

Einführung in die Grundlagen der Numerik (WS 22/23)

Manuel Hinz

10. November 2022

Inhaltsverzeichnis

1	Orthogonalität	3
1.1	Grundlegende Definitionen	3
1.2	Bestapproximationseigenschaft	4
1.3	Orthonormalbasen	5
2	Das lineare Ausgleichsproblem	7
2.1	Problemstellung und Normalengleichung	7
2.2	Methode der Orthogonalisierung	9
2.3	Grundüberlegungen zu Orthogonalisierungsverfahren	10
2.4	QR -Zerlegung mittels Givens-Rotationen	11
2.5	QR -Zerlegung mittels Householder-Transformationen	13
2.6	Pseudoinverse	14
3	Iterative Verfahren für große, dünn besetzte, Gleichungssysteme	18
3.1	Motivation	18
3.2	Grundidee von Projektionsmethoden	19
3.3	Verfahren des steilsten Abstiegs	21
3.4	Krylovräume	22
3.5	Arnoldi-Verfahren	23
3.6	Verfahren der vollständigen Orthogonalisierung	24
3.7	Das GMRES-Verfahren	26
3.8	Der symmetrische Lanczos-Prozess	28

Vorwort

Diese Mitschrift von der Vorlesung Einführung in die Grundlagen der Numerik (Dölz, WS 2022/2023) wird von mir neben der Vorlesung geschrieben und ist dementsprechend Fehleranfällig. Fehler gerne an mh@mssh.dev!

Kapitel 1

Orthogonalität

1.1 Grundlegende Definitionen

Definition 1.1. Sei X ein \mathbb{R} -Vektorraum und $\langle \cdot, \cdot \rangle : X \times X \rightarrow \mathbb{R}$ eine Abbildung. $\langle \cdot, \cdot \rangle$ heißt **Skalarprodukt** oder inneres Produkt, falls

$$\forall f \in X \setminus \{0\} : \langle f, f \rangle > 0 \quad (\text{Positivität})$$

$$\forall f, g \in X : \langle f, g \rangle = \langle g, f \rangle \quad (\text{Symmetrie})$$

$$\forall \alpha, \beta \in \mathbb{R}, f, g, h \in X : \langle \alpha f + \beta g, h \rangle = \alpha \langle f, h \rangle + \beta \langle g, h \rangle \quad (\text{Linearität im ersten Argument})$$

Bemerkung 1.2. Symmetrie und Linearität im ersten Argument implizieren, dass $\langle \cdot, \cdot \rangle$ eine bilineare Abbildung ist.

Definition 1.3. Sei X ein \mathbb{R} -Vektorraum mit Skalarprodukt $\langle \cdot, \cdot \rangle$. Wir bezeichnen die zugehörige **Norm** (in Abhängigkeit von einem Vektor $f \in X$) mit

$$\|f\| = \sqrt{\langle f, f \rangle}.$$

Lemma 1.4. Sei X ein \mathbb{R} -Vektorraum mit Skalarprodukt $\langle \cdot, \cdot \rangle$. Dann gilt die Cauchy-Schwarz-Ungleichung:

$$\forall f, g \in X : \langle f, g \rangle \leq \|f\| \cdot \|g\| \quad (\text{C.S.})$$

mit Gleichheit genau dann, wenn f und g linear abhängig sind.

Beweis. O.B.d.A. $f, g \neq 0$, da sonst offensichtlich Gleichheit gilt. Sei $\alpha \neq 0$, dann gilt mit $f, g \in X$ und $\alpha \in \mathbb{R}$:

$$0 \leq \|f - \alpha g\|^2 = \langle f - \alpha g, f - \alpha g \rangle = \|f\|^2 - 2\alpha \langle f, g \rangle + \alpha^2 \|g\|^2$$

Wählen wir jetzt $\alpha = \frac{\langle f, g \rangle}{\|g\|^2}$ folgt:

$$\begin{aligned} 0 &\leq \|f\|^2 - \frac{2\langle f, g \rangle^2}{\|g\|^2} + \frac{\langle f, g \rangle^2}{\|g\|^2} \\ &\implies \langle f, g \rangle^2 \leq \|f\|^2 \cdot \|g\|^2. \end{aligned}$$

□

Eingefügte Bemerkung. Rechnung zur Begründung von $\langle f - \alpha g, f - \alpha g \rangle = \|f\|^2 - 2\alpha \langle f, g \rangle + \alpha^2 \|g\|^2$:

$$\begin{aligned} &\langle f - \alpha g, f - \alpha g \rangle \\ &= \langle f, f - \alpha g \rangle - \alpha \langle g, f - \alpha g \rangle \\ &= \langle f, f \rangle - \alpha \langle f, g \rangle - \alpha \langle g, f \rangle + \alpha^2 \langle g, g \rangle \\ &= \|f\|^2 - 2\alpha \langle f, g \rangle + \alpha^2 \|g\|^2 \end{aligned}$$

Beispiel 1.5. 1. $X = \mathbb{R}^n$ und $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$ (Euklidisches Skalarprodukt)

2. $X = \mathbb{R}^n$, $\langle x, y \rangle = x^\top A y$, wobei A positiv definit und symmetrisch ist

3. $I = [a, b]$, $w : I \rightarrow \mathbb{R}$ beschränkt und strikt positiv:

$$X = \left\{ f : I \rightarrow \mathbb{R} : \int_a^b f(x)^2 w(t) dt < \infty \right\} = L^2(I, w)$$

mit

$$\langle f, g \rangle = \int_a^b f(t)g(t)w(t)dt$$

Eingefügte Bemerkung. Die Definition von $L^2(I, w)$ ist hier nicht ganz richtig, man müsste natürlich noch Äquivalenzklassen, bzgl. Gleichheit bis auf Nullmengen, bilden. Dies wird hier, da Analysis 3 / Wtheo. nicht nicht vorausgesetzt wird, ignoriert.

Definition 1.6. Sei X ein \mathbb{R} -VR mit Skalarprodukt $\langle \cdot, \cdot \rangle$. $f, g \in X$ heißen **orthogonal**, falls $\langle f, g \rangle = 0$.

Bemerkung 1.7. Im \mathbb{R}^n mit dem euklidischen Skalarprodukt stimmt Definition 1.6, wegen

$$\langle x, y \rangle = \|x\| \|y\| \cos(\theta), \theta = \angle(x, y),$$

mit unserem bisherigen Verständnis überein.

1.2 Bestapproximationseigenschaft

Definition 1.8. Sei V ein \mathbb{R} -VR mit Skalarprodukt $\langle \cdot, \cdot \rangle$ und U ein Unterraum.

$$U^\perp = \{v \in V : \langle v, u \rangle = 0, \forall u \in U\}$$

heißt das **orthogonale Komplement** von U .

Satz 1.9. Unter den Annahmen von Definition 1.8 und der zusätzlichen Annahme, dass U endlich dimensional ist, gilt folgendes für $v \in V$:

$$\|v - u\| = \min_{w \in U} \|v - w\|$$

genau dann, wenn $v - u \in U^\perp$.

Beispiel 1.10. $V = \mathbb{R}^2$, $U = \text{span} \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$ mit euklidischem Skalarprodukt $\langle \cdot, \cdot \rangle$. Dann ist $U^\perp = \text{span} \left\{ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$.

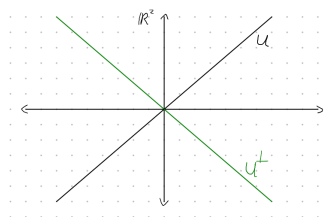


Abbildung 1.1: U und U^\perp

Beweis von Satz 1.9. Sei $v \in V$ und seien $u, w \in U$. Dann gilt:

$$\begin{aligned} \|v - w\|^2 &= \langle v - w, v - w \rangle = \langle (v - u) + (u - w), (v - u) + (u - w) \rangle \\ &= \|v - u\|^2 + 2 \underbrace{\langle v - u, u - w \rangle}_{\in U} + \|u - w\|^2 \geq \|v - u\|^2 \end{aligned}$$

mit Gleichheit genau dann, wenn $w - u = 0$ (da dann der $\|u - w\|$ Term verschwindet). □

Bemerkung 1.11. Der Satz sagt, dass es zu jedem $v \in V$ ein eindeutiges, bestmögliches $u \in U$ gibt.

Definition 1.12. Die Lösung aus Satz 1.9 heißt **orthogonale Projektion** von v auf U . Die Abbildung

$$P : V \rightarrow U, v \mapsto P(v) \text{ mit } \|v - Pv\| = \min_{w \in U} \|v - w\|$$

ist linear und wird **orthogonale Projektion** genannt.

Eingefügte Bemerkung (Beweis der Linearität). Für $v_1, v_2 \in V$ und $\alpha \in \mathbb{R}$ gilt:

$$\begin{aligned} v_1 - Pv_1 &\in U^\perp \\ v_2 - Pv_2 &\in U^\perp \end{aligned}$$

Daher

$$\alpha(v_1 - Pv_1) + (v_2 - Pv_2) = (\alpha v_1 + v_2) - (\alpha Pv_1 + Pv_2) \in U^\perp.$$

Aber dann muss $\alpha Pv_1 + Pv_2$ schon, wegen der Eindeutigkeit, $P(\alpha v_1 + v_2)$ sein.

Bemerkung 1.13. Satz 1.9 gilt auch, wenn U durch $W = w_0 + U$ ersetzt wird. Die orthogonale Projektion ist analog definiert..

Frage: Die Orthogonale Projektion hat offenbar gute Eigenschaften. Aber: wie berechnen wir sie? Wie wählen wir U ?

- Berechnung ist leicht
- U wählen schwierig

1.3 Orthonormalbasen

Definition 1.14. Sei X ein \mathbb{R} -VR mit Skalarprodukt $\langle \cdot, \cdot \rangle$ und $X_n \subset X$ ein endlich dimensionaler Teilraum mit Basis $\{\varphi_1, \dots, \varphi_n\}$. Die Basis heißt **Orthogonalbasis**, falls

$$\forall i \neq j : \langle \varphi_i, \varphi_j \rangle = 0$$

gilt und Orthonormalbasis (ONB), falls zusätzlich $\|\varphi_i\| = 1$ gilt. Das impliziert:

$$\langle \varphi_i, \varphi_j \rangle = \delta_{i,j}.$$

Beispiel 1.15. 1. \mathbb{R}^n mit euklidischem Skalarprodukt und kanonischer Basis

2. $X = L^2(I, 1)$ mit entsprechendem Skalarprodukt und X_n der Raum der trigonometrischen Polynome bis Grad n . Dann ist folgendes eine ONB:

$$\left\{ \frac{1}{\sqrt{2\pi}}, \frac{\sin(x)}{\sqrt{\pi}}, \frac{\cos(x)}{\sqrt{\pi}}, \dots, \frac{\sin(nx)}{\sqrt{\pi}}, \frac{\cos(nx)}{\sqrt{\pi}} \right\}$$

Eingefügte Bemerkung. Trigonometrische Polynome sind Funktionen der Form

$$f(t) = \sum_{k=1}^n a_k \cos(kx) + b_k \sin(kx).$$

Die größte Faktor vor dem x ist der Grad eines trigonometrischen Polynoms.

Satz 1.16. Sei $\{\varphi_1, \dots, \varphi_n\}$ eine ONB von $X_n \subset X$. Dann gilt

$$1. f = \sum_{i=1}^n \langle \varphi_i, f \rangle \varphi_i$$

2. $\|f\|^2 = \sum_{i=1}^n \langle \varphi_i, f \rangle^2$

3. Die orthogonale Projektion f_n von $f \in X \setminus X_n$ ist gegeben durch

$$f_n = \sum_{i=1}^n \langle \varphi_i, f \rangle \varphi_i$$

4. im Fall von 3.:

$$\|f_n\|^2 = \sum_{i=1}^n \langle \varphi_i, f \rangle^2 \leq \|f\|^2$$

Beweis. 1.:

$$\begin{aligned} f \in X_n &\implies \exists \alpha_i \in \mathbb{R} : f = \sum_{i=1}^n \alpha_i \varphi_i \\ \implies \langle \varphi_i, f \rangle &= \langle \varphi_i, \sum_{j=1}^n \alpha_j \varphi_j \rangle = \sum_{j=1}^n \alpha_j \langle \varphi_i, \varphi_j \rangle = \alpha_i \end{aligned}$$

2.:

$$\begin{aligned} \|f\|^2 &= \langle f, f \rangle \\ &= \left\langle \sum_{i=1}^n \alpha_i \varphi_i, \sum_{j=1}^n \alpha_j \varphi_j \right\rangle = \sum_{i,j=1}^n \alpha_i \alpha_j \delta_{i,j} = \sum_{i=1}^n \alpha_i^2 \end{aligned}$$

3.:

$f \in X \setminus X_n$:

$$\begin{aligned} \|f - \underbrace{\tilde{f}_n}_{\in X_n}\| &= \left\langle f - \sum_{i=1}^n \tilde{\alpha}_i \varphi_i, f - \sum_{i=1}^n \tilde{\alpha}_i \varphi_i \right\rangle \\ &= \|f\|^2 - 2 \sum_{i=1}^n \tilde{\alpha}_i \underbrace{\langle \varphi_i, f \rangle}_{=\alpha_i} + \sum_{i,j=1}^n \alpha_i \alpha_j \langle \varphi_i, \varphi_j \rangle \\ &= \|f\|^2 - \sum_{i=1}^n \tilde{\alpha}_i \alpha_i + \sum_{i=1}^n \tilde{\alpha}_i^2 \stackrel{\text{Quadratische Ergänzung}}{=} \|f\|^2 - \sum_{i=1}^n \alpha_i^2 + \sum_{i=1}^n \underbrace{(\alpha_i - \tilde{\alpha}_i)^2}_{\geq 0} \end{aligned} \quad (1.1)$$

Dies wird minimiert, wenn $\tilde{\alpha}_i = \alpha_i$ ist.

4.:

$f \in X_n$ wurde in 2. gezeigt. Sonst:

$$f \notin X_n \implies \text{mit } \alpha_i = \tilde{\alpha}_i \text{ in (1.1) :}$$

$$0 \leq \|f - f_n\|^2 = \|f\|^2 - \sum_{i=1}^n \underbrace{\alpha_i^2}_{\langle \varphi_i, f \rangle^2}$$

Es folgt die Behauptung. □

Vorteile von Orthogonalität:

- Bestapproximation
- Einfache Basisdarstellung

Kapitel 2

Das lineare Ausgleichsproblem

2.1 Problemstellung und Normalengleichung

Gegeben seien Punkte $(t_i, b_i) \in \mathbb{R}^2$ mit $i = 1, \dots, m$. Wir nehmen an, dass es eine Gestzmäßigkeit im Sinne eines parameterabhängigen Modelles

$$b_i = b(t_i) = b(t_i; \underbrace{x_1, \dots, x_n}_{\text{Parameter}}),$$

wobei die Parameter x_1, \dots, x_n unbekannt seien, gibt. In der Praxis sind die Messungen zusätzlich mit Fehlern behaftet und das Modell gilt nur approximativ. Zusätzlich gibt es oft mehr Messungen als Parameter, d.h. $m > n$.

Frage: Gegeben die Messungen, können wir zugehörige Parameter bestimmen?

Annahme: b ist linear in den Parametern, d.h. es gibt Funktionen

$$a_i : \mathbb{R} \rightarrow \mathbb{R}$$

s.d.

$$b(t; x_1, \dots, x_n) = a_1(t)x_1 + \dots + a_n(t)x_n.$$

Idee: Formuliere ein lineares Gleichungssystem:

$$b_i \approx b(t_i; x_1, \dots, x_n) = a_1(t_i)x_1 + \dots + a_n(t_i)x_n, i = 1, \dots, m$$

kurz $Ax \approx b$ mit $A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m$.

Problem: Durch Modell- und Messfehler gilt das Gleichungssystem nur ungefähr, und wir mehr Gleichungen als Unbekannte ("das Gleichungssystem ist überbestimmt"). Wir können unser Gleichungssystem also im Allgemeinen nicht lösen.

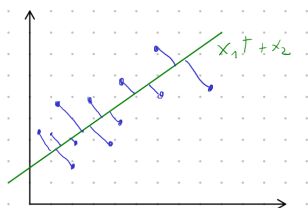


Abbildung 2.1: Datenpunkte und approximierte Gerade

Beispiel 2.1.

Idee: Finde Parameter, sodass das Modell "bestmöglich" mit den Messpunkten übereinstimmt, d.h. finde $(x_1, \dots, x_n)^t = x \in \mathbb{R}^n$ s.d.:

$$\|Ax - b\| = \min_{y \in \mathbb{R}^n} \|Ay - b\| \quad (2.1)$$

Definition 2.2. Die Gleichung (2.1) heißt **lineares Ausgleichsproblem**. Der Term $Ax - b$ heißt **Residuum**.

Bemerkung: $V = \mathbb{R}^m, U = \text{Bild}(A) \subset V, \dim(\text{Bild}(A)) \leq n \leq m$
Grundannahme

Statte V mit euklidischem Skalarprodukt aus.

$\xRightarrow{\text{Satz 1.9}}$ Es gibt genau ein $Ax \in \text{Bild}(A)$ so, dass

$$\|Ax - b\| = \min_{w \in U} \|w - b\|$$

gilt.

Aber: Wie berechnen wir x ?

Satz 2.3. Sei $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, m \geq n, x \in \mathbb{R}^n$ ist genau dann eine Lösung von (2.1) bezüglich der euklidischen Norm, falls

$$A^t Ax = A^t b. \quad (2.2)$$

Insbesondere ist das lineare Ausgleichsproblem genau dann lösbar, falls $\text{rang}(A) = n$.

Beweis.

$$\begin{aligned} \|Ax - b\| &= \min_{y \in \mathbb{R}^n} \|Ay - b\| \\ &\xLeftrightarrow{\text{Satz (1.9)}} Ax - b \in U^\perp = \text{Bild}(A)^\perp \\ &\iff \forall y \in \mathbb{R}^n : \langle Ax - b, Ay \rangle = 0 \\ &\iff \forall y \in \mathbb{R}^n : \langle A^t Ax - A^t b, y \rangle = 0 \\ &\iff A^t Ax = A^t b \end{aligned}$$

Die letzte Gleichung ist genau dann invertierbar, wenn $A^t A$ vollen Rang hat, also wenn A vollen Rang (n) hat. \square

Bemerkung 2.4. Im Beweis verwenden wir, dass $Ax - b$ orthogonal zu $U = \text{Bild}(A)$,

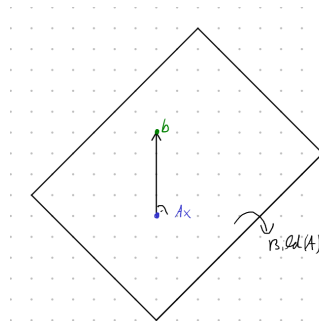


Abbildung 2.2: Hyperebene und Projektion

d.h. eine Normale zur Hyperebene $\text{Bild}(A)$ im \mathbb{R}^m , ist. Deshalb heißt (2.2) auch **Normalengleichung**.

Bemerkung 2.5. Für $m = n$ und $\text{rang}(A) = n$ ist die Lösung des linearen Ausgleichsproblems exakt (im mathematischen Sinne).

Satz 2.6. Für $A \in \mathbb{R}^{m \times n}$ ist $A^t A$ symmetrisch und positiv semidefinit. Falls $m \geq n$ ist $A^t A$ genau dann positiv definit, wenn $\text{rang}(A) = n$.

Beweis. • Symmetrisch: klar

• positiv semidefinit:

$$\forall x \in \mathbb{R}^n : x^t (A^t A) x = (Ax^t)(Ax) = \|Ax\|_2^2 \geq 0$$

- positiv definit: $\text{rang}(A) = n \implies Ax = 0 \iff x = 0 \implies \|Ax\|_2 = 0 \iff x = 0 \implies \text{Behauptung.}$

□

Einfachste Möglichkeit zur Lösung von (2.2): Berechne $A^t A$, $A^t b$, löse LGS mittels Cholesky. Kosten sind ungefähr:

$$\frac{n^2 m}{2} + m \cdot n + \frac{n^3}{6} + \frac{n^2}{2} + \frac{n^2}{2} \approx \frac{mn^2}{2} \text{ für } m \gg n.$$

Eingefügte Bemerkung. Anmerkung vom Dozent: $A^t A$ eig. immer schlecht zu berechnen.

Aber: Dieser Vorgang ist schlechter konditioniert als das lineare Ausgleichsproblem:

Eingeschobene Definition / Wiederholung

$$\begin{aligned} \text{cond}(A) &= \|A\| \|A^{-1}\| \\ \|A\| &= \max_{\|x\|=1} \|Ax\| \end{aligned}$$

Falls $A \in \mathbb{R}^{n \times n}$ spd (symmetrisch, positiv definit) gilt $\text{cond}_2((A^t A)) = \text{cond}_2(A)^2$.
Für $A \in \mathbb{R}^{m \times n}$ gelten ähnliche Überlegungen, siehe Deuffhard & Hohmann.

Beispiel 2.7. Sei $A = \begin{bmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}$ mit $\epsilon > \underbrace{\epsilon}_{\text{Maschinenengenauigkeit}}, \epsilon^2 < \epsilon$.

$$\implies A^t A = \begin{bmatrix} 1 + \epsilon^2 & 1 \\ 1 & 1 + \epsilon^2 \end{bmatrix} \stackrel{\text{im Computer}}{=} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$\implies A^t A$ ist im Computer singular, obwohl A vollen Rang hat!

Idee / Wunsch: Gebe einen Algorithmus an, der das lineare Ausgleichsproblem löst und nur auf A arbeitet.

2.2 Methode der Orthogonalisierung

Definition 2.8. Eine Matrix $Q \in \mathbb{R}^{n \times n}$ heißt **orthogonal**, wenn $Q^t Q = I$, d.h. falls die Spalten von Q eine ONB bzgl. des euklidischen Skalarprodukts bilden. Schreibe $Q \in O(n)$.

Notation: $\langle \cdot, \cdot \rangle_2, \|\cdot\|_2$ für das euklidische Skalarprodukt / die euklidische Norm.

Lemma 2.9. Für alle $Q \in O(n)$ gilt

1. $\|Qx\|_2 = \|x\|_2$ (Invarianz der Norm bzgl. orthogonaler Projektionen)
2. $\text{cond}_2(Q) = 1$

Beweis. 1.: $\|Qx\|_2^2 = \langle Qx, Qx \rangle_2 = \langle Q^t Qx, x \rangle_2 = \langle x, x \rangle_2 = \|x\|_2^2$

2.: $\|Q\|_2 = \max_{\|x\|_2=1} \|Qx\|_2 = 1$ und auch $\|Q^{-1}\|_2 = 1 \implies \text{Behauptung.}$

□

Satz 2.10. $A \in \mathbb{R}^{m \times n}, m \geq n, \text{rang}(A) = n$. Dann hat A eine QR-Zerlegung:

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$$

wobei $Q \in O(m), R \in \mathbb{R}^{n \times n}$ eine obere Dreiecksmatrix ist.

Beweis. Schreibe das Gram-Schmidt-Orthogonalisierungsverfahren in Matrixform:

$$Q = \underbrace{\begin{bmatrix} A_n & \dots & A_2 & A_1 \end{bmatrix}}_{[B_n \dots B_1]} \underbrace{\begin{bmatrix} 1 & \dots & \dots & \dots & -\frac{\langle A_n, A_1 \rangle_2}{\|A_1\|_2^2} \\ & \ddots & & & \vdots \\ & & 1 & -\frac{\langle A_3, A_2 \rangle_2}{\|A_2\|_2^2} & -\frac{\langle A_3, A_1 \rangle_2}{\|A_1\|_2^2} \\ & & & 1 & -\frac{\langle A_2, A_1 \rangle_2}{\|A_1\|_2^2} \\ \mathbf{0} & & & & 1 \end{bmatrix}}_{R'} \underbrace{\begin{bmatrix} \frac{1}{\|B_1\|_2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\|B_n\|_2} \end{bmatrix}}_{R''}$$

$\Rightarrow Q \in \mathbb{R}^{m \times n}$, $R'R''$ ist obere Dreiecksmatrix mit nicht-null Diagonaleinträgen

\Rightarrow invertierbar: $R = (R'R'')^{-1}$

$\Rightarrow QR = A$, wenn wir Q zu einer ONB von \mathbb{R}^m erweitern. □

— Ende von Vorlesung 02 am 13.10.2022 —

Satz 2.11. Sei $A \in \mathbb{R}^{m \times n}$, $m \geq n$, $\text{rang}(A) = n$, $b \in \mathbb{R}^n$. Sei $A = QR$ eine QR-Zerlegung von A und

$$\underbrace{Q^t A}_{=R} = Q^t b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad \begin{matrix} \in \mathbb{R}^n \\ \in \mathbb{R}^{m-n} \end{matrix}.$$

Dann ist $x = R_1^{-1}b_1$ die Lösung des linearen Ausgleichsproblems, wobei $R_1 \in \mathbb{R}^{n \times n}$ der obere Teil von R ist.

Beweis.

$$\begin{aligned} \|Ax - b\|_2^2 &\stackrel{\text{Lemma 2.9}}{=} \|Q^t(Ax - b)\|_2^2 \\ &= \left\| \begin{bmatrix} R_1 x - b \\ b_2 \end{bmatrix} \right\|_2^2 = \|R_1 x - b_1\|_2^2 + \|b_2\|_2^2 \\ &\geq \|b_2\|_2^2 \end{aligned}$$

$n = \text{rang}(A) = \text{rang}(R) = \text{rang}(R_1) \Rightarrow R_1$ invertierbar \Rightarrow Behauptung □

Problem:

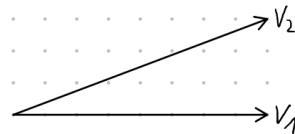


Abbildung 2.3: Problemstellung

$w_2 = v_2 - \frac{\langle v_2, v_1 \rangle_2}{\langle v_1, v_1 \rangle_2} v_1$ ist problematisch, falls $v_1 \approx v_2$ (Auslöschung). Beim Gram-Schmidt-Verfahren können Rundungsfehler auftreten. Es ist instabil.

Ziel: Stabiler Algorithmus um QR-Zerlegungen zu berechnen.

2.3 Grundüberlegungen zu Orthogonalisierungsverfahren

Problemstellung: Gegeben $v_1 = \alpha e_1 \in \mathbb{R}^2$, $v_2 \in \mathbb{R}^2$ transformiere v_2 auf $\tilde{w}_2 = \beta e_2$, gebe β an.

Gram-Schmidt: $\beta = \|w\|_2$

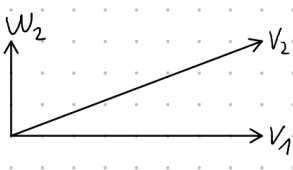


Abbildung 2.4: Gram-Schmidt

Drehungen: $\tilde{w}_2 = Qv_2$

$$Q = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix}$$

$$\beta = \|v_2\|_2$$

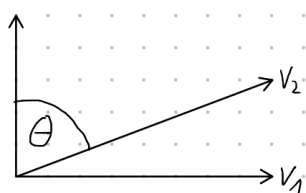


Abbildung 2.5: Drehungsansatz

Spiegelungen: $\tilde{w}_2 = Qv_2$, $Q = I - 2\frac{vv^t}{v^t v}$ und $\beta = \|v_2\|_2$

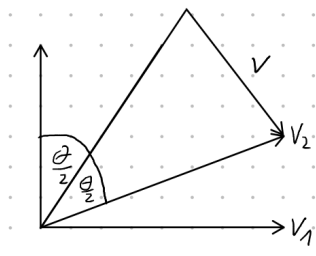


Abbildung 2.6: Spiegelungsansatz

Idee: Benutze orthogonale Transformationen Q_1, \dots, Q_n um $A \in \mathbb{R}^{m \times n}$, $\text{rang}(A) = n$, sukzessive zu reduzieren.

$$A \rightsquigarrow Q_1 A \rightsquigarrow Q_2 Q_1 A \rightsquigarrow \dots \rightsquigarrow \begin{bmatrix} R_1 \\ 0 \\ 0 \end{bmatrix}$$

Weil $\text{cond}_2(Q) = 1$ ist die Vorgehensweise stabil, bzw. gut konditioniert.

Aber: Wie wählen wir Q_1, \dots, Q_n ?

2.4 QR-Zerlegung mittels Givens-Rotationen

Definition 2.12. Eine Matrix der Form

$$\delta_{k,l} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & c & & s & \\ & & & 1 & & \\ & & -s & & c & \\ & & & & & 1 & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{bmatrix}$$

, wobei die s, c Einträge in der k, l ten Zeile / Spalte sind, heißen Givens-Rotationen.

Bemerkung: Für $c = \cos(\theta)$, $s = \sin(\theta)$ ist $\delta_{k,l}$ eine Drehung um θ in in der Koordinaten (k, l) . $\delta_{k,l}$ ist Orthogonal.

Frage: Wie wählen wir c, s ?

Gegeben $x \in \mathbb{R}^n$, eliminiere l te Koordinate zu 0.

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} x_k \\ x_l \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$$

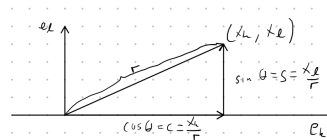


Abbildung 2.7: Trigonometriesetting

$$r^2 = x_k^2 + x_l^2 \implies \pm \sqrt{x_k^2 + x_l^2}$$

Aber: Diese Berechnungsweise ist nicht unbedingt stabil ($x_k \gg x_l$)

Stabile Variante:

$$\begin{aligned} \text{Falls } |x_l| > |x_k| &\implies \tau = \frac{x_k}{x_l}, s = \frac{1}{\sqrt{1+\tau^2}}, c = s\tau \\ \text{Sonst: } \tau = \frac{x_l}{x_k}, c &= \frac{1}{\sqrt{1+\tau^2}}, s = c\tau \end{aligned} \quad (2.3)$$

Beispielprozess:

$$\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \rightsquigarrow \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ 0 & * & * \end{bmatrix} \rightsquigarrow \begin{bmatrix} * & * & * \\ * & * & * \\ 0 & * & * \\ 0 & * & * \end{bmatrix} \rightsquigarrow \begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \end{bmatrix}$$

Algorithm 2.13

Input: $A \in \mathbb{R}^{m \times n}$, $m \geq n$

Output: R von der QR -Zerlegung (A wird zerstört "in place")

for $j = 1, \dots, n$ **do**

for $i = m, m-1, \dots, j+1$ **do**

 Berechne c, s wie in (2.3)

$$A[i-1:j, j:n] = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^t A[i-1:j, j:n]$$

end for

end for

$m \approx n$:

c, s : In jedem Eintrag einmal Wurzeln ziehen: $\implies \frac{n^2}{2}$ Quadratwurzeln und $\frac{4n^3}{3}$ Multiplikationen

$m \gg n$: $m \cdot n$ Quadratwurzeln und $2m \cdot n^2$ Multiplikationen

Bemerkung 2.14. Der Algorithmus 2.4 berechnet nur R von der QR-Zerlegung. Zur Berechnung von Q müssten zusätzliche Operationen investiert werden um die Givens-Rotation auf I anzuwenden.

Für das lineare Ausgleichsproblem benötigen wir $Q^t b$, weshalb wir den Algorithmus auf $\begin{bmatrix} A & | & b \end{bmatrix}$ anwenden können (da $R = Q^t A$).

Bemerkung 2.15. Für $m = n$ ist die QR-Zerlegung eine (teure) Alternative zur LR-Zerlegung.

2.5 QR-Zerlegung mittels Householder-Transformationen

Definition 2.16. Für $v \in \mathbb{R}^n, v \neq 0$, heißt

$$Q = I - 2 \frac{\overbrace{vv^t}^{\in \mathbb{R}^{n \times n}}}{\underbrace{v^t v}_{\in \mathbb{R}}}$$

Householder-Transformation / Reflexion / Spiegelung.

Wichtig!

Nicht vv^t berechnen, das ist sehr uneffizient!

Für $a, v \in \mathbb{R}^n, v \neq 0$ ist $Qa = \left(I - 2 \frac{vv^t}{v^t v}\right) a = a - 2 \frac{\langle v, a \rangle_2}{\langle v, v \rangle_2} v$

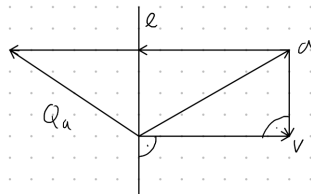


Abbildung 2.8: Householder-Transformationssetting

Qa ist a an l gespiegelt.

Lemma 2.17. Für eine Householder-Transformation $Q \in \mathbb{R}^{n \times n}$ gilt:

1. Q ist symmetrisch
2. Q ist orthogonal
3. Q ist involutionsch (eine Involution), d.h. $Q^2 = I$

Beweis. Nachrechnen. □

Frage: Gegeben $a \in \mathbb{R}^n$, wie müssen wir v wählen, so dass $Qa = \alpha e_1$ für $\alpha \in \mathbb{R}$?

Beobachte:

1. $|\alpha| = \|\alpha e_1\|_2 = \|Qa\|_2 = \|a\|_2$

$$2. \underbrace{a - 2 \frac{\langle v, a \rangle}{\langle v, v \rangle}}_{\in \mathbb{R}} v = Qa$$

$$\implies v \in \text{span}(\alpha e_1 - a) \implies \alpha = \pm \|a\|_2$$

$$\text{Vermeide Auslöschung} \implies \alpha = -\text{sign}(a_1) \cdot \|a\|_2$$

Effiziente Berechnung: Beobachte:

$$\begin{aligned} \|v\|_2^2 &= \langle v, v \rangle_2 = \langle a - \alpha e_1, a - \alpha e_1 \rangle_2 \\ &= \|a\|_2^2 - 2\alpha a_1 + \alpha^2 \\ &= -2\alpha(a_1 - \alpha) \\ \implies Qa &= a - 2 \frac{\langle v, a \rangle_2}{\|v\|_2^2} = a + \frac{\langle v, a \rangle_2}{\alpha(a_1 - \alpha)} v \end{aligned}$$

— Ende von Vorlesung 03 am 18.10.2022 —

Lemma 2.18. *Sie $\alpha \in \mathbb{R}^n, a \neq 0, a \notin \text{span}\{e_1\}$. Sei*

$$v = a - \alpha e_1, \alpha = -\text{sign}(a_1) \cdot \|a\|_2 \quad (2.4)$$

Dann ist

$$\left(I - 2 \frac{vv^t}{v^t v} \right) a = a + \frac{v^t a}{\alpha(a_1 - \alpha)} v = \alpha e_1. \quad (2.5)$$

Beweis. Siehe oben. □

Algorithm 2.19

Input: $A \in \mathbb{R}^{m \times n}, m \geq n$ “Mehr Zeilen als Spalten”

Output: $A \in \mathbb{R}^{m \times n}$, obere rechte Dreiecksmatrix R , Rest Householder-Transformationen

for $j = 1, \dots, n$ **do**

▷ Iterieren über die Spalten

Berechne v, α wie in (2.4), mit $a = A[j : m, j] \in \mathbb{R}^{m-j+1}$

$v = \frac{1}{v_1} v$

▷ Erster Eintrag wird nicht gespeichert, daher normalisieren wir

Berechne $A[j : m, j : n] = \left(I - 2 \frac{vv^t}{v^t v} \right) A[j : m, j : n]$ wie in (2.5)

if $j < m$ **then**

$A[j+1:m, j] = v[2:m-j+1]$

▷ Index startet von 1

end if

end for

Bemerkung 2.20. Die Skalierung $v = \frac{1}{v_1} v$ stellt sicher, dass die der erste Eintrag von v nicht gespeichert werden muss.

Aufwand: $m \sim n \rightsquigarrow \frac{2}{3}n^3$ Multiplikationen

$m \gg n \rightsquigarrow 2n^2m$ Multiplikationen

Schneller als Givensrotationen, stabiler als Normalengleichungen

2.6 Pseudoinverse

Ausgangspunkt: Wir wollen ein stabiles numerisches Verfahren, dass

$$Ax = b, A \in \mathbb{R}^{m \times n}, m \geq n, \text{rang}(A) = n, b \in \mathbb{R}^n$$

“lösen” kann, d.h. es gilt

$$\|Ax - b\|_2 = \min_{y \in \mathbb{R}^n} \|Ay - b\|_2$$

Mathematisch können wir die Abbildung $b \mapsto x$, wegen der Normalengleichung (2.2), schreiben als

$$x = \underbrace{(A^t A)^{-1} A^t}_{:= A^\dagger} b = A^\dagger b$$

$A^\dagger \in \mathbb{R}^{n \times m}$. Wegen $A^\dagger A = I$ heißt A^\dagger auch **Pseudoinverse**.

Frage: Können wir den Begriff der Inversen noch weiter verallgemeinern? Auf beliebige Matrizen?

Satz 1.9: $A \in \mathbb{R}^{m \times n}, U = \text{Bild}(A)$

$$\begin{aligned} \implies \|Ax - b\|_2 &= \min_{y \in \mathbb{R}^n} \|Ay - b\|_2 \xLeftrightarrow{\text{Satz 1.9}} Ax - b \in \text{Bild}(A)^\perp \\ \iff Ax - Pb - \underbrace{(b - Pb)}_{\in U^\perp: \text{Satz 1.9}} &\in \text{Bild}(A)^\perp, Pb \text{ ist die orthogonale Projektion von } b \text{ auf } U \\ \iff \underbrace{Ax}_{\in U} - \underbrace{Pb}_{\in U} &\in \text{Bild}(A)^\perp \\ \iff Ax = Pb \end{aligned}$$

Falls $\text{rang}(A) < n$ (z.B., falls $m < n$) ist $Ax = Pb$ nicht eindeutig lösbar (aber es existiert immer eine Lösung).

Für $\tilde{x} \in \mathbb{R}^n$ mit $A\tilde{x} = Pb, x' \in \ker(A)$ ist $A(\tilde{x} + x') = Pb$.

$$\begin{aligned} L(b) &= \left\{ x \in \mathbb{R}^n : \|Ax - b\|_2 = \min_{y \in \mathbb{R}^n} \|Ay - b\|_2 \right\} \\ &= \{x \in \mathbb{R}^n : Ax = Pb\} \\ &= \tilde{x} + \ker(A) \end{aligned}$$

Sind gewisse Lösungen sinnvoller als andere?

Wähle: $x \in \tilde{x} + \ker(A)$ mit minimaler Norm als "eindeutige" Lösung von $Ax = b$.

$$\begin{aligned} \xRightarrow{\text{Bem. 1.13}} \|x - 0\|_2 &= \min_{y \in \tilde{x} + \ker(A)} \|y - 0\|_2 \iff x \in (\tilde{x} + \ker(A))^\perp \\ &\iff x \in \ker(A)^\perp \end{aligned}$$

Anmerkung

Hier ist nicht ganz klar, was mit $(\tilde{x} + \ker(A))^\perp$ gemeint ist, da dies z.B. für $\ker(A) = \text{span}\{(0, 1)^t\}$ und $\tilde{x} = (1, 0)^t$ nur $\{0\}$ ist, was natürlich nicht der Intuition entspricht!

Statt der ursprünglichen Definition müssen wir hier wieder zurück schieben ($-\tilde{x}$ rechnen), was kein Problem ist, da wir o.B.d.A. $\tilde{x} \perp \ker(A)$ voraussetzen dürfen, bevor wir das Skalarprodukt berechnen!

Zum Beispiel ist also $v = (1, 0)^t$ im obigen Beispiel doch im orthogonalen Komplement, da $\langle v, \tilde{x} + u - \tilde{x} \rangle_2 = 0$ für $u \in \ker(A)$

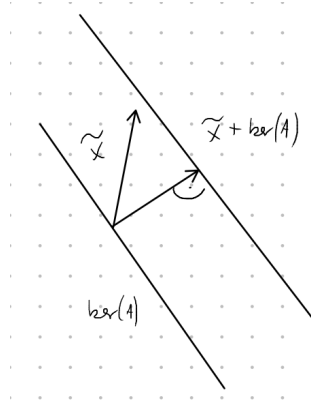


Abbildung 2.9: Setting

Bemerkung 2.21. Diese Wahl von x für $b \mapsto x$ ist linear: Für $b_1, b_2 \in \mathbb{R}^m$ ist:

$$\left. \begin{array}{l} Ax_1 = b_1 \quad x_1 \in \ker(A)^\perp \\ Ax_2 = b_2 \quad x_2 \in \ker(A)^\perp \end{array} \right\} \implies P(x_1 + x_2) = P(x_1) + P(x_2) = Ax_1 + Ax_2 = A(x_1 + x_2), x_1 + x_2 \in \ker(A)^\perp$$

Definition 2.22. Sei $A \in \mathbb{R}^{m \times n}$. Die Abbildungsmatrix $A^\dagger \in \mathbb{R}^{n \times m}$ von $b \mapsto x$ heißt **Pseudoinverse** oder **Moore-Pensore-Inverse** von A . D.h. gegeben $b \in \mathbb{R}^n$, dann ist $x = A^\dagger b$ die eindeutige Lösung von

$$\min_{y \in \ker(A)^\perp} \|Ay - b\|_2 = \|Ax - b\|_2.$$

Satz 2.23. $A \in \mathbb{R}^{m \times n}$. Dann ist $A^\dagger \in \mathbb{R}^{n \times m}$ eindeutig über die Moore-Penrose-Axiome definiert:

1. $(A^\dagger A)^t = AA^\dagger$
2. $(AA^\dagger)^t = A^\dagger A$
3. $A^\dagger AA^\dagger = A^\dagger$
4. $AA^\dagger A = A$

Beweis. Siehe Literatur oder später

□

Frage: Wie berechnen wir $x = A^\dagger b$?

Sei $A \in \mathbb{R}^{m \times n}$, $\text{rang}(A) = p \leq \min(m, n)$. Bringe A mittels orthogonaler Transformationen (z.B. Householder) auf obere Dreiecksgestalt, d.h.:

$$Q^t A = \begin{bmatrix} R & S \\ * & 0 & 0 \end{bmatrix} \quad (2.6)$$

wobei $S \in \mathbb{R}^{p \times (n-p)}$. Setze Analog $x = \begin{bmatrix} x_1 \in \mathbb{R}^p \\ x_2 \in \mathbb{R}^{n-p} \end{bmatrix}$, $Q^t b = \begin{bmatrix} b_1 \in \mathbb{R}^p \\ b_2 \in \mathbb{R}^{m-p} \end{bmatrix}$

Lemma 2.24. Mit obigen Bezeichnungen ist $x = A^\dagger b$ genau dann, wenn

$$x_1 = R^{-1}b_1 - R^{-1}Sx_2.$$

Beweis.

$$\begin{aligned}\|Ax - b\|_2^2 &= \|Q^t(Ax - b)\|_2^2 \\ &= \left\| \begin{pmatrix} Rx_1 + Sx_2 - b \\ -b_2 \end{pmatrix} \right\|_2^2 \\ &= \|Rx_1 + Sx_2 - b_1\|_2^2 + \|b_2\|_2^2\end{aligned}$$

ist minimal, falls $Rx_1 = b_1 - Sx_2$. □

Wir sehen $p = \text{rang}(A) = n \implies$ wie vorher, lineares Ausgleichsproblem!

Sonst: $x_2 = ?$

Lemma 2.25. Sei $p < n$, $V = R^{-1}S \in \mathbb{R}^{n \times (n-p)}$ und $u = R^{-1}b_1 \in \mathbb{R}^p$. Dann ist

$$\begin{aligned}x &= A^\dagger b \\ \iff (I + V^t V)x_2 &= V^t u \\ x_1 &= u - Vx_2\end{aligned}$$

Beweis.

$$\begin{aligned}\|x\|_2^2 &= \|x_1\|_2^2 + \|x_2\|_2^2 \\ &\stackrel{\text{Lemma 2.24}}{=} \|u - Vx_2\|_2^2 + \|x_2\|_2^2 \\ &= \|u\|_2^2 - 2\langle u, Vx_2 \rangle_2 + \langle Vx_2, Vx_2 \rangle_2 + \langle x_2, x_2 \rangle_2 \\ &= \|u\|_2^2 + \langle x_2, (I + V^t V)x_2 - 2V^t u \rangle_2 = \varphi(x_2)\end{aligned}$$

Minimiere $\varphi(x_2)$:

$$\begin{aligned}\varphi'(x_2) &= -2V^t u + 2(I + V^t V)x_2 \\ \varphi'(x_2) &= 2(I + V^t V) \implies \text{spd}\end{aligned}$$

φ minimal $\iff \varphi'(x_2) = 0 \implies$ Behauptung. □

Algorithm 2.26

Input: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

Output: $x = A^\dagger b$

Berechne QR -Zerlegung (2.6) von A

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = Q^t b$$

$V = R^{-1}S$ mittels Rückwertssubstitution

$u = R^{-1}b_1$ mittels Rückwertssubstitution

Löse $(I + V^t V)x_2 = V^t u$ mittels Cholesky-Zerlegung

$$x_1 = u - Vx_2$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

—Ende von Vorlesung 04 am 20.10.2022—

Kapitel 3

Iterative Verfahren für große, dünn besetzte, Gleichungssysteme

3.1 Motivation

Sei $\Omega \subset \mathbb{R}^d, d \in \mathbb{N}$. Betrachte die stationäre Wärmeleitungsgleichung, eine partielle Differenzialgleichung

$$\begin{cases} -\Delta u(x) = f(x) & x \in \Omega \\ u(x) = 0 & x \in \partial\Omega \end{cases} \quad (3.1)$$

mit Wärmequelle $f \in C(\Omega)$ und dem Laplace-Operator:

$$\Delta u = \sum_{i=1}^n \frac{\partial^2 u(x)}{\partial x_i^2}. \quad (3.2)$$

Die Lösung $u \in C^2(\Omega)$, falls existent, beschreibt die Temperaturverteilung im Raum Ω .

Diese Gleichung ist i.A. nicht von Hand lösbar!

Idee: Berechne approximative Lösung im Computer.

Ansatz: Für $g \in C^2(\mathbb{R})$ ist

$$\begin{aligned} g''(x) &= \lim_{h \searrow 0} \frac{g'(x+h) - g'(x)}{h} \approx \frac{g'(x+h) - g'(x)}{h} \\ &\approx \frac{\frac{g(x+h) - g(x)}{h} - \frac{g(x) - g(x-h)}{h}}{h} \\ &\approx \frac{g(x+h) - 2g(x) + g(x-h)}{h^2} \end{aligned}$$

\rightsquigarrow Ersetze $\frac{\partial^2 u}{\partial x_i^2}$ in (3.2)

\rightsquigarrow Überziehe Ω mit einem regelmäßigen Gitter mit Maschenweite $h = \frac{1}{n}, n \in \mathbb{N}$.

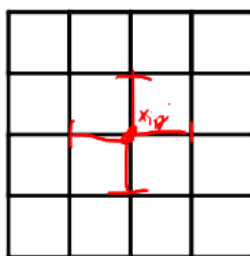


Abbildung 3.1: Gitter

Bezeichne die Gitterpunkte mit x_{ij} und $u_{ij} = u(x_{ij})$.

$$\stackrel{d=2}{\implies} \frac{1}{h^2} (4u_{ij} - u_{i+1j} - u_{i-1j} - u_{ij+1} - u_{ij-1}) = f_{ij} : i, j \in 1, \dots, n-1$$

$$u_{ij} = 0, i \in \{0, n\} \text{ oder } j \in \{0, n\}$$

Wir erhalten ein lineares Gleichungssystem mit $N = (n-1)^2$ Unbekannten und $O(1)$ Einträgen pro Zeile.
 $\implies A \in \mathbb{R}^{n \times n}$ hat $O(N)$ Einträge. Wir haben das Lösen einer (linearen) partiellen Differentialgleichung durch das Lösen eines linearen Gleichungssystems ersetzt.

Beispiel 3.1. $\Omega = (0, 1)^2, n = 4 \implies h = \frac{1}{4}$. Erhalte:

$$\left[\begin{array}{ccc|cc} 4 & -1 & & -1 & & \\ -1 & 4 & -1 & & -1 & \\ & -1 & 4 & & & -1 \\ \hline -1 & & & 4 & -1 & \\ & -1 & & -1 & 4 & -1 \\ & & -1 & & -1 & 4 \\ \hline & & & -1 & & \\ & & & & -1 & \\ & & & & & -1 \end{array} \right] \begin{bmatrix} u_{11} \\ u_{12} \\ u_{13} \\ u_{21} \\ u_{22} \\ u_{23} \\ u_{31} \\ u_{32} \\ u_{33} \end{bmatrix} = \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}$$

Aber: Um die Lösung von (3.1) gut zu approximieren ist oft $N \gg 1$ erforderlich. Für kleine bis mittlere N , d.h. in 2022 je nach Modell ~ 10 Millionen, sind graphenbasierte Löser eine Option.

Was tun für große N ?

Beobachtung: Matrix-Vektor-Multiplikation sind für dünn besetzte Matrizen in $O(N)$ berechenbar.

Frage: Wie bauen wir gute Löser für LGS (lineare Gleichungssysteme) nur unter Anwendung von Matrix-Vektor-Multiplikationen?

Idee: Benutze Orthogonalität um eine Bestapproximationseigenschaft zu erhalten.

3.2 Grundidee von Projektionsmethoden

Sei $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n$ und K, L Unterräume vom \mathbb{R}^n .

Idee: Finde eine approximative Lösung \tilde{x} zu $Ax = b$ mit

$$\tilde{x} \in K \text{ und } b - A\tilde{x} \perp_2 L$$

Kanonische Wahl: $L = AK$.

Falls wir eine Startnäherung x_0 zu x kennen, können wir \tilde{x} in $x_0 + K$ suchen:

Finde $\tilde{x} \in x_0 + K$ mit $b - A\tilde{x} \perp_2 L$

Beobachtung: $\tilde{x} \in x_0 + K \implies \exists d \in K : \tilde{x} = x_0 + d$

$$\implies \underbrace{b - A(x_0 + d)}_r \perp_2 L$$

$$\iff r_0 - Ad \perp_2 L$$

Eine approximative Lösung $\tilde{x} = x_0 + d$ muss also erfüllen:

$$\begin{cases} \tilde{x} = x_0 + d \\ \langle r_0 - Ad, w \rangle_2 = 0 \quad \forall w \in L \end{cases} \quad (3.3)$$

Idee: Wähle x_0, K, L , berechne $d \in K$ durch Lösen eines Unterproblems. Setze $x_1 = x_0 + d$, wähle neue Unterräume, beginne von vorne.

Wie implementieren wir diese Idee im Computer?

Sei $K = \text{span}\{v_1, \dots, v_n\}$, $L = \{w_1, \dots, w_n\}$

$V = [v_1 | \dots | v_n]$ und $W = [w_1 | \dots | w_n]$

(3.3) ist äquivalent zu

$$\begin{cases} \tilde{x} = x_0 + Vy & y \in \mathbb{R}^m \\ W_i^t AV y = W_i^t r_0 & i = 1, \dots, n \end{cases} \iff \underbrace{W^t A}_{m \times m} Vy = W^t r_0 \quad (3.4)$$

$$\implies \tilde{x} = x_0 + V(W^t AV)^{-1} W^t r_0$$

Algorithm 3.2 Prototyp einer iterativen Projektionsmethode

Input: $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, Fehlertoleranz α

Output: Näherung $x_{i+1} \approx x$

i=0

while Fehlertoleranz noch nicht erreicht **do**

 Wähle K_i, L_i

 Wähle Basen V, W von K_i, L_i

$r_1 = Ax_i$

$y = (W^t AV)^{-1} W^t r_i$

$x_{i+1} = x_i + Vy_i$

$i = i + 1$

end while

Aber: $W^t AV$ ist nicht notwendigerweise invertierbar:

Beispiel 3.3.

$$A = \left[\begin{array}{c|c} 0 & I \\ \hline I & I \end{array} \right] \in \mathbb{R}^{2m \times 2m}$$

$$K = L = \text{span}\{e_1, \dots, e_m\} \implies V = W = \left[\begin{array}{c} I_m \\ 0 \end{array} \right] \in \mathbb{R}^{2m \times m}$$

$$\implies W^t AV = 0 \text{ ist nicht invertierbar.}$$

Lemma 3.4. Sei einer der folgenden Bedingungen erfüllt:

1. A ist spd, $K = L$

2. A invertierbar, $L = AK$

Dann ist $W^t AV$ für alle Basen von K, L invertierbar.

Beweis. 1.: $L = K \implies W = V\delta$ mit $\delta \in \mathbb{R}^{m \times m}$ invertierbar.
 $\implies B = W^t AV = \delta^t V^t AV$

$$0 < \underbrace{y^t Ay}_{\substack{\text{spd, invertierbar}}}, y = Vx$$

2.: $L = AK \implies W = AV\delta, \delta \in \mathbb{R}^{m \times m}$ invertierbar

$$\implies B = W^t AV = \delta^t \underbrace{V^t A^t AV}_{\text{spd}} \implies \text{invertierbar} \implies \text{Beh.}$$

□

— Ende von Vorlesung 05 am 25.10.2022 —

3.3 Verfahren des steilsten Abstiegs

Idee: Wähle $K = L = \text{span}\{r_i\} = \text{span}\{b - Ax_i\}$

$$\begin{aligned} \implies x_{i+1} &= x_i + \underbrace{\alpha_i r_i}_{d_i \in K} \\ \implies \alpha &= \frac{r_i^t r_i}{r_i^t A r_i} \end{aligned}$$

Algorithm 3.5 Verfahren des steilsten Abstiegs

Input: A, b , Startvektor x_0 , Fehlertoleranz

Output: Näherung $x_{i+1} \approx x$

while Fehlertoleranz noch nicht erreicht **do**

▷ Praxis $\epsilon = 10^{-8}$, $\|r_i\|_2 < \epsilon$

$r_i = b - Ax_i$

$\alpha_i = \frac{r_i^t r_i}{r_i^t A r_i}$

$x_{i+1} = x_i + \alpha_i r_i$

end while

Bemerkung 3.6. Wegen (3.3) gilt:

$$\begin{aligned} 0 &= \langle r_i - Ad_i, r_i \rangle_2 \\ &= \langle b - Ax_i - Ad_i, r_i \rangle_2 \\ &= \langle b - Ax_{i+1}, r_i \rangle_2 \\ &= \langle Ax - Ax_{i+1}, r_i \rangle_2 \\ &= \langle x - x_{i+1}, r_i \rangle_A = (\star) \end{aligned}$$

Mit

$$\langle \cdot, \cdot \rangle_A = \langle A \cdot, \cdot \rangle_2$$

Aus (\star) folgt

$$\begin{aligned} 0 = (\star) &\iff x - x_{i+1} \perp_A r_i \\ &\iff x - x_{i+1} \perp_A x_i + \text{span}\{r_i\} \\ &\stackrel{\text{Satz 1.9}}{\iff} \|x - x_{i+1}\|_A = \min_{y \in x_i + \text{span}\{r_i\}} \|x - y\|_A \\ &\iff \frac{1}{2} \|x - x_{i+1}\|_A^2 = \min_{\alpha \in \mathbb{R}} \frac{1}{2} \|x - x_i - \alpha r_i\|_A^2 \end{aligned}$$

Betrachte $f(x_i) = \frac{1}{2} \|x - x_i\|_A^2 = \frac{1}{2} \langle A(x - x_i), x - x_i \rangle_2$

$$f'(x_i) = - \underbrace{Ax}_{=b} + Ax_i = -r_i$$

D.h. am Punkt x_i gehen wir in Richtung des steilsten Abstiegs,

Satz 3.7. Sei $A \in \mathbb{R}^{n \times n}$ spd. Dann gilt für die Iterierung des Verfahrens des steilsten Abstiegs dass:

$$\begin{aligned} \|x - x_{i+1}\|_A &\leq \frac{\lambda_{\max}(A) - \lambda_{\min}(A)}{\lambda_{\max}(A) + \lambda_{\min}(A)} \|x - x_i\|_A \\ &\stackrel{\frac{1}{\lambda_{\min}}}{=} \frac{\text{cond}_2(A) - 1}{\text{cond}_2(A) + 1} \|x - x_i\|_A \end{aligned}$$

wobei $\lambda_{\max}, \lambda_{\min}$ die größten, kleinsten Eigenwerte sind.

Beweis. Übung □

Bemerkung 3.8. Im Prinzip lassen sich mit Hilfe der Normalengleichung auch allgemeinere (invertierbare) Matrizen behandeln. Hierbei wird die Kondition verschlechtert d.h. die Konvergenz verschlechtert sich.

3.4 Krylovräume

Beobachte: Im Verfahren des steilsten Abstiegs gilt:

$$\begin{aligned} x_i &= x_0 + \alpha_0 r_0 + \dots + \alpha_{i-1} r_{i-1} = x_0 + \alpha_0 r_0 + \dots + (\alpha_{i-2} I + \alpha_{i-1} (I - \alpha_{i-2})) r_{i-2} \\ &= x_0 q_{i-1}(A) r_0, q_{i-1} \in \Pi_{i-1} \end{aligned}$$

Idee: Finde eine bessere Approximation von x_i in $x_0 + K_{i-1}(A, r_0)$.

Definition 3.9. Sei $A \in \mathbb{R}^{n \times n}, v \in \mathbb{R}^n, n \geq 1$. Der Raum

$$K_m = \text{span}(v, Av, A^2 v, \dots, A^{m-1} v)$$

heißt **Krylovraum** von A zu v .

Es gilt $K_m(A, v) \subseteq K_{m+1}(A, v)$

Lemma 3.10. Sei $\mathbb{R}^{n \times n}, v \in \mathbb{R}^n$. Dann gilt:

1. $\dim K_m(A, v) \leq \min\{m, n\}$
2. $\dim(K_m(A, v)) = \dim(K_{m+1}(A, v)) = m \implies \dim(K_{m+i}(A, v)) = m, i = 0, 1, \dots$
3. Für m wie im 2. gilt

$$\{Ax : x \in K_m(A, v)\} \subseteq K_m(A, v)$$

D.h. $K_m(A, v)$ ist invariant unter A .

Beweis. Übung □

Bemerkung 3.11. Betrachte $Ax = b$ mit Startnäherung x_0 , Residuum $r_0 = b - Ax_0$. Für $i = 0, \dots$ Wähle

$$\begin{aligned} x_{i+1} &\in x_0 + K_{i+1}(A, r_0) \\ \implies x_{i+1} x_0 &= q_i(A) r_0 \\ \implies r_{i+1} &= b - Ax_{i+1} = \underbrace{b - Ax_0}_{r_0} - Aq_i(A) r_0 \\ &= q_{i+1}(A) r_0 \in K_{i+2}(A, r_0) \end{aligned}$$

Das bedeutet: Sind K, L geeignete Krylovräume in einer Projektionsmethode, dann können wir immer garantieren, dass das Ergebnis in einem Krylovraum ist.

Aber: Die Vektoren $v, Av, \dots, A^n v$ sind numerisch keine guten Basen der Krylovräume, da sie zunehmend in eine ähnliche Richtung zeigen.

3.5 Arnoldi-Verfahren

Gesucht: Numerisch gutartige Basis von $K_m(A, v)$, welche einfach zu einer gutartigen Basis von $K_{m+1}(A, v)$ erweitert werden kann.

Idee: Arrangiere die Vektoren v, Av, \dots in einer "wachsenden" Matrix

$$[v | Av | A^2 v | \dots]$$

und wende auf jede Spalte ein Orthogonalisierungsverfahren (Gram-Schmidt, Householder, Givens) an.

Algorithm 3.12 Arnoldi-Verfahren (Gram-Schmidt Variante)

Input: $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n, m \in \mathbb{N}$

Output: $V_m = [v_1 | \dots | v_m]$ ONB von $K_m(A, v)$, $v_{m+1} \in \mathbb{R}^n, H_{m+1,m} \in \mathbb{R}^{(m+1) \times m}$

$$v_1 = \frac{v}{\|v\|_2}$$

for $j = 1, m$ **do**

$$z = Av_j$$

▷ $\neq A^{j-1}r_0$

$$h_{ij} = \langle z, v_i \rangle_2, i = 1, \dots, j$$

$$w_j = z - \sum_{i=1}^j h_{ij} v_i$$

$$h_{j+1,j} = \|w_j\|_2$$

if $h_{j+1,j} = 0$ **then**

stop

▷ Krylovraum stagniert

end if

$$v_{j+1} = \frac{w_j}{h_{j+1,j}}$$

end for

— Ende von Vorlesung 06 am 27.10.2022 —

Lemma 3.13. Falls das Arnoldi-Verfahren nicht vorzeitig abbricht, ist v_1, \dots, v_m eine ONB von $K_m(A, r_0)$.

Beweis. Orthogonalität: Ok

Orthonormal: Ok

Basis von $K_m(A, r_0)$: $j = 1$ ok

$j \implies j+1$

$h_{j+1,j} v_j = w_j$ (folgt aus der letzten Zeile von Algorithmus 3.5).

$$\begin{aligned} w_j &= \underbrace{Av_j}_{\in K_j(A, r_0) \implies v_j = q_{j-1}(A)v} - \underbrace{\sum_{i=1}^j h_{i,j} v_i}_{\tilde{q}_{j-1}(A)v, \tilde{q}_{j-1} \in \Pi_{j-1}} = (\star) \\ (\star) &= Aq_{j-1}(A)v_j - \tilde{q}_{j-1}(A)v_j = \underbrace{q_j(A)v}_{\in K_{j+1}(A, r_0)} \end{aligned}$$

□

Bemerkung: Diese Matrix $H_{h+1,j} \in \mathbb{R}^{(j+1) \times j}$ hat eine bestimmte Struktur, die Hessenberg-Struktur genannt wird.

Vorteile dieser Struktur

Zum Beispiel kann man eine QR-Zerlegung finden, in dem man Pro Spalte eine Givensrotation anwendet.

Lemma 3.14. Seien $V_m \in \mathbb{R}^{n \times n}, H_{m+1,m} \in \mathbb{R}^{(m+1) \times m}$, wie im Arnoldi-Verfahren erzeugt. Sei $H_{m,m} \in \mathbb{R}^{m \times m}$ wie $H_{m+1,m}$, aber ohne die letzte Zeile. Dann gilt:

$$\underbrace{A}_{\in \mathbb{R}^{n \times n}} \underbrace{V_m}_{\in \mathbb{R}^{n \times m}} = \underbrace{V_m}_{\in \mathbb{R}^{n \times m}} \underbrace{H_{m,m}}_{\in \mathbb{R}^{m \times m}} + \underbrace{w_m e_m^t}_{\in \mathbb{R}^{m \times m}} = \underbrace{V_{m+1}}_{\in \mathbb{R}^{n \times m+1}} \underbrace{H_{m+1,m}}_{\in \mathbb{R}^{m+1 \times m}} \quad (3.5)$$

$$V_m^t A V_m = H_{m,m} \quad (3.6)$$

Beweis. Gemäß Algorithmus haben wir

$$Av_j = z = \underbrace{w_j}_{h_{j+1,j}v_j} + \sum_{i=1}^j h_{ij}v_i = \sum_{i=1}^{j+1} h_{ij}v_i : j = 1, \dots, m$$

Daher gilt (3.5) (folgt aus Matrix Schreibweise).

Für (3.6):

$$V_m^t AV_m = \underbrace{V_m^t V_m}_{=I} H_{m,m} + \underbrace{V_m^t w_m e_m^t}_{=0} = H_{m,m}$$

□

Lemma 3.15. Sei j der Iterationsindex, bei dem das Arnoldi-Verfahren das erste Mal abbricht. Dann gilt:

$$K_j(A, r_0) = \dots = K_m(A, r_0)$$

$$AV_m = V_m H_{m,m}$$

Beweis. Übung.

□

3.6 Verfahren der vollständigen Orthogonalisierung

Ziel: Kombinieren von unserem Wissen über Projektionsmethoden mit demjenigen Wissen über Krylovräume.

Zutaten:

- Finde $\tilde{x} \in x_0 + K$ s.d. $bA\tilde{x} \perp L$
- Wähle K, L als Krylovräume in jeder Iteration
- Wähle Basen V, W von K, L in jeder Iteration

$$(3.4) \implies \begin{cases} \tilde{x} = x_0 + Vy \\ W^t AVy = w^t r_0 \end{cases}$$

Idee: Setze $r_0 = b - Ax_0$, $K = L = K_m(A, r_0)$ im m -ter Iteration, $V = W$ ONB von $K_m(A, r_0)$, berechnet mittels Arnoldi-Verfahren.

Zutaten: aktualisiert:

- $r_0 = b - Ax_0$
- $\beta = \|r_0\|_2, v_1 = \frac{r_0}{\beta}$
-

$$\begin{cases} x_m = x_0 + V_m y_m \\ \underbrace{V_m^t AV_m}_{\stackrel{3.6}{=} H_{m,m}} \underbrace{y_m}_{\beta v_1} = V_m^t r_0 \iff H_{m,m} y_m = \beta e_1 \end{cases}$$

Algorithm 3.16 Verfahren der vollständigen Orthogonalisierung**Input:** $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$, $m \in \mathbb{N}$ **Output:** $x_m \in x_0 + K_m(A, r_0)$, $b - Ax_m \perp K_m(A, r_0)$, $x_m \approx A^{-1}b$

$$r_0 = b - Ax_0, \beta = \|r_0\|_2, v_1 = \frac{r_0}{\beta}$$

for $j = 1, m$ **do**Bestimme $V_j, H_{j,j}$ mit Arnoldi-Verfahren (Stop beim Abbruch)Löse $\underbrace{H_{j,j}}_{\in \mathbb{R}^{j \times j}} y_j = \beta e_1$

$$x_j = x_0 + V_j y_j$$

Konvergenztest

end for

Was ist ein geeigneter Konvergenztest?

Lemma 3.17. *Im Algorithmus 3.6 gilt*

$$r_m = b - Ax_m = -h_{m+1,m}(e_m^t y_m) V_{m+1}$$

d.h. es gilt auch

$$\|r_m\|_2 = \underbrace{|h_{m+1,m}(e_m^t y_m)|}_{>0} = h_{m+1,m} |e_m^t y_m|$$

Beweis.

$$\begin{aligned} b - Ax_m &= b - A(x_0 + V_m y_m) \\ &= r_0 - AV_m y_m = (\star) \end{aligned}$$

$$\begin{aligned} (\star) &= \underbrace{r_0}_{\beta v_1} - \underbrace{AV_m}_{\stackrel{3.5}{=} V_m H_{m,m} - w_m e_m^t} \\ &= \beta v_1 - \underbrace{V_m \underbrace{H_{m,m} y_m}_{=\beta e_1}}_{\beta v_1} - \underbrace{w_m}_{=h_{m+1,m}} (e_m^t y_m) \\ &= -h_{m+1,m}(e_m^t y_m) v_{m+1} \end{aligned}$$

□

Bemerkung:

- Hauptkosten (Alles mit Vektoren der Länge n)
 - Matrix-Vektor-Multiplikation (1, Mal, sehr teuer)
 - Skalarprodukte
 - Vektor-Updates
- \implies Berechnen des Residuums wie in Lemma (3.17) lohnt sich.
- Speicherbedarf und Aufwand per Iteration werden in jeder Iteration teuer!

Umgehen von großen m Wir können Neustarten, um m wieder auf 1 zu setzen und hohe Kosten von großen m zu verhindern.

3.7 Das GMRES-Verfahren

Idee: Lemma (3.17) gibt uns eine explizite Darstellung von $\|r_j\|_2$. Können wir die Idee dahinter benutzen, um $\|r_j\|_2$ in jeder Iteration zu minimieren?

$$\begin{aligned} r_j &= b - A \underbrace{x_j}_{=x_0 + V_j y_j} \\ &= \underbrace{b - Ax_0}_{=r_0 = \beta v_1 = V_{m+1} \beta e_1}_{\in \mathbb{R}^{m+1}} - \underbrace{AV_j}_{\stackrel{3.5}{=} V_{j+1} H_{j+1,j}} y_j \\ &= V_{j+1}(\beta e_1 - H_{j+1,j} y_j) \end{aligned}$$

$$\Rightarrow \|r_j\|_2 = \|V_{j+1}(\beta e_1 - H_{j+1,j} y_j)\|_2$$

$$\text{Da } \|V_{j+1} z_{j+1}\|_2^2 = z_{j+1}^t z_{j+1} = \|z_{j+1}\|_2^2$$

$$\Rightarrow \|r_j\|_2 = \left\| \underbrace{\beta e_1}_{\in \mathbb{R}^{j+1}} - \underbrace{H_{j+1,j}}_{\in \mathbb{R}^{j+1 \times j}} \underbrace{y_j}_{\in \mathbb{R}^j} \right\| \quad (3.7)$$

$\Rightarrow \|r_j\|_2$ wird minimal, falls y_j als Lösung des linearen Ausgleichsproblems $H_{j+1,j} y_j = \beta e_1$ gewählt wird.

Algorithm 3.18 GMRES-Verfahren, Generalized Minimal Residual Method, Prototyp

Input: $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n, x_0 \in \mathbb{R}^n, m \in \mathbb{N}$

Output: $x_m \in x_0 + K_m(A, r_0), \|r_m\|_2 = \|b - Ax_m\|_2$ minimal

$$r_0 = b - Ax_0, \beta = \|r_0\|_2, v_1 = \frac{r_0}{\beta}$$

for $j = 1, m$ **do**

 Bestimme $V_{j+1}, H_{j+1,j}$ mit Arnoldi-Verfahren (Stop beim Abbruch)

 Löse $H_{j+1,j} y_j = \beta e_1$

$$x_j = x_0 + V_j y_j$$

 Konvergenztest

end for

Bemerkung 3.19. 1. Bis auf die Minimalitätseigenschaft von $\|r_j\|_2$ stimmen die Algorithmen 3.6 und 3.7 weitgehend überein. D.h. sie haben auch ähnliche Nachteile.

2. Algorithmus 3.7 kann auch als Projektionsmethode mit $K = K_m(A, r_0)$ und $L = AK$ hergeleitet werden. (2 Zeiler, wenn man die richtige Formel sieht)

Frage: wie können wir den Algorithmus 3.7 praxistauglich machen. D.h.: Was ist ein geeigneter Konvergenztest und wie lösen wir das lineare Ausgleichsproblem?

Beobachte: (3.7) sagt: $\|r_j\|_2$ ist genau der Fehler des linearen Ausgleichsproblems. Diesen Fehler können wir explizit berechnen!

Analog zum Beweis von Satz 2.11:

$$\text{Sei } H_{j+1,j} = Q_j R_j = \underbrace{Q_j}_{\in \mathbb{R}^{j+1, j+1}} \begin{bmatrix} R_j \\ 0 \\ \underbrace{\quad}_{\in \mathbb{R}} \end{bmatrix} \text{ eine QR-Zerlegung.}$$

Sei

$$Q^t \beta e_1 = \begin{bmatrix} \underbrace{\quad}_{\in \mathbb{R}^j} \\ b_1 \\ b_2 \\ \underbrace{\quad}_{\in \mathbb{R}} \end{bmatrix}$$

$$\begin{aligned}
 \|r_j\|_2^2 &\stackrel{(3.7)}{=} \|\beta e_1 - H_{j+1,j} y_j\|_2^2 = \|Q^t(\beta e_1 - H_{j+1,j} y_j)\|_2^2 = (\star) \\
 (\star) &= \left\| \begin{pmatrix} b_1 - \tilde{R}_j y_j \\ b_2 \end{pmatrix} \right\|_2^2 \\
 &= \underbrace{\|b_1 - \tilde{R}_j y_j\|_2^2}_{=0 \text{ falls } H_{j+1,j} \text{ bzw. } \tilde{R}_j \text{ vollen Rang hat}} + \underbrace{\|b_2\|_2^2}_{=|b|^2} \\
 &\implies \|r_j\|_2 = |b_2|
 \end{aligned}$$

Wir berechnen b_2 als Nebenprodukt beim Lösen des linearen Ausgleichsproblems.

Aber: In jeder Iteration eine QR-Zerlegung zu berechnen ist teuer. Wie geht es besser?

Beobachte:

- $H_{j+1,j}$ hat Hessenbergstruktur, d.h. $H_{j+1,j}$ ist eine rechte obere Dreiecksmatrix, wo zusätzlich die untere Diagonale nicht notwendigerweise 0 ist. Dafür kann mit j Givensrotationen eine QR-Zerlegung berechnet werden.
- Beim Iterationsschritt $j \implies j+1$ werden lediglich eine neue Spalte und eine neue Zeile an $H_{j+1,j}$ drangehängt um $H_{j+2,j+1}$ zu erhalten. Die restlichen Einträge bleiben unverändert.

Sei $H_{j+1,j} = Q_j R_j$ eine QR-Zerlegung.

$$\begin{bmatrix} Q_j^t & 0 \\ 0 & 1 \end{bmatrix} H_{j+2,j+1} = \begin{bmatrix} Q_j^t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} H_{j+1,j} & \star \\ 0 & \star \end{bmatrix} =$$

Die letzte Matrix kann mittels einer einzigen Givensrotation in obere Dreiecksgestalt gebracht werden.

\implies Erweitere QR-Zerlegung in jedem Schritt.

— Ende von Vorlesung 08 am 08.11.2022 —

Algorithm 3.20 GMRES

Input: $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n, x_0 \in \mathbb{R}^n, m \in \mathbb{N}$

Output: $x_m \in x_0 + K_m(A, r_0), \|r_m\|_2 = \|b - Ax_m\|_2$ minimal

$$r_0 = b - Ax_0, \beta = \|r_0\|_2, v_1 = \frac{r_0}{\beta}$$

$$\hat{b} = \beta e_1$$

for $j = 1, m$ **do**

Bestimme $V_{j+1}, H_{j+1,j}$ mit Arnoldi-Verfahren \triangleright Durch Anhängen von Spalten/ Zeilen, Stop beim Abbruch

Wende $G_{12}, G_{23}, \dots, G_{j-1,j}$ auf die letzte Spalte von $H_{j+1,j}$ an

$$\text{Bestimme } G_{j,j+1} \text{ so, dass } H_{j+1,j} = G_{j,j+1} H_{j+1,j} = \begin{bmatrix} \tilde{R}_{j+1} \\ 0 \end{bmatrix}$$

$$\hat{b} = G_{j,j+1} \hat{b}$$

if \hat{b} klein **then**

$$\text{Löse } \tilde{R}_j y_j = [\hat{b}_i]_{i=1, \dots, j}$$

$$\text{Gebe } x_j = x_0 + V_j y_j \text{ zurück}$$

end if

end for

Berechne x_m wie oben, beginne von vorne mit $x_0 = x_m$

Bemerkung 3.21. 1. Außer, dass GMRES für invertierte Matrizen für $m = n$ konvergiert, ist bis heute wenig über Konvergenzaussagen bekannt.

2. Außer der Matrix-Vektor-Multiplikation im Arnoldi-Verfahren sind an der j -ten Iteration nur Vektor/Matrizen der Größe j , bzw. $j \times j$ beteiligt. Der Lösungsvektor wird erst nach erfülltem Konvergenzkriterium zusammengesetzt.
3. Für $m \ll n$ sprechen wir von einem **Restarted-GMRES**. Oft ist z.B. $m = 20$ ausreichend.
4. Das Verfahren der vollständigen Orthogonalisierung kann analog abgeändert werden.
5. Falls das Arnoldi-Verfahren abbricht, ist die Näherungslösung exakt ("Lucky Breakdown", Übung).
6. Das GMRES-Verfahren ist heutzutage (2022) eines der beliebtesten Verfahren zum Lösen LGS ohne weitere, besondere Eigenschaften.

3.8 Der symmetrische Lanczos-Prozess

Frage: Können wir ein besseres Verfahren herleiten, wenn wir zusätzliche Annahmen zu unserer Matrix treffen? z.B. Symmetrie oder spd?

Beobachte: Für A symmetrisch ist

$$H_{m,m} \stackrel{(3.6)}{=} V_m^t A V_m$$

symmetrisch und hat Hessenberg-Struktur

$$H_{m,m} = \begin{bmatrix} \alpha_1 & \beta_1 & & & & \\ \beta_1 & \alpha_2 & & & & \\ & \beta_2 & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & \beta_{n-1} & \alpha_{n-1} & \beta_{n-1} \\ & & & & \beta_n & \alpha_n \end{bmatrix}$$

Die vom Arnoldi-Verfahren generierte Matrix $T_m = H_{m,m}$ ist tridiagonal und symmetrisch. **Idee:**

- Die null-Einträge $h_{ij} = \langle v_j, v_i \rangle, i = 1, \dots, j-2$ müssen gar nicht erst berechnet werden
- Wir können die Symmetrie im Algorithmus explizit ausnutzen.

Algorithm 3.22 Lanczos-Verfahren

Input: $A \in \mathbb{R}^{n \times n}$ symmetrisch, $v \in \mathbb{R}^n, x_0 \in \mathbb{R}^n, m \in \mathbb{N}$

Output: $V_m = [v_1 \dots v_m]$ von $K_m(A, v), v_{m+1} \in \mathbb{R}^n, \beta \in \mathbb{R}, T_m \in \mathbb{R}^{m \times m}$

$\beta_1 = 0, v_0 = 0$

for $j = 1, m$ **do**

$h_{ij} = 0, i = 1, \dots, j-2$

$h_{j-1,j} = \beta_j$

$w_j = z - \sum_{i=1}^{j-1} h_{ij} v_i = Av_j - \beta_j v_{j-1}$

▷ In der Praxis hat w keinen Index

$\alpha_j = \langle w_j, v_j \rangle_2 = h_{jj}$

$w_j = w_j - \alpha_j v_j$

▷ $w = Av_j - \alpha_j v_j - \beta_j v_{j-1}$

$\beta_{j+1} = \|w_j\|_2$

if $\beta_{j+1} = 0$ **then**

▷ in der Praxis: testen ob Betrag klein

Stop

end if $v_{j+1} = \frac{w_j}{\beta_{j+1}}$

end for

Berechne x_m wie oben, beginne von vorne mit $x_0 = x_m$

Bemerkung 3.23. Das Lanczos-Verfahren implementiert die Berechnung der ONB als Drei-Term-Rekursion, die für höhere Iterationszahlen instabil werden kann.

Idee: Wende unser neues Verfahren auf die Methode der vollständigen Orthogonalisierung an.

Algorithm 3.24 Lanczos-Verfahren für lineare Gleichungssysteme

Input: $A \in \mathbb{R}^{n \times n}$ symmetrisch, $b \in \mathbb{R}^n, x_0 \in \mathbb{R}^n, m \in \mathbb{N}$

Output: $x_m \in x_0 + K_m(A, r_0), x_m \approx A^{-1}b$

$$r_0 = b - Ax_0, \beta = \|r_0\|_2, v_1 = \frac{r_0}{\beta}$$

for $j = 1, m$ **do**

 Bestimme V_j, T_j mittel Lanczos-Verfahren (Stop bei Abbruch)

 Löse $T_j y_j = \beta e_1$

$x_0 = V_j y_j$ Konvergenztest

end for

Berechne x_m wie oben, beginne von vorne mit $x_0 = x_m$

Bemerkung 3.25. 1. Lemma 3.17, d.h.

$$\|r_j\|_2 = \|b - Ax_j\|_2 = \beta_{j+1} |e_j^t y_j|$$

gilt weiterhin. Wir können den Konvergenztest also ohne Berechnung von x_j durchführen.

2. Da T_j tridiagonal ist, kann $T_j y_j = \beta e_1$ in $O(j)$ gelöst werden.

3. Speicherbedarf und Rechenaufwand wachsen immer noch mit j .

—Ende von Vorlesung 09 am 10.11.2022—