung von x_{k+1} aus x_k, Garantie der Konvergenz nach n Schritten). Andererseits möchten wir auch die Idee des „steilsten Abstiegs“nicht aufgeben, denn unsere Funktion φ fällt ja in Richtung des negativen Gradienten besonders schnell ab und wir sehen diese Richtung daher heuristisch als eine „gute Suchrichtung“an. Die Idee ist nun, die Freiheit in der Wahl von p_{k+1} demhingehend zu benutzen, d.h. wir wählen das p_{k+1}, welches „am nächsten“an r_k, also in Richtung des negativen Gradienten liegt. Wir wählen also das p_{k+1}, für das gilt

$$||p_{k+1} - r_k|| = \min_{p \in \mathrm{Span}\{Ap_1,\ldots,Ap_k\}^\perp} ||p - r_k|| \qquad (*)$$

Dies mutet zunächst eigenartig an, denn im Abschnitt 4.2.2. hatten wir uns Mühe gegeben, die Suchrichtungen so zu wählen, dass das zugehörige Optimierungsproblem besonders einfach gelöst werden kann. Und nun bestimmen wir die jeweilige Suchrichtung über eine neue Optimierungsaufgabe. Macht das überhaupt Sinn? Wir werden im Folgenden sehen, dass sich die neue Optimierungsaufgabe $(*)$ mit überraschender Einfachheit lösen lässt, denn es wird sich herausstellen, dass die neue Suchrichtung p_{k+1} einfach nur eine Linearkombination der vorhergehenden Suchrichtung p_k und des Residuums r_k ist.

Grundvoraussetzung: Im Folgenden seien mit denselben Bezeichnungen und Voraussetzungen wie in 4.2.2 die A-konjugierten Suchrichtungen so gewählt, dass (*) erfüllt ist für $k = 0, \ldots, m$. Ferner sei $P_k = [p_1, \ldots, p_k]$.

Ziel: Zeige $p_{k+1} \in \mathrm{Span}\,\{p_k, r_k\}$.

Lemma 4.4 *Sei $k \in \{1, \ldots, m\}$ und $z_k \in \mathbb{R}^k$ so, dass*

$$||r_k - AP_k z_k|| = \min_{z \in \mathbb{R}^k} ||r_k - APz||.$$

Dann gilt: $p_{k+1} = r_k - AP_k z_k$.

Beweis: Sei $\hat{p} := r_k - AP_k z_k$, dann ist durch die Voraussetzung des Lemmas $\hat{p}$ gerade die orthogonale Projektion von r_k auf $\mathcal{R}\,(AP_k)^\perp$, also ist

$$||\hat{p} - r_k|| = \min_{p \in \mathcal{R}(AP_k)^\perp} ||p - r_k||.$$

Damit folgt: $\hat{p} = p_{k+1}$. □

Satz 4.5 *Ist $r_k \neq 0$ für $k = 0, \ldots, m$, so gilt für $k = 0, \ldots, m$:*

1) $r_{k+1} = r_k - \alpha_{k+1} A p_{k+1}$
2) $\mathrm{Span}\,\{p_1, \ldots, p_{k+1}\} = \mathrm{Span}\,\{r_0, \ldots, r_k\} = \mathcal{K}_{k+1}\,(A, r_0)$
3) $r_{k+1} \perp r_j$ für $j = 0, \ldots, k$
4) $p_{k+1} \in \mathrm{Span}\,\{p_k, r_k\}$

Beweis:

1) Wegen $x_{k+1} = x_k + \alpha_{k+1} p_{k+1}$ gilt

$$r_{k+1} = b - Ax_{k+1} = \underbrace{b - Ax_k}_{= r_k} - \alpha_{k+1} A p_{k+1}.$$

2) Durch wiederholtes Anwenden von 1) folgt:

$$\operatorname{Span}\{Ap_1, \ldots, Ap_k\} \subseteq \operatorname{Span}\{r_0, \ldots, r_k\}, \quad k = 1, \ldots, m.$$

Im Lemma haben wir gezeigt, dass für alle $k = 0, \ldots, m$ gilt:

$$p_{k+1} = r_k - AP_k z_k \in \operatorname{Span}\{r_0, \ldots, r_k\}.$$

Damit erhalten wir

$$\operatorname{Span}\{p_1, \ldots, p_{k+1}\} \subseteq \operatorname{Span}\{r_0, \ldots, r_k\}$$

für $k = 0, \ldots, m$. Ferner gilt mit 1):

$$r_{k+1} \in \operatorname{Span}\{r_k, Ap_{k+1}\} = \operatorname{Span}\{r_k, Ar_0, \ldots, Ar_k\}$$

für $k = 0, \ldots, m$. Dann ist also:

$$\begin{array}{rcl} r_1 & \in & \operatorname{Span}\{r_0, Ar_0\}, \\ r_2 & \in & \operatorname{Span}\{r_0, Ar_0, Ar_1\} \subseteq \operatorname{Span}\left\{r_0, Ar_0, A^2 r_0\right\}, \\ usw. & \vdots & usw. \end{array}$$

Mit Induktion erhalten wir schließlich

$$\operatorname{Span}\{p_1, \ldots, p_{k+1}\} \subseteq \operatorname{Span}\{r_0, \ldots, r_k\} \subseteq \mathcal{K}_{k+1}(A, r_0).$$

Die Gleichheit folgt aus Dimensionsgründen.

3) Wir zeigen $P_k^T r_k = 0$ d.h. $p_1, \ldots, p_k \perp r_k$ für alle $k = 1, \ldots, m$. Wegen 2) gilt dann auch $r_0, \ldots, r_{k-1} \perp r_k$ wie gewünscht. Nun gilt $x_{k+1} = x_0 + P_k y_k$, wobei y_k die Funktion

$$\begin{aligned} \varphi(x_0 + P_k y) &= \frac{1}{2}(x_0 + P_k y)^T A(x_0 + P_k y) - (x_0 + P_k y)^T b \\ &= \varphi(x_0) + y^T P_k^T (Ax_0 - b) + \frac{1}{2} y^T P_k^T A P_k y \end{aligned}$$

minimiert. Der Gradient von $y \mapsto \varphi(x_0 + P_k y)$ wird also an der Stelle $y = y_k$ gleich Null, d.h. es gilt

$$P_k^T A P_k y_k + P_k^T (Ax_0 - b) = 0.$$

Dies ist gleichbedeutend mit $0 = P_k^T (b - Ax_0 - AP_k y_k) = P_k^T (b - Ax_k) = P_k^T r_k$.

4) Ist $k = 1$, so folgt mit 2), dass $p_2 \in \text{Span}\{r_0, r_1\}$. Wegen $p_1 = r_0$ gilt dann $p_2 \in \text{Span}\{p_1, r_1\}$. Für $k > 1$ partitionieren wir den Vektor z_k aus Lemma 4.4 als

$$z_k = \begin{bmatrix} w \\ \mu \end{bmatrix}, \quad w \in \mathbb{R}^{k-1},\ \mu \in \mathbb{R}.$$

Mit $r_k = r_{k-1} - \alpha_k A p_k$ wegen 1) erhalten wir aus Lemma 4.4:

$$\begin{aligned} p_{k+1} &= r_k - AP_k z_k \\ &= r_k - AP_{k-1} w - \mu A p_k \\ &= r_k - AP_{k-1} w + \frac{\mu}{\alpha_k}(r_k - r_{k-1}) \\ &= \left(1 + \frac{\mu}{\alpha_k}\right) r_k + s_k, \end{aligned}$$

wobei

$$\begin{aligned} s_k &= -\frac{\mu}{\alpha_k} r_{k-1} - AP_{k-1} w \\ &\in \text{Span}\{r_{k-1}, AP_{k-1} w\} \\ &\subseteq \text{Span}\{r_{k-1}, Ap_1, \ldots, Ap_{k-1}\} \\ &\subseteq \text{Span}\{r_0, \ldots, r_{k-1}\}. \end{aligned}$$

(Man beachte, dass α_k nach Konstruktion von Null verschieden ist!) Wegen 3) sind r_k und s_k dann orthogonal. Damit können wir das Optimierungsproblem in Lemma 4.4 lösen, indem wir w und μ bestimmen, so dass

$$\|p_{k+1}\|^2 = \left(1 + \frac{\mu}{\alpha_k}\right)^2 \|r_k\|^2 + \|s_k\|^2$$

minimal wird. Dann ist aber insbesondere s_k so, dass auch $\|s_k\|$ (bei festem μ und variablen w) minimal ist. Nun wird $\|r_{k-1} - AP_{k-1} z\|$ aber nach Lemma 4.4 durch $z = z_{k-1}$ minimiert und es ergibt sich $p_k = r_{k-1} - AP_{k-1} z_{k-1}$. Folglich ist s_k ein Vielfaches von p_k. Damit haben wir aber

$$p_{k+1} \in \text{Span}\{r_k, s_k\} = \text{Span}\{r_k, p_k\}. \quad \square$$

Folgerung: Gegebenenfalls nach Skalierung von p_{k+1} haben wir

$$p_{k+1} = r_k + \beta_k p_k.$$

Wegen $p_k^T A p_{k+1} = 0$ gilt außerdem:

$$\beta_k = -\frac{p_k^T A r_k}{p_k^T A p_k}.$$

Damit lässt sich p_{k+1} unmittelbar aus p_k und r_k konstruieren, ohne dass wir die Minimierungsaufgabe $(*)$ explizit lösen müssen.

Wir fassen nun die erzielten Ergebnisse in folgendem Algorithmus zusammen:

Algorithmus: (CG, Konjugierte-Gradienten-Methode - Hestenes/Stiefels, 1952)
Berechnet für $A \in \mathbb{R}^{n\times n}$ symmetrisch, positiv definit und $b \in \mathbb{R}^n$ die Lösung $x = A^{-1}b$ des LGS $Ax = b$.

1) Start: $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0$, $p_1 = r_0$

2) Iteriere, für $k = 1, 2, \ldots$ bis n oder Konvergenz:

   (a) $\alpha_k = \dfrac{p_k^T r_k - 1}{p_k^T A p_k}$

   (b) $x_k = x_{k-1} + \alpha_k p_k$

   (c) $r_k = b - Ax_k$

   (d) $\beta_{k+1} = -\dfrac{p_k^T A r_k}{p_k^T A p_k}$

   (e) $p_{k+1} = r_k + \beta_{k+1} p_k$

Bemerkung: Die Kürze und Einfachheit des Algorithmus lässt vergessen, wie viele theoretische Resultate sich in seinem Hintergrund verstecken. So ist z.B. Konvergenz des Algorithmus nach spätestens n Schritten garantiert, denn der *CG*-Algorithmus ist ja ein Spezialfall des Algorithmus mit A-konjugierten Suchrichtungen aus Abschnitt 4.2.2. Die Iterierte x_k erfüllt daher die Bedingung

$$\varphi(x_k) = \min_{x \in x_0 + \mathcal{R}_k} \varphi(x),$$

wobei $\varphi(x) = \frac{1}{2}x^T Ax - x^T b$ und $\mathcal{R}_k = x_0 + \mathrm{Span}\{p_1, \ldots, p_k\}$. Nun ist aber $\mathrm{Span}\{p_1, \ldots, p_k\} = \mathcal{K}_k(A, r_0)$ nach Satz 4.5, d.h. wir minimieren φ über dem affinen Krylovraum $x_0 + \mathcal{K}_k(A, r_0)$. Unsere Iterierte x_k erfüllt also

$$\varphi(x_k) = \min_{x \in x_0 + \mathcal{K}_k(A, r_0)} \varphi(x).$$

Aus diesem Grund nennt man den *CG*-Algorithmus ein **Krylovraumverfahren**.

4.2.4 Konvergenzeigenschaften von CG

Der Zusammenhang des *CG*-Algorithmus mit Krylovräumen erlaubt eine detaillierte Konvergenzanalyse. Dazu führen wir zunächst eine spezielle Norm ein:

Definition: *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit. Dann heißt die durch*

$$||x||_A := \sqrt{x^T A x}$$

definierte Norm auf $\mathbb{R}^n$ die A-Norm.

Ziel: Abschätzung des Fehlers

$$e_k := A^{-1}b - x_k = A^{-1}(b - Ax_k) = A^{-1}r_k$$

wobei (x_k) die durch *CG* erzeugte Iterationsfolge ist.

Satz 4.6 (Optimalität von CG im Sinne der A-Norm) *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit und (x_k) die für einen Startwert x_0 durch CG erzeugte Folge. Ist $r_{k-1} \neq 0$, so gilt:*

$$||e_k||_A = ||A^{-1}b - x_k||_A < ||A^{-1}b - x||_A$$

für alle $x \in x_0 + \mathcal{K}_k(A, r_0)$ mit $x_k \neq x$.

Beweis: Wir wissen: $x_k \in x_0 + \mathcal{K}_k(A, r_0)$. Sei nun $x \in x_0 + \mathcal{K}_k(A, r_0)$ beliebig und $\Delta x = x_k - x$, d.h. $\Delta x \in \mathcal{K}_k(A, r_0)$, sowie

$$\hat{e} := A^{-1}b - x = A^{-1}b - (x_k - \Delta x) = e_k + \Delta x$$

Dann gilt:

$$\begin{aligned} ||\hat{e}||_A^2 &= \hat{e}^T A\hat{e} = (e_k + \Delta x)^T A(e_k + \Delta x) \\ &= e_k^T A e_k + 2e_k^T A\Delta x + \Delta x^T A\Delta x \end{aligned}$$

und

$$2e_k^T A\Delta x = 2r_k^T A^{-1} A\Delta x = 2r_k^T \Delta x = 0$$

da $\Delta x \in \mathcal{K}_k(A, r_0) = \text{Span}\{r_0, \ldots, r_{k-1}\}$ und $r_k \perp r_j$ für $j = 0, \ldots, k-1$ gemäß Satz 4.5. Wir erhalten damit:

$$||\hat{e}||_A^2 = ||e_k||_A^2 + ||\Delta x||_A^2 > ||e_k||_A^2, \quad \text{falls } \Delta x \neq 0. \quad \square$$

Korollar 4.7 *Sei $\widetilde{\Pi}_k := \{p \in \Pi_k | p(0) = 1\}$. Mit den Bezeichnungen und Voraussetzungen aus Satz 4.6 (insbesondere $r_{k-1} \neq 0$), gibt es genau ein Polynom $p_k \in \widetilde{\Pi}_k$ mit*

$$||p_k(A)\, e_0||_A = \min_{p\in\widetilde{\Pi}_k} ||p(A)\, e_0||_A$$

Ferner gilt: $e_k = p_k(A)\, e_0$ und

$$\frac{||e_k||_A}{||e_0||_A} = \min_{p\in\widetilde{\Pi}_k} \frac{||p(A)\, e_0||_A}{||e_0||_A} \leq \inf_{p\in\widetilde{\Pi}_k} \max_{\lambda\in\sigma(A)} |p(\lambda)| \qquad (*)$$

Beweis: Es gilt: $x_k \in x_0 + \mathcal{K}_k(A, r_0)$, d.h.

$$x_k = x_0 + \hat{p}_{k-1}(A)\, r_0$$

für ein $\hat{p}_{k-1} \in \Pi_{k-1}$. Außerdem gilt:

$$r_k = b - Ax_k = \underbrace{b - Ax_0}_{=:r_0} - A\hat{p}_{k-1}(A)\, r_0$$

Damit erhalten wir

$$e_k = A^{-1}r_k = \underbrace{A^{-1}r_0}_{=e_0} - \hat{p}_{k-1}(A)\, r_0 = e_0 - \hat{p}_{k-1}(A)\, Ae_0 = \underbrace{(I - \hat{p}_{k-1}(A)\, A)}_{=p_k(A)\in\tilde{\Pi}_k} e_0$$

Damit folgt die Eindeutigkeit von p_k, sowie die Gleichheit in $(*)$ aus dem vorigen Satz. Für die Ungleichung in $(*)$ sei $(v_1, \ldots, v_n)$ eine Orthonormalbasis aus Eigenvektoren von A zu den Eigenwerten $\lambda_1, \ldots, \lambda_n$. Ferner sei $p \in \tilde{\Pi}_k$, sowie

$$e_0 = c_1 v_1 + \ldots + c_n v_n \qquad \text{mit } c_1, \ldots, c_n \in \mathbb{R}.$$

Dann gilt:

$$p(A)\, e_0 = c_1 p(\lambda_1)\, v_1 + \ldots + c_n p(\lambda_n)\, v_n.$$

Wegen der Orthogonalität der v_i erhalten wir

$$||e_0||_A^2 = e_0^T A e_0 = \sum_{i=1}^{n} c_i^2 \lambda_i$$

und

$$||p(A)\, e_0||_A^2 = \sum_{i=1}^{n} c_i^2 p(\lambda_i)^2\, \lambda_i \leq \max_{\lambda \in \sigma(A)} p(\lambda)^2 \sum_{i=1}^{n} c_i^2 \lambda_i.$$

Daraus folgt aber

$$\frac{||p(A)\, e_0||_A^2}{||e_0||_A^2} \leq \max_{\lambda \in \sigma(A)} |p(\lambda)|^2. \quad \square$$

Bemerkung:

1) Aus Korollar 4.7 können wir folgern, dass CG schnell konvergiert, falls A ein „gutes“Spektrum hat, d.h. für das Polynome p mit $p(0) = 1$ und kleinem Grad existieren, so dass $|p(\lambda)|$ für alle $\lambda \in \sigma(A)$ klein ist. Dies ist z.B. der Fall, falls

   (a) die Eigenwerte in Clustern auftreten,

   (b) alle Eigenwerte weit weg vom Ursprung liegen.
   (Dann ist $\kappa_2(A) = \frac{\lambda_{max}}{\lambda_{min}}$ nicht zu groß.)

2) Mit Hilfe von Tschebyscheffpolynomen kann man die folgende quantitive Abschätzung beweisen:

$$\frac{||e_k||_A}{||e_0||_A} \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k,$$

   wobei $\kappa := \kappa_2(A)$ und

$$\frac{||e_k||_2}{||e_0||_2} \leq 2\sqrt{\kappa} \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k.$$

3) Verbesserung der Konvergenzrate von CG erreicht man durch Vorkonditionierung:

   (a) Für allgemeine LGS: $Ax = b$ betrachte:

$$M^{-1} A x = M^{-1} b$$

   wobei $M^{-1}A$ ein “gutes” Spektrum hat und $Mz = c$ leicht zu lösen ist.

   (b) Für LGS $Ax = b$ mit A symmetrisch und positiv definit betrachte:

$$\left(C^{-1} A C^{-T}\right)(Cx) = C^{-1} b$$

   wobei $C^{-1}AC^{-T}$ ein “gutes” Spektrum hat und $Cz = d$ leicht gelöst werden kann. $C^{-1}AC^{-T}$ ist wieder symmetrisch und positiv definit.

4.2.5 CG und Lanczos

In diesem Abschnitt verwenden wir dieselben Bezeichnungen wie in den vorangegangenen Abschnitten. Betrachten wir dann einmal die folgenden Matrizen:

$$R_k = [r_0, \ldots, r_{k-1}], \quad P_k = [p_1, \ldots, p_k], \quad B_k = \begin{bmatrix} 1 & -\beta_2 & & 0 \\ & 1 & \ddots & \\ & & \ddots & -\beta_n \\ 0 & & & 1 \end{bmatrix}$$

Aus den Gleichungen $p_1 = r_0$ und $p_i = r_{i-1} + \beta_i p_{i-1}$ für $i = 2, \ldots, n$ (siehe Abschnitt 4.2.3) erhalten wir

$$R_k = P_k B_k.$$

Dann ist die Matrix $R_k^T A R_k$ aber tridiagonal, denn

$$R_k^T A R_k = B_k^T P_k^T A P_k B_k = B_k^T \begin{bmatrix} p_1^T A p_1 & & 0 \\ & \ddots & \\ 0 & & p_k^T A p_k \end{bmatrix} B_k.$$

Außerdem wissen wir aus Satz 4.5, dass die $r_0, \ldots, r_{k-1}$ orthogonal sind und einen Krylovraum aufspannen, d.h. $\frac{r_0}{\|r_0\|}, \ldots, \frac{r_{k-1}}{\|r_{k-1}\|}$ ist eine Orthonormalbasis von $\mathcal{K}_k(A, r_0)$.
Daraus ergibt sich eine sehr interessante Folgerung. Ist nämlich $q_1 := \frac{r_0}{\|r_0\|}$ und sind $q_1, \ldots, q_k$ die durch den Lanczos-Algorithmus erzeugten Vektoren, so gilt wegen des impliziten Q-Theorems

$$q_j = \pm \frac{r_{j-1}}{\|r_{j-1}\|}, \quad j = 1, \ldots, k.$$

Die beim Lanczosalgorithmus erzeugte Tridiagonalmatrix T_k entspricht also (bis auf einige Vorzeichen) der Matrix $R_k^T A R_k$. Wir merken uns also:

$$\boxed{\text{„}CG = \text{Lanczos“}}$$

Anwendung: Im Laufe des CG-Algorithmus können wir die Tridiagonalmatrix $R_k^T A R_k$ berechnen und erhalten damit Informationen über extreme Eigenwerte von A. Insbesondere lässt erhalten wir dadurch Information über die Konditionzahl $\kappa_2(A) = \frac{\lambda_{max}}{\lambda_{min}}$.

4.2.6 GMRES

Situation: $A \in \mathbb{C}n \times n$ invertierbar, n groß, A schwach besetzt, $b \in \mathbb{C}^n$
(A kann also Hermitesch und i.A. indefinit oder auch nicht-Hermitesch sein.)

Ziel: Bestimme $x \in \mathbb{C}^n$ mit $Ax = b$.

Im Abschnitt 4.2 haben wir festgestellt, dass (gewisse affine) Krylovräume „gute Suchräume“ sind, d.h. wir finden dort gute Approximationen an die gesuchte Lösung. Es liegt daher nahe, auch für den allgemeinen Fall ein Krylovraumverfahren zu verwenden. Im CG-Algorithmus haben wir benutzt, dass die gesuchte Lösung $\hat{x} + A^{-1}b$ das eindeutig bestimmte Minimun der Funktion $\varphi = \frac{1}{2}x^T Ax - x^T b$. Dies gilt aber i.A. nur unter der Voraussetzung, dass $A \in \mathbb{R}^{n \times n}$ symmetrisch positiv definit ist.

Idee: Zu einem gegebenen Startvektor $x_0 \in \mathbb{C}^n$ und $r_0 := b - Ax_0$ bestimme x_k mit

$$\|b - Ax_k\|_2 = \min_{x \in x_0 + \mathcal{K}_k(A, r_0)} \|b - Ax\|_2.$$

A Hermitesch	$\rightsquigarrow$	MINRES (mininmal residuals), Paige/Saunders 1975
A allgemein	$\rightsquigarrow$	GMRES (generalized minimal residuals), Saad/Schultz 1986

Frage: Wie lösen wir das Least-Squares-Problem $\|b - Ax_k\|_2 = \min_{x \in x_0 + \mathcal{K}_k(A, r_0)} \|b - Ax\|_2$?

Antwort: In Abschnitt 4.2.5. haben wir festgestellt, dass CG im wesentlichen dem Lanczos-Algorithmus entspricht. Nun befassen wir uns mit unsymmetrischen Matrizen, also erwarten wir:

$$\boxed{\text{„GMRES} = \text{Arnoldi“}}$$

Nach k Schritten des Arnoldi-Algorithmus haben wir die Arnoldi-Konfiguration

$$AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^T = Q_{k+1} H_{k+1}$$

mit $Q_k = [q_1, \dots q_k]$, $Q_{k+1} = [Q_k, q_{k+1}]$ isometrisch und

$$H_{k+1,k} = \begin{bmatrix} h_{11} & \dots & \dots & h_{1k} \\ h_{21} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & h_{k,k-1} & h_{kk} \\ 0 & \dots & 0 & h_{k+1,k} \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}.$$

Ist $q_1 = \frac{r_0}{\|r_0\|}$, so gilt $\text{Span}\{q_1, \dots, q_k\} = \mathcal{K}_k(A, r_0)$. Sei nun $x \in x_0 + \mathcal{K}_k(A, r_0)$, d.h. $x = x_0 + q_k y$ für ein $y \in \mathbb{C}^k$. Dann gilt

$$\begin{aligned} \|b - Ax\| &= \|b - A(x_0 + q_k y)\| \\ &= \|r_0 - AQ_k y\| \\ &= \|r_0 - Q_{k+1} H_{k+1,k} y\| \\ &= \|Q_{k+1}^* r_0 - H_{k+1,k} y\| \qquad \text{da } Q_{k+1} \text{ isometrisch} \\ &= \left\| \|r_0\| \cdot e_1 - H_{k+1,k} y \right\| \qquad \text{da } q_2, \dots, q_{k+1} \perp q_1 = \tfrac{r_0}{\|r_0\|}. \qquad (**) \end{aligned}$$

Erinnerung: Lösung von Least-Squares-Problemen $\|c - My\| \stackrel{!}{=} \min$, $M \in \mathbb{C}^{k\times n}$, $k \le n$:

1) berechne eine QR-Zerlegung von M:

$$M = QR, \quad Q \in \mathbb{C}^{n\times n} \text{ unitär}, \quad R = \begin{matrix} k \\ n-k \end{matrix} \overset{k}{\begin{bmatrix} R_1 \\ 0 \end{bmatrix}};$$

2) Da Q unitär ist, gilt

$$\|c - My\|^2 = \|Q^*c - Ry\|^2 = \left\| \begin{bmatrix} c_1 - R_1 y \\ c_2 \end{bmatrix} \right\|^2, \qquad \text{wobei} \quad Q^*c = \begin{matrix} k \\ n-k \end{matrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}.$$

Falls R_1 invertierbar ist, so wird dies minimal, wenn $R_1 y = c$. Löse also das lineare Gleichungssystem $R_1 y = c$.

Kommen wir zurück zu unserem Least-Squares-Problem $(**)$. Die Matrix $H_{k+1,k}$ ist in Hessenbergform und wir wollen das LS-Problem für alle k lösen. Angenommen, wir haben das Problem bereits für $k-1$ gelöst, d.h. wir haben eine QR-Zerlegung für $H_{k,k-1}$ berechnet:

$$H_{k,k-1} = \widetilde{Q}_k \widetilde{R}_k, \quad \widetilde{Q}_k \text{ unitär}, \quad \widetilde{R}_{k-1} = \begin{bmatrix} R_{k-1} \\ 0 \end{bmatrix}, \quad R_{k-1} \text{ obere Dreiecksmatrix.}$$

Dann gilt:

$$\begin{bmatrix} \widetilde{Q}_k^* & 0 \\ 0 & 1 \end{bmatrix} \cdot H_{k+1,k} = \begin{bmatrix} \widetilde{Q}_k^* & 0 \\ 0 & 1 \end{bmatrix} \left[\begin{array}{c|c} H_{k,k-1} & h_{kk} \\ \hline 0 & h_{k+1,k} \end{array}\right] = \left[\begin{array}{c|c} \widetilde{R}_{k-1} & \widetilde{Q}_k^* h_{kk} \\ \hline 0 & h_{k+1,k} \end{array}\right]$$

$$= \begin{matrix} \\ k-1 \\ 1 \\ 1 \end{matrix} \begin{matrix} \begin{matrix} k-1 & 1 \end{matrix} \\ \begin{bmatrix} R_{k-1} & * \\ 0 & * \\ 0 & h_{k+1,k} \end{bmatrix} \end{matrix}.$$

Das Element $h_{k+1,k}$ kann nun durch eine einzige Givens-Rotation eliminiert werden. Wir erhalten also eine QR-Zerlegung von $H_{k+1,k}$ aus der bereits berechneten von $H_{k,k-1}$ durch Anwenden einer Givens-Rotation und durch Berechnung von $\widetilde{Q}_k^* h_{kk}$ (kostet $\mathcal{O}(n)$ flops). Zusammenfassend erhalten wir den folgenden Algorithmus.

Algorithmus (GMRES) Berechnet für $A \in \mathbb{C}^{n \times n}$ invertierbar, $b \in \mathbb{C}^n$ und einen Startvektor $x_0 \in \mathbb{C}^n$ die Lösung $\hat{x} = A^{-1}b$ von $Ax = b$.

1) Start: $r_0 = b - Ax_0$, $h_{10} = \|r_0\|$.

2) Iteriere: für $k = 1, 2, \ldots$ bis Konvergenz:

   a) $q_k = \dfrac{r_k}{h_{k,k-1}}$

   b) $r_k = Aq_k - \displaystyle\sum_{j=1}^{k} h_{jk} q_j$ mit $h_{jk} = Q_j^* r_k$

   c) $h_{k+1,k} = \|r_k\|$

   d) bestimme y_k, so dass $\left\| \|r_0\| \cdot e_1 - H_{k+1,k} y_k \right\|$ minimal wird

   e) $x_k = x_0 + Q_k y_k$

Bemerkung: Wie *CG* lässt sich auch GMRES auf polynomiale Approximation in $\widetilde{\Pi}_k = \{p \in \Pi_k | p(0) = 1\}$ zurückführen:

$$x = x_0 + \hat{p}(A)r_0 \qquad \text{für ein } \hat{p} \in \Pi_{k-1},$$

da $x = x_0 + \mathcal{K}_k(A, r_0)$. Damit folgt

$$r_k := b_a x_k = b - Ax_0 - A\hat{p}(A)r_0 = \Big(I - A\hat{p}(A)\Big)r_0 = p(A)r_0 \qquad \text{für ein } p \in \widetilde{\Pi}_k.$$

Damit lässt sich GMRES umfornulieren zu der Aufgabe:

$$\text{Finde } p \in \widetilde{\Pi}_k, \text{ so dass } \|p(A)r_0\| \text{ minimal wird.}$$

Denn ist $p_k \in \widetilde{\Pi}_k$, so dass $r_k = p_k(A)r_0$, so gilt

$$\|r_k\| = \|p_k(A)r_0\| \leq \|p(A)r_0\|$$

für alle $p \in \widetilde{\Pi}_k$.

Satz 4.8 *Sei $A \in \mathbb{C}^{n\times n}$ diagonalisierbar und $V^{-1}AV = \Lambda$ diagonal. Dann gilt:*

$$\frac{\|r_k\|}{\|r_0\|} \leq \kappa(V) \inf_{p\in\widetilde{\Pi}_k} \max_{\lambda\in\sigma(A)} \Big|p(\lambda)\Big|.$$

Beweis: Für jedes Polynom $p \in \widetilde{\Pi}_k$ gilt:

$$\begin{aligned} \|p(A)\| &= \|p(V\Lambda V^{-1}\| = \|Vp(\Lambda)V^{-1}\| \\ &\leq \|V\| \cdot \|p(\Lambda)\| \cdot \|V^{-1}\| = \kappa(V) \cdot \|p(\Lambda)\|, \end{aligned}$$

sowie

$$\|p(\Lambda)\| = \max_{\lambda\in\sigma(A)} |p(\lambda)|,$$

da Λ diagonal ist. Damit erhalten wir

$$\begin{aligned} \|r_k\| &= \|p_k(A)r_0\| \leq \inf_{p\in\widetilde{\Pi}_k} \|p(A)r_0\| \leq \inf_{p\in\widetilde{\Pi}_k} \|p(A)\| \cdot \|r_0\| \\ &\leq \|r_0\| \cdot \kappa(V) \inf_{p\in\widetilde{\Pi}_k} \max_{\lambda\in\sigma(A)} |p(\lambda)|. \quad \square \end{aligned}$$

Folgerung: GMRES konvergiert schnell, falls

1) sich dass Spektrum von A „vernünftig“verhält;

2) $\kappa(V)$ klein ist, d.h. wenn A nicht zu weit von einer normalen Matrix entfernt ist (denn ist A normal, so kann die diagonalisierende Matrix V unitär gewählt werden, hat also Konditionszahl eins).

Bemerkung: Konvergenzbeschleunigung erhalten wir wieder durch Präkonditionierung, d.h. statt $Ax = b$ lösen wir das System $M^{-1}Ax = M^{-1}b$, wobei sich das LGS $My = c$ leicht lösen lassen muss.

Bemerkung: Methoden zur Lösung von $Ax = b$, A invertierbar mit $A \neq A^*$:

1) *CGN* (das *N* steht hier für „Normalengleichung“)
Statt $Ax = b$ betrachte die Normalengleichung, also das LGS $A^*Ax = A^*b$ mit der positiv definiten Matrix A^*A und löse dieses mit dem *CG*-Algorithmus.

   Nachteil: Quadrierung der Konditionszahl: $\kappa(A^*A) = \kappa(A)^2$.

   Vorteil: Die Eigenwerte von A^*A sind gerade die Quadrate der Singulärwerte von A. Daher ist *CGN* sinnvoll für Matrizen A mit „schlechtem Spektrum“, aber „guten Singulärwerten“.

2) *BiCG* (Biconjugate gradients)

   CG: das berechnete $x_k \in x_0 + \mathcal{K}_k(A, r_0)$ liefert $r_k \perp r_0, \ldots, r_{k_1}$,
   also $r_k \perp \mathcal{K}_k(A, r_0)$

   BiCG: wähle s_0 mit $s_0^* r_0 = 1$ und bestimme das $x_k \in x_0 + \mathcal{K}_k(A, r_0)$
   mit $r_k \perp \mathcal{K}_k(A, s_0)$

   Diese Vorgehensweise entspricht dem unsymmetrischen Lanczos-Algorithmus. Wir erhalten damit folgende Tabelle von Entsprechungen:

	$Ax = \lambda x$	$Ax = b$
$A = A^*$	Lanczos	CG
$A \neq A^*$	Arnoldi	GMRES
	Lanczos	BiCG

3) Übersicht über verschiedene Klassen von Krylovraummethoden

   gemeinsamer Nenner: $\mathcal{K} = \mathcal{K}_k(A, r_0)$ Krylovraum

   entscheidende Größe: $r_k = b - Ax_k$ (Residuum)

   a) Ritz-Galerkin-Ansatz: wähle $x_k \in x_0 + \mathcal{K}$, so dass $r_k \perp \mathcal{K}$
      $\rightsquigarrow$ CG, FOM, GENCG

   b) Minimale-Residuen-Ansatz: wähle $x_k \in x_0 + \mathcal{K}$, so dass $\|r_k\|$ minimal ist
      $\rightsquigarrow$ MINRES, GMRES, ORTHODIR

   c) Petrov-Galerkin-Ansatz: wähle $x_k \in x_0 + \mathcal{K}$, so dass $r_k \perp \mathcal{L}$, $\mathcal{L} \subseteq \mathbb{C}^n$, $\dim \mathcal{L} = k$
      $\rightsquigarrow$ BiCG, QMR

   d) Minimalfehler-Ansatz: wähle $x_k \in x_0 + \mathcal{K}$, so dass $\|x_k - A^{-1}b\|$ minimal ist
      $\rightsquigarrow$ SYMMLQ, GMERR

   Weiter gibt es noch hybride Methoden, wie (CGS,Bi-CGSTAB,...)

Zum Abschluss sei noch einmal auf das Zauberwort hingewiesen, um das niemand herumkommt, der sich intensiv mit der Lösung von linearen Gleichungssystemen beschäftigt:

Präkonditionierung