

**EINFÜHRUNG IN DIE
NUMERISCHE MATHEMATIK II ¹**

Numerische Lineare Algebra

Prof. Dr. Hans Babovsky

Institut für Mathematik

Technische Universität Ilmenau

¹Version vom Sommer 2010

Inhaltsverzeichnis

1	Kondition linearer Gleichungssysteme	2
1.1	Matrix- und Vektornormen	2
1.2	Fehlerabschätzungen und Kondition	4
2	Direkte Lösungsverfahren	6
2.1	Gestaffelte Systeme	6
2.2	Gauß-Elimination (ohne Pivotisierung)	8
2.3	Spaltenpivotisierung	12
2.4	Das Cholesky-Verfahren	16
3	Iterationsverfahren	19
3.1	Lineare Gleichungssysteme als Fixpunktprobleme	19
3.2	Die wichtigsten Fixpunkt-Iterationsverfahren	22
3.3	Relaxationsverfahren	27
4	Gradientenverfahren	33
4.1	Lineare Gleichungssysteme als Extremwertprobleme	33
4.2	Das Verfahren der konjugierten Gradienten (cg-Verfahren)	36
5	QR-Zerlegungen	40
5.1	Zur Existenz und Eindeutigkeit von QR -Zerlegungen	40
5.2	Givens-Rotationen	42
5.3	Householder-Reflexionen	46
6	Eigenwert- und Ausgleichsprobleme	48
6.1	Vektoriterationen	48
6.2	Ein QR -Algorithmus für symmetrische EW-Probleme	50
6.3	Lineare Ausgleichsprobleme	54
6.4	Singularwertzerlegungen und Pseudoinverse	59

1 Kondition linearer Gleichungssysteme

Ein wichtiger Aspekt bei der numerischen Lösung eines Problems ist die *Kondition* des Problems, welche Auskunft darüber gibt, welchen Einfluss Rundungsfehler auf das Ergebnis haben. Dies wollen wir zunächst für lineare Gleichungssysteme untersuchen. Zur Lösung des Gleichungssystems $Ax = B$ müssen die Koeffizienten a_{ij} und b_i als Eingabeparameter vorgegeben sein. Diese Parameter sind in häufig nicht genau darstellbar – sei es, dass sie nur ungenau bekannt sind, oder dass sie als reelle Zahlen einem Rundungsfehler unterliegen. Den Einfluss dieser Fehler auf die Lösung wollen wir untersuchen.

(1.1) Beispiel: Die exakte Lösung des Gleichungssystems $Ax = b$ mit

$$A = \begin{pmatrix} 2.09 & 11.32 \\ 8.84 & 47.83 \end{pmatrix}, \quad b = \begin{pmatrix} 2.96 \\ 12.47 \end{pmatrix}$$

ist $x = (-4, 1)^T$. Wir definieren die gerundeten Matrizen

$$\hat{A} = \begin{pmatrix} 2.1 & 11.3 \\ 8.8 & 47.8 \end{pmatrix}, \quad \hat{b} = \begin{pmatrix} 3.0 \\ 12.5 \end{pmatrix}$$

Die Lösung von $Ax = \hat{b}$ ist $x = (-19.116, 3.794)^T$, die von $\hat{A}x = b$ ist $x = (0.614, 0.148)^T$ und die von $\hat{A}x = \hat{b}$ gleich $x = (2.287, -0.16)^T$. Ähnliche Abhängigkeiten von Rundungsfehlern treten z.B. bei der Matrix

$$\tilde{A} = \begin{pmatrix} 2.1 & 11.3 \\ -8.8 & 47.8 \end{pmatrix}$$

nicht auf. Versuchen Sie eine geometrische Begründung!

1.1 Matrix- und Vektornormen

(1.1) Definition: (a) Eine **Vektornorm** auf \mathbb{R}^n ist eine Abbildung $\|\cdot\|_V : \mathbb{R}^n \rightarrow \mathbb{R}_+$, welche die Vektor-Normeigenschaften erfüllt:

- (i) $\|x\|_V > 0$, falls $x \neq 0$,
- (ii) $\|\lambda x\| = |\lambda| \|x\|$ für alle $\lambda \in \mathbb{R}$, $x \in \mathbb{R}^n$,
- (iii) $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in \mathbb{R}^n$ (Dreiecksungleichung).

(b) Eine **Matrixnorm** auf $\mathbb{R}^n \times \mathbb{R}^n$ ist eine Abbildung $\|\cdot\|_M : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}_+$, welche die Matrix-Normeigenschaften erfüllt:

- (i) $\|A\|_M > 0$, falls $A \neq 0$,
- (ii) $\|\lambda A\|_M = |\lambda| \|A\|_M$ für alle $\lambda \in \mathbb{R}$, $A \in \mathbb{R}^{n \times n}$,
- (iii) $\|A + B\| \leq \|A\| + \|B\|$ für alle $A, B \in \mathbb{R}^{n \times n}$ (Dreiecksungleichung).
- (iv) $\|A \cdot B\|_M \leq \|A\|_M \cdot \|B\|_M$ für alle $A, B \in \mathbb{R}^{n \times n}$.

(c) Ein Normpaar $(\|\cdot\|_M, \|\cdot\|_V)$ heißt **kompatibel**, wenn für beliebige $x \in \mathbb{R}^n$ und $A \in \mathbb{R}^{n \times n}$ gilt: $\|Ax\|_V \leq \|A\|_M \|x\|_V$.

(1.2) Beispiele: (a) Häufig verwendete Vektornormen sind

$$\begin{aligned} \|x\|_\infty &:= \max_{k=1..n} |x_k| && \text{(Maximumnorm)}, \\ \|x\|_2 &:= \left(\sum_{i=1}^n x_i^2 \right)^{1/2} && \text{(Euklidische Norm)}, \\ \|x\|_1 &:= \sum_{i=1}^n |x_i| && \text{(L}_1\text{-Norm)}. \end{aligned}$$

(b) Häufig verwendete Matrixnormen sind

$$\begin{aligned} \|A\|_Z &:= \max_{i=1..n} \sum_{j=1}^n |a_{ij}| && \text{(Zeilensummennorm)}, \\ \|A\|_S &:= \max_{j=1..n} \sum_{i=1}^n |a_{ij}| && \text{(Spaltensummennorm)}, \\ \|A\|_F &:= \left(\sum_{i,j=1}^n a_{ij}^2 \right)^{1/2} && \text{(Frobenius-Norm)}. \end{aligned}$$

(c) Kompatible Normpaare sind z.B. $(\|\cdot\|_Z, \|\cdot\|_\infty)$, $(\|\cdot\|_S, \|\cdot\|_1)$ und $(\|\cdot\|_F, \|\cdot\|_2)$.

Zu einer vorgegebenen Vektornorm $\|\cdot\|_V$ kann stets eine kompatible Matrixnorm $\|\cdot\|_M$ konstruiert werden, wie das folgende Lemma zeigt.

(1.3) Lemma: Sei $\|\cdot\|_V$ eine Vektornorm. Dann ist durch

$$\|A\|_M := \max_{x \neq 0} \frac{\|Ax\|_V}{\|x\|_V} = \max_{\|x\|=1} \|Ax\|_V.$$

eine Matrixnorm definiert (die **durch $\|\cdot\|_V$ induzierte Norm**. Das Matrixpaar $(\|\cdot\|_M, \|\cdot\|_V)$ ist kompatibel.

Beweis: Normeigenschaften: (i) Ist $A \neq 0$, so gibt es ein $x \in \mathbb{R} \setminus \{0\}$ mit $Ax \neq 0$. Hieraus folgt $\|A\|_M > 0$.

(ii) und (iii) folgen aus den entsprechenden Eigenschaften für Vektornormen.

(iv) Ist $B = 0$, so ist auch $AB = 0$ und die Behauptung ist erfüllt. Andernfalls ist

$$\begin{aligned}\|AB\|_M &= \max_{x \neq 0} \frac{\|ABx\|_V}{\|x\|_V} = \max_{x: Bx \neq 0} \frac{\|ABx\|_V}{\|x\|_V} = \max_{x: Bx \neq 0} \left(\frac{\|ABx\|_V}{\|Bx\|_V} \frac{\|Bx\|_V}{\|x\|_V} \right) \\ &= \max_{Bx \neq 0} \frac{\|ABx\|_V}{\|Bx\|_V} \cdot \max_{Bx \neq 0} \frac{\|Bx\|_V}{\|x\|_V} \leq \|A\|_M \cdot \|B\|_M\end{aligned}$$

Kompatibilität: folgt unmittelbar aus der Definition von $\|\cdot\|_M$. \circ

Ist $A \in \mathbb{R}^{n \times n}$, so heißt

$$\rho(A) := \max\{|\lambda| : \lambda \text{ ist Eigenwert von } A\}$$

der **Spektralradius von A** . Dieser kann zur Berechnung einer Bestimmten Matrixnorm herangezogen werden. Aus der linearen Algebra ist bekannt:

(1.4) Lemma: Sei $\|\cdot\|_2$ die durch die Vektornorm $\|\cdot\|_2$ induzierte Matrixnorm. Dann ist für beliebige $A \in \mathbb{R}^{n \times n}$

$$\|A\|_2 = \sqrt{\rho(A^T A)}$$

Für symmetrische Matrizen A gilt

$$\|A\|_2 = \rho(A)$$

(1.5) Bemerkung: Ohne dies immer zu betonen, benutzen wir im Folgenden immer nur kompatible Normpaare, d.h. wir wenden die Ungleichung $\|Ax\|_V \leq \|A\|_M \|x\|_V$ an, ohne dies immer neu zu rechtfertigen. Des weiteren verzichten wir häufig auf die Indizes “V” und “M”. Aus dem Zusammenhang geht in der Regel schnell hervor, ob eine Norm eine Vektor- oder eine Matrixnorm ist.

1.2 Fehlerabschätzungen und Kondition

Im Folgenden seien x Lösung von $Ax = b$ und \tilde{x} Lösung des gestörten Systems $Ax = \tilde{b}$ mit $\tilde{b} = b + \Delta b$. $\|\cdot\|_M$ und $\|\cdot\|_V$ seien kompatible Normen. Gesucht ist eine Abschätzung des Fehlers

$$\Delta x := \tilde{x} - x.$$

Offenbar erfüllt Δx die Gleichung

$$A \cdot \Delta x = \Delta b.$$

Aus den beiden Abschätzungen

$$\|b\|_V = \|Ax\|_V \leq \|A\|_M \cdot \|x\|_V \quad \text{und} \quad \|\Delta x\|_V = \|A^{-1}\Delta b\|_V \leq \|A^{-1}\|_M \cdot \|\Delta b\|_V$$

folgt als Beziehung zwischen dem relativen Fehler der Lösung und dem der rechten Seite die Abschätzung

$$\frac{\|\Delta x\|_V}{\|x\|_V} \leq \kappa(A) \cdot \frac{\|\Delta b\|_V}{\|b\|_V}$$

wobei $\kappa(A)$ die **Konditionszahl von A** ist, definiert durch

$$\kappa(A) := \|A^{-1}\|_M \cdot \|A\|_M.$$

Ähnliche Überlegungen können durchgeführt werden, wenn zusätzlich zur rechten Seite auch die **Matrix A gestört** ist. Es seien x die exakte Lösung von $Ax = b$ und \tilde{x} die Lösung von $(A + \Delta A)\tilde{x} = b + \Delta b$, sowie $\Delta x = \tilde{x} - x$. Dann ist

$$A\Delta x = \Delta b - \Delta A \cdot x - \Delta A \cdot \Delta x$$

und damit

$$\|\Delta x\| \leq \|A^{-1}\| \cdot (\|\Delta b\| + \|\Delta A\|\|x\| + \|\Delta A\|\|\Delta x\|),$$

also

$$(1 - \|A^{-1}\| \cdot \|\Delta A\|) \cdot \|\Delta x\| \leq \|A^{-1}\| \cdot (\|\Delta b\| + \|\Delta A\| \cdot \|x\|).$$

Ist die Störung ΔA so klein, dass $\|A^{-1}\| \cdot \|\Delta A\| < 1$, so folgt

$$\|\Delta x\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\Delta A\|} \cdot (\|\Delta b\| + \|\Delta A\|\|x\|).$$

Für den relativen Fehler $\|\Delta x\|/\|x\|$ gilt daher wegen $\|x\| \geq \|b\|/\|A\|$ die Abschätzung

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\|A^{-1}\|\|A\|}{1 - \|A^{-1}\|\|\Delta A\|} \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right).$$

Mit $1 > \|A^{-1}\|\|\Delta A\| = \kappa(A)\|\Delta A\|/\|A\|$ folgt

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A) \frac{\|\Delta A\|}{\|A\|}} \cdot \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right).$$

Aus diesen Überlegungen lässt sich die folgende “**Daumenregel**” herleiten: Bei einem Rechner mit d -stelliger Genauigkeit (d.h. $\|\Delta A\|/\|A\| \approx 5 \cdot 10^{-d}$ und $\|\Delta b\|/\|b\| \approx 5 \cdot 10^{-d}$) kann man bei einer Matrix mit Konditionszahl $\kappa(A) \approx 10^\alpha$ eine Lösung erwarten, welche auf $d - \alpha - 1$ Dezimalstellen genau ist (bezogen auf den größten Wert!).

(1.6) Beispiel: Die beiden Inversen zu Beispiel (1.0.1) sind

$$A^{-1} = \begin{pmatrix} -459.462 & 108.742 \\ 84.918 & -20.077 \end{pmatrix}, \quad \tilde{A}^{-1} = \begin{pmatrix} 0.239 & -0.057 \\ 0.044 & 0.01 \end{pmatrix}.$$

Damit erhalten wir die Konditionszahlen bzgl. der Norm $\|\cdot\|_Z$

$$\begin{aligned} \kappa_Z(A) &= 56.67 \cdot 568.204 \approx 32200, \\ \kappa_Z(\tilde{A}) &= 56.67 \cdot 0.296 \approx 16.8. \end{aligned}$$

Das zweite Gleichungssystem ist also weniger empfindlich gegenüber Störungen als das erste.

2 Direkte Lösungsverfahren

2.1 Gestaffelte Systeme

Wir wenden uns nun der Frage zu, wie lineare Gleichungssysteme numerisch – insbesondere auch durch Computerprogrammierung – gelöst werden können.

Besonders einfach ist das Gleichungssystem $Ax = b$ zu lösen, wenn A eine Dreiecksgestalt hat. (Solche Systeme heißen *gestaffelte Systeme*.)

(2.1) Definition: Eine Matrix A ist eine **rechte (obere) Dreiecksmatrix**, falls A die Form hat

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ 0 & a_{22} & \vdots \\ \vdots & \ddots & \ddots \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix},$$

wenn also gilt $a_{ij} = 0$ für $i > j$. Ist dagegen A von der Form

$$A = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ \vdots & a_{22} & \ddots & \vdots \\ & & \ddots & 0 \\ a_{n1} & \cdots & & a_{nn} \end{pmatrix}$$

(d.h. $a_{ij} = 0$ falls $i < j$), so heißt A **linke (untere) Dreiecksmatrix**.

Ein Gleichungssystem $Ax = b$ heißt **gestaffeltes System**, wenn A (linke oder rechte) Dreiecksmatrix ist.

(2.2) Bemerkung: Ist A obere oder untere Dreiecksmatrix, so ist die Determinante

$$\det(A) = \prod_{i=1}^n a_{ii}.$$

Da A als regulär vorausgesetzt ist, folgt

$$a_{ii} \neq 0, \quad i = 1, \dots, n.$$

Ist A obere Dreiecksmatrix, so lässt sich das Gleichungssystem $A \cdot x = b$ leicht durch *Rückwärtseinsetzen* lösen. Aus der letzten der n Gleichungen folgt nämlich $x_n = b_n/a_{nn}$, aus der vorletzten $x_{n-1} = (b_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1}$, etc. Der folgende Algorithmus löst das Gleichungssystem.

(2.3) Algorithmus (*Rückwärtseinsetzen*) zur Lösung eines linearen Gleichungssystems in oberer Dreiecksgestalt.

S1: Berechne $x_n := b_n/a_{nn}$;

S2: Für $k = n - 1$ bis 1 berechne

$$x_k := (b_k - a_{k,k+1}x_{k+1} - \cdots - a_{k,n}x_n)/a_{kk}.$$

Entsprechend lassen sich Gleichungen mit unteren Dreiecksmatrizen durch *Vorwärtseinsetzen* gelöst, bei denen zunächst $x_1 = a_{11}/b_1$ und anschließend “von oben nach unten” die anderen Koeffizienten bestimmt werden.

Als *erweiterte Matrix* eines Gleichungssystems $Ax = b$ bezeichnen wir im Folgenden die

$n \times (n + 1)$ -Matrix $(A|b)$.

Ist von einer Matrix A eine Darstellung der Form

$$A = L \cdot R \quad (2.1)$$

mit einer linken Dreiecksmatrix L und einer rechten Dreiecksmatrix R bekannt, so kann das Gleichungssystem $Ax = b$ durch Lösung zweier gestaffelter Systeme lösen. Zunächst löst man das System $Ly = b$ durch Vorwärtseinsetzen, anschließend das System $Rx = y$ durch Rückwärtseinsetzen.

2.2 Gauß-Elimination (ohne Pivotisierung)

Eine Möglichkeit, das Gleichungssystem $Ax = b$ in ein gestaffeltes System überzuführen, bietet das *Gaußsche Eliminationsverfahren*. Hierbei werden nacheinander alle Elemente von A unterhalb der Diagonale durch elementare Zeilenoperationen eliminiert. Die zentralen Grundlagen für diesen sind in der folgenden Bemerkung zusammengefasst. Hierbei bezeichnen wir die i -te Zeile der erweiterten Matrix $(A|b)$ mit z_i , d.h. wir schreiben

$$(A|b) = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}.$$

(2.4) Bemerkungen: (a) Die Lösung des Gleichungssystem wird durch folgende elementare Zeilenoperation

$$\begin{pmatrix} z_1 \\ \vdots \\ z_k \\ \vdots \\ z_n \end{pmatrix} \rightarrow \begin{pmatrix} z_1 \\ \vdots \\ z_k - \lambda z_\ell \\ \vdots \\ z_n \end{pmatrix},$$

bei der die Zeile z_k durch $z_k - \lambda z_\ell$ ($\lambda \in \mathbb{R}$, $k \neq \ell$) ersetzt wird, nicht verändert.

(b) Wird in (a) $\lambda = a_{k,\ell}/a_{\ell,\ell}$ gewählt, so erhält $a_{k,\ell}$ nach dieser Operation den Wert 0 (“wird eliminiert”).

(c) Die Operation (a) kann durch die Matrixmultiplikation

$$(A|b) \rightarrow L \cdot (A|b) \quad (2.2)$$

mit der $n \times n$ -Matrix $L = (\ell_{ij})$,

$$\ell_{ij} = \begin{cases} 1 & \text{falls } i = j \\ -\lambda & \text{falls } i = k \text{ und } j = \ell \\ 0 & \text{sonst} \end{cases}$$

durchgeführt werden.

Der Gauß-Algorithmus kann nun wie folgt dargestellt werden.

(2.5) Algorithmus (Gauß-Elimination ohne Pivotisierung): Im folgenden bezeichne a_{ij} die Elemente und z_j die j -te Zeile der gerade aktuellen erweiterten Matrix.

(S1) Für $i=1(1)n-1$:

(S2) Für $j=i+1(1)n$:

berechne $l_{ji} := a_{ji}/a_{ii}$;

setze $z_j := z_j - l_{ji} \cdot z_i$.

(2.6) Beispiel:

$$(A|b) = \left(\begin{array}{ccc|c} 1 & 3 & -2 & 1 \\ 2 & 2 & 1 & 1 \\ -3 & -1 & 1 & 1 \end{array} \right).$$

Es ist $l_{21} = a_{21}/a_{11} = 2$ und $l_{31} = a_{31}/a_{11} = -3$. Die Operationen $z_j := z_j - l_{j1} \cdot z_1$, $j = 2, 3$, führen auf das System

$$(A^{(1)}|b^{(1)}) = \left(\begin{array}{ccc|c} 1 & 3 & -2 & 1 \\ 0 & -4 & 5 & -1 \\ 0 & 8 & -5 & 4 \end{array} \right).$$

Es folgt $l_{32} = a_{32}/a_{22} = -2$; das System in Dreiecksform lautet

$$(A^{(2)}|b^{(2)}) = \left(\begin{array}{ccc|c} 1 & 3 & -2 & 1 \\ 0 & -4 & 5 & -1 \\ 0 & 0 & 5 & 2 \end{array} \right).$$

Man überlegt sich nun leicht, dass die Schleife (S1) für $i = 1$ auf das Ergebnis $A^{(1)} = L^{(1)} \cdot A$ führt mit

$$L^{(1)} = \begin{pmatrix} 1 & & & & \\ & -l_{21} & 1 & & \\ & \vdots & \ddots & \ddots & \\ & -l_{n-1,1} & & 1 & \\ & -l_{n1} & & & 1 \end{pmatrix} .$$

Entsprechend erhält man für $i = 2, \dots, n$ nacheinander die Matrizen $A^{(i)} = L^{(i)} \cdot A^{(i-1)}$, wobei $L^{(i)}$ diejenige Matrix ist, welche aus der Einheitsmatrix entsteht, wenn die i -te Spalte unterhalb der Diagonale durch die Elemente $-l_{i+1,i}, \dots, -l_{n,i}$ "aufgefüllt" wird. Bezeichnen wir die rechte Dreiecksmatrix, welche als Ergebnis des Eliminationsverfahrens entsteht, mit R , so folgt

$$R = \tilde{L} \cdot A \quad (2.3)$$

mit

$$\tilde{L} = \underbrace{\begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & -l_{n,n-1} & 1 \end{pmatrix}}_{L^{(n-1)}} \cdots \underbrace{\begin{pmatrix} 1 & & & & \\ & -l_{21} & 1 & & \\ & \vdots & \ddots & \ddots & \\ & -l_{n-1,1} & & 1 & \\ & -l_{n1} & & & 1 \end{pmatrix}}_{L^{(1)}} .$$

Wichtige Eigenschaften von $L^{(j)}$ und \tilde{L} zeigt der folgende Hilfssatz, den wir ohne Beweis angeben. (Die Beweise sind elementar.)

(2.7) Lemma: (a) $(L^{(j)})^{-1}$ entsteht aus $L^{(j)}$ durch Streichen der Minus-Zeichen vor l_{kj} .

(b) Die Inverse von \tilde{L} ist

$$(\tilde{L})^{-1} =: L = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ \vdots & l_{32} & \ddots & & \\ l_{n-1,1} & \vdots & \ddots & 1 & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{pmatrix}$$

Aus Gleichung (2.3) und Lemma (2.8)(b) folgt nun leicht das wichtige Ergebnis

(2.8) Satz: $R := A^{(n-1)}$ sei das Ergebnis der Gauß-Elimination. L sei die linke Dreiecksmatrix, welche aus der Einheitsmatrix entsteht durch Hinzufügen der Elemente l_{ij} ($i > j$) aus Algorithmus (2.6). Dann hat A die LR -Zerlegung

$$A = L \cdot R.$$

Damit kann das Gaußsche Eliminationsverfahren interpretiert werden als Verfahren zur Zerlegung von A in ein Produkt einer unteren Dreiecksmatrix L mit einer oberen Dreiecksmatrix R (**LR-Zerlegung**²).

(2.7) Beispiel (*Fortsetzung*): Die Matrix A hat die LR -Zerlegung

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 3 & -2 \\ 0 & -4 & 5 \\ 0 & 0 & 5 \end{pmatrix}$$

Ist eine LR -Zerlegung von A bekannt, so kann das Gleichungssystem $A \cdot x = b$ auf die Lösung zweier gestaffelter Systeme zurückgeführt werden, wie am Ende des Abschnitts 2.1 gezeigt wurde. Dies führt auf die folgende Modifikation der Gauß-Elimination.

(2.9) Algorithmus (Gauß-Elimination) zur Berechnung von $A \cdot x = b$:

- S1: Konstruiere die LR -Zerlegung von A mit Hilfe des Algorithmus (2.6) (ohne rechte Seite);
- S2: Löse $L \cdot y = b$ durch Vorwärtseinsetzen;
- S3: Löse $R \cdot x = y$ durch Rückwärtseinsetzen.

Insbesondere zur Lösung großer Gleichungssysteme ist es wichtig zu wissen, wie der

²*links-rechts-Zerlegung*; wird in englischsprachiger Literatur gewöhnlich als LU (d.h. *lower-upper*) - Zerlegung bezeichnet.

Rechenaufwand mit der Größe der Dimension wächst. Als Kennzahl für den numerischen Aufwand benutzen wir hier die Anzahl der Multiplikationen.

(2.10) Rechenaufwand: Für große n werden für

$$\begin{array}{ll} \text{den Schritt } S1 & \text{ca. } n^3/3 \\ \text{jeden der Schritte } S2 \text{ und } S3 & \text{ca. } n^2/2 \text{ Multiplikationen benötigt.} \end{array}$$

Der Hauptaufwand liegt damit in $S1$; dieser Schritt muss aber bei mehrmaliger Lösung (insbesondere bei der Bestimmung von A^{-1}) nur einmal durchgeführt werden. Soll also das Gleichungssystem $Ax = b$ nur einmal gelöst werden, so kann hierfür der Algorithmus (2.6) verwendet werden, während bei mehrmaliger Lösung (d.h. bei der Lösung mit verschiedenen rechten Seiten) die Variante (2.10) verwendet werden sollte.

(2.11) Bemerkung: Die j -te Spalte von A^{-1} ist gegeben als Lösung des Gleichungssystems $Ax = e_j$, wobei e_j der j -te kanonische Einheitsvektor ist. Zur vollständigen Bestimmung von A^{-1} sind also n Gleichungssysteme der Form $Ax = b$ zu lösen, wobei die LR -Zerlegung nur einmal durchgeführt werden muss. Der Gesamt-Rechenaufwand ist von der Ordnung $\mathcal{O}(n^3)$. Nach (2.11) sind $4n^3/3$ Multiplikationen erforderlich.

2.3 Spaltenpivotisierung

Die bisher eingeführte Gauß-Elimination ist nicht immer durchführbar. Beispielsweise kann das Diagonalelement der aktuell bearbeiteten Spalte gleich 0 sein. Einen weiteren Effekt zeigt das folgende Beispiel.

(2.12) Beispiel: Das Gleichungssystem

$$(A|b) := \left(\begin{array}{ccc|c} 0.289 & 0.312 & -0.445 & 1 \\ -0.652 & -0.703 & 2.02 & 1 \\ 0.544 & -0.294 & 0.263 & 1 \end{array} \right)$$

soll auf einem Rechner mit dreistelliger Genauigkeit gelöst werden. Die exakte Lösung ist (mit dreistelliger Genauigkeit) gleich

$$x = \begin{pmatrix} 2.99 \\ 5.00 \\ 3.20 \end{pmatrix}.$$

Die Kondition von A bezüglich der Zeilensummennorm (diese ist die durch die Maximumnorm induzierte Matrixnorm, insbesondere also kompatibel zu $\|\cdot\|_\infty$)

$$\kappa_Z(A) = 24.5.$$

Ein relativer Fehler (bzgl. der Maximumnorm) von 10^{-2} der Eingabedaten sollte also auf einen relativen Fehler des Ergebnisses von höchstens 0.245 führen. Dies entspricht einem absoluten Fehler von $0.245 \cdot \|x\|_\infty = 1.225$.

Die numerische Gauß-Elimination mit dreistelliger Genauigkeit führt zunächst auf

$$(A^{(1)}|b^{(1)}) = \left(\begin{array}{ccc|c} 0.289 & 0.312 & -0.445 & 1 \\ 0 & 0.00212 & 1.01 & 3.26 \\ 0 & -0.881 & 1.10 & -0.88 \end{array} \right)$$

mit $\ell_{21} = -2.26$ und $\ell_{31} = 1.88$ und schließlich auf

$$(A^{(2)}|b^{(2)}) = \left(\begin{array}{ccc|c} 0.289 & 0.312 & -0.445 & 1 \\ 0 & 0.00212 & 1.01 & 3.26 \\ 0 & 0 & 421 & 1360 \end{array} \right)$$

mit $\ell_{32} = 416$. Die numerische Lösung dieses gestaffelten Systems lautet

$$x_{\text{num},1} = \begin{pmatrix} 9.60 \\ -1.08 \\ 3.23 \end{pmatrix}$$

und ist offensichtlich völlig unbrauchbar. Ein besseres Ergebnis erhalten wir, wenn wir nach dem ersten Schritt die zweite und dritte Zeile vertauschen. Dies führt auf die numerische Lösung

$$x_{\text{num},2} = \begin{pmatrix} 3.00 \\ 5.03 \\ 3.23 \end{pmatrix}.$$

Offenbar hat das kleine Diagonalelement $a_{22}^{(1)}$ im zweiten Schritt zum ‘Aufschaukeln’ von Rundungsfehlern geführt – das System wurde *numerisch instabil*.

Die Beispiele zeigen, dass es aus Gründen der Durchführbarkeit und der *numerischen Stabilität* (d.h. geringer Einfluß von Rundungsfehlern) nötig sein kann, vor Beginn der Elimination einer Spalte eine Zeilenvertauschung vorzunehmen. Aus Stabilitätsgründen

ist es dabei ratsam, vor der Elimination der j -ten Spalte die j -te Zeile mit derjenigen Zeile $j+k$ ($k \geq 0$) zu vertauschen, welche an der j -ten Stelle das betragsgrößte Element enthält. Dies führt zur Modifikation des Gaußschen Eliminationsalgorithmus [1.6]:

(2.13) Algorithmus (Gauß-Elimination mit Pivotisierung): Führe im Algorithmus [1.6] in Schleife (S1) vor Beginn der Schleife (S2) die folgenden Schritte durch:

- (i) Bestimme das Maximum max der Spaltenelemente $a_{i,i}, a_{i+1,i}, \dots, a_{n,i}$ und die Nummer $j_0 \geq i$ der Zeile, in der das Maximum auftritt;
- (ii) falls $j_0 > i$, vertausche die Zeilen (i) und (j_0).

Zeilenvertauschungen können durch linksseitige Multiplikation der Matrix mit einer *Permutationsmatrix* beschrieben werden.³ Beispielsweise ist mit

$$P := \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

PA diejenige Matrix, welche aus den Zeilen von A in der Reihenfolge (2) – (3) – (1) besteht. (Nachprüfen!)

(2.14) Satz: R sei die obere Dreiecksmatrix, welche als Ergebnis des Algorithmus (2.14) entsteht. Dann gibt es eine Permutationsmatrix P und eine untere Dreiecksmatrix L derart, dass $PA = LR$. Die Diagonalelemente von L sind 1. Die Elemente l_{ij} für $j > i$ sind bis auf Permutation die Elemente, welche bei den Zeilensubtraktionen berechnet werden.

Beweisskizze: Wir bringen zunächst durch Zeilenvertauschung das betragsgrößte Element der ersten Spalte in Diagonalposition und eliminieren die Elemente unterhalb der Diagonale. Das Ergebnis ist eine Matrix der Form $L^{(1)}P_1A$. Vor Elimination der j -ten Spalte führen wir zunächst eine Permutation P_j der Spalten j, \dots, n durch, um das betragsgrößte Element auf die Diagonale zu bringen. Das Ergebnis von Algorithmus (2.14) ist also eine Dreiecksmatrix R mit der Darstellung

$$R = L^{(n-1)}P_{n-1} \cdots L^{(2)}P_2L^{(1)}P_1A$$

³Eine Permutationsmatrix P ist eine $n \times n$ -Matrix, welche nur 0 und 1 als Koeffizienten hat, wobei in jeder Zeile und in jeder Spalte genau eine 1 steht.

Durch Einführung der neuen Permutationsmatrizen $\tilde{P}_{n-1} = P_{n-1}$ und $\tilde{P}_{n-j} = P_{n-1} \cdots P_{n-j+1}$ für $j > 1$ sowie $\tilde{L}^{(n-j-1)} := \tilde{P}_{n-j} L^{(n-j-1)} \tilde{P}_{n-j}^{-1}$ folgt die Darstellung

$$R = \tilde{L}^{(n-1)} \cdots \tilde{L}^{(2)} A$$

Da $L^{(j)}$ von der Form wie in Lemma (2.8) ist, gilt dies auch für $\tilde{L}^{(j)}$. (Begründung?) ○

Wir demonstrieren dieses Ergebnis an einem Beispiel.

(2.15) Beispiel: Für die Matrix

$$A = \begin{pmatrix} 0 & 4 & -2 \\ 3 & 1 & 1 \\ 6 & 0 & 1 \end{pmatrix}$$

soll eine Zerlegung der Form $PA = LR$ gefunden werden. Hierzu stellen wir eine erweiterte Matrix der Form

$$\left(\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 4 & -2 \\ 0 & 1 & 0 & 0 & 1 & 0 & 3 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 6 & 0 & 1 \end{array} \right)$$

auf. Am Ende soll die erste Matrix die Permutationsmatrix P , die zweite die linke Dreiecksmatrix L und die dritte die rechte Dreiecksmatrix R enthalten. Hierzu notieren wir in der ersten Matrix lediglich die Zeilenvertauschungen, in der zweiten die Koeffizienten ℓ_{ij} und in der dritten die sich aufbauende Dreiecksmatrix. Die Vertauschung der Zeilen 1 und 3 führt zunächst auf das System

$$\left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 1 & 1 & 0 & 0 & 6 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 3 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 4 & -2 \end{array} \right)$$

Anschließend werden die Hälfte der ersten Zeile von der zweiten subtrahiert

$$\left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 1 & 1 & 0 & 0 & 6 & 0 & 1 \\ 0 & 1 & 0 & 1/2 & 1 & 0 & 0 & 1 & 1/2 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 4 & -2 \end{array} \right)$$

und die zweite und dritte Zeile vertauscht. (Vorsicht: Von L werden nur die Elemente unterhalb der Hauptdiagonalen vertauscht!)

$$\left(\begin{array}{ccc|ccc} 0 & 0 & 1 & 1 & 0 & 0 & 6 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 4 & -2 \\ 0 & 1 & 0 & 1/2 & 0 & 1 & 0 & 1 & 1/2 \end{array} \right)$$

Schließlich wird das Element a_{32} eliminiert.

$$\left(\begin{array}{ccc|ccc} 0 & 0 & 1 & 1 & 0 & 0 & 6 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 4 & -2 \\ 0 & 1 & 0 & 1/2 & 1/4 & 1 & 0 & 0 & 1 \end{array} \right)$$

Damit erhalten wir

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/2 & 1/4 & 1 \end{pmatrix}, \quad R = \begin{pmatrix} 6 & 0 & 1 \\ 0 & 4 & -2 \\ 0 & 0 & 1 \end{pmatrix}.$$

2.4 Das Cholesky-Verfahren

Für eine wichtige Klasse von Matrizen lässt sich eine Zerlegung in Dreiecksmatrizen einfacher durchführen.

(2.16) Definition: (a) Eine $n \times n$ -Matrix A heißt **symmetrisch**, falls für alle $i, j = 1, \dots, n$ gilt: $a_{ij} = a_{ji}$.

(b) Eine symmetrische Matrix A heißt **positiv definit**, wenn für alle Vektoren $x \neq 0$ gilt

$$x^T A x > 0.$$

Es ist naheliegend, für die LR -Zerlegung symmetrischer Matrizen einen symmetrischen Ansatz der Form $A = GG^T$ mit einer linken Dreiecksmatrix

$$G = \begin{pmatrix} g_{11} & & & \\ \vdots & \ddots & & \\ g_{m1} & \cdots & g_{nn} & \end{pmatrix}$$

zu wählen. Wir untersuchen zunächst die prinzipielle Möglichkeit eines solchen Ansatzes.

(2.17) Lemma: Ist $A \in \mathbb{R}^{n \times n}$ positiv definit und ist $Q \in \mathbb{R}^{n \times n}$ regulär, so ist auch $B := Q^T A Q$ positiv definit.

Beweis: Sei $x \in \mathbb{R}^n \setminus \{0\}$. Wegen der Regularität von Q ist $Qx \neq 0$; wegen der Positiv-Definitheit von A ist

$$x^T B x = (Qx)^T A (Qx) < 0 \quad \circ$$

(2.18) Lemma: Für jede positiv definite Matrix $A = (a_{ij})$ gilt

- (a) A ist invertierbar.
- (b) $a_{ii} > 0$ für $i = 1, \dots, n$.
- (c) $\max_{ij} |a_{ij}| = \max_i a_{ii}$.

Beweis: Übung.

(2.19) Satz: A sei symmetrisch und positiv definit. Dann existieren eine linke Dreiecksmatrix $L = (l_{ij})$ und eine Diagonalmatrix $D = (d_{ij})$ mit $l_{ii} = 1$ und $d_{ii} > 0$, $i = 1, \dots, n$ derart, dass $A = LDL^T$.

Beweis: Wir schreiben A in der Form

$$A = \left(\begin{array}{c|c} a_{11} & z^T \\ \hline z & B^{(1)} \end{array} \right)$$

mit der symmetrischen positiv definiten $(n-1) \times (n-1)$ -Matrix $B^{(1)}$. Wegen Lemma (2.18)(b) ist $a_{11} > 0$. Wir können also die erste Spalte eliminieren ohne Pivotisierung:

$$L_1 A = \left(\begin{array}{c|c} a_{11} & z^T \\ \hline 0 & B^{(2)} \end{array} \right)$$

mit einer geeigneten Matrix

$$L_1 = \begin{pmatrix} 1 & & & \\ -\ell_{21} & 1 & & \\ \dots & & \dots & \\ -\ell_{n1} & 0 & \dots & 1 \end{pmatrix}$$

z^T bleibt bei dieser Operation unverändert. Die zentrale Beobachtung ist nun, dass auch z^T eliminiert wird, wenn wir $L_1 A$ von rechts mit L_1^T multiplizieren (Begründung?):

$$L_1 A L_1^T = \left(\begin{array}{c|c} a_{11} & 0 \\ \hline 0 & B^{(2)} \end{array} \right)$$

Wegen Lemma (2.17) ist diese Matrix positiv definit. Wir können nun iterativ die Matrix in eine Diagonalmatrix verwandeln. Nach Elimination der $(i-1)$ -ten Spalte und Zeile eliminieren wir wie früher die i -te Spalte durch Multiplikation von links mit einer geeigneten Matrix L_i . Anschließend wird die i -te Zeile eliminiert durch Multiplikation (von rechts) mit L_i^T . Das Ergebnis ist eine Diagonalmatrix D der Form

$$D = L_{n-1} \cdots L_1 A L_1^T \cdots L_{n-1}^T$$

Die gewünschte Form erhalten wir durch Multiplikation mit L (von links) bzw. mit L^T (von rechts), wobei

$$L = (L_{n-1} \cdots L_1)^{-1} \quad \circ$$

(2.20) Folgerung: Ist A positiv definit, so gibt es eine linke Dreiecksmatrix $G = (g_{ij})$ derart, dass $A = GG^T$.

Beweis: Die Diagonalmatrix D aus Satz (2.19) hat positive Diagonalelemente d_{ii} . Definieren wir $\sqrt{D} := \text{diag}(\sqrt{d_{ii}}, i = 1, \dots, n)$, so folgt $A = GG^T$ mit $G = L\sqrt{D}$, denn

$$GG^T = L\sqrt{D}\sqrt{D}L^T = LDL^T = A \quad \circ$$

Zur numerischen Berechnung von $G = (g_{ij})$ müssen wir nicht auf das Gaußsche Eliminationsverfahren zurückgreifen. Vielmehr liefert der Ansatz $A = GG^T$ bei Koeffizientenvergleich unmittelbar die Formeln zu ihrer Berechnung. Wir erhalten die folgenden

Gleichungen.

$$\begin{aligned} g_{11}^2 &= a_{11} \\ g_{21}g_{11} &= a_{21} & g_{21}^2 + g_{22}^2 &= a_{22} \\ g_{31}g_{11} &= a_{31} & g_{31}g_{21} + g_{32}g_{22} &= a_{32} & g_{31}^2 + g_{32}^2 + g_{33}^2 &= a_{33} \\ & & & & & \vdots \end{aligned}$$

In geschickter Reihenfolge angeordnet, können diese Formeln leicht zur Berechnung von G verwandt werden. Dies geschieht im

(2.21) Algorithmus (Cholesky-Zerlegung):

$$\begin{aligned} &\text{Für } i=1(1)n: \\ &\quad \text{Berechne } g_{ii} := \sqrt{a_{ii} - \sum_{j=1}^{i-1} g_{ij}^2} \\ &\quad \text{Für } j:=i+1(1)n \\ &\quad \quad \text{Berechne } g_{ji} := \left(a_{ji} - \sum_{k=1}^{i-1} g_{jk}g_{ik} \right) / g_{ii} \end{aligned}$$

(2.22) Rechenaufwand: Die numerische Durchführung des Cholesky-Verfahrens erfordert für große n ca. $n^3/6$ Multiplikationen und n Quadratwurzel-Berechnungen.

3 Iterationsverfahren

Die Lösung des Gleichungssystems $Ax = b$ besteht in der Ermittlung der n unbekannt Koeffizienten des Vektors x . Bei der Berechnung mittels LR -Zerlegung werden hierzu die n^2 unbekannt Koeffizienten der Matrizen L und R berechnet, was – wie oben gezeigt – für große n mit der Ordnung $\mathcal{O}(n^3)$ einen gewaltigen Rechenaufwand erfordert. Alternative Verfahren bestehen darin, bei fester Matrix nur die n Koeffizienten von x zu manipulieren. Diese werden nun vorgestellt.

3.1 Lineare Gleichungssysteme als Fixpunktprobleme

In diesem Abschnitt führen wir die Lösung des linearen Gleichungssystems $Ax = b$ zurück auf ein Fixpunktproblem.

Zur Erinnerung (vgl. Num. Math. I, Abschnitt 1.3):

- $x^* \in D$ (D abgeschlossene Teilmenge eines normierten Raums) heißt *Fixpunkt* der Abbildung $F : D \rightarrow D$, wenn $F(x^*) = x^*$.
- Eine durch die Vorschrift $x^{(0)} \in D$, $x^{(k+1)} := F(x^{(k)})$ definierte Iteration heißt *Fixpunktiteration*.
- Ist F stetig und konvergiert die Folge $x^{(k)}$ einer Fixpunktiteration gegen einen Grenzwert x^∞ , so ist x^∞ Fixpunkt von F .
- Ist F eine *Kontraktion*, d.h. $\exists \gamma < 1 \forall x, y \in D : \|F(x) - F(y)\| < \gamma \|x - y\|$, so besitzt F in D genau einen Fixpunkt x^* und jede Fixpunktiteration konvergiert gegen x^* . (Banachscher Fixpunktsatz)

Zur Umformulierung des linearen Gleichungssystems $Ax = b$ als Fixpunktproblem schreiben wir A als Differenz

$$A = B - C$$

mit einer invertierbaren Matrix B . Elementare Umformungen zeigen nun, dass x genau dann das lineare Gleichungssystem löst, wenn x Fixpunkt der Abbildung

$$F(x) = \underbrace{B^{-1}C}_{=:M}x + \underbrace{B^{-1}b}_{=:c} = Mx + c$$

ist. Wegen

$$\|F(x) - F(y)\| = \|M(x - y)\| \leq \|M\| \|x - y\|$$

besagt der Banachsche Fixpunktsatz, dass die Iterationsvorschrift

$$x^{(k+1)} := Mx^{(k)} + c \tag{3.1}$$

für beliebige Startvektoren $x^{(0)}$ gegen die Lösung des Gleichungssystems konvergiert, falls $\|M\| < 1$. Dieses Ergebnis wollen wir jetzt verschärfen.

(3.1) Satz: Die Fixpunkt-Iterationsfolge (3.1) konvergiert genau dann für beliebige Startvektoren $x^{(0)}$ gegen die Lösung von $Ax = b$, wenn der Spektralradius $\rho(M) < 1$ ist.

Beweis: $x = A^{-1}b$ sei die gesuchte Lösung und $d^{(k)} := x^{(k)} - x$ der Fehler der k -ten Iterierten. Dann ist

$$d^{(k)} = x^{(k)} - x = M(x^{(k-1)} - x) = \dots = M^k d^{(0)}. \tag{3.2}$$

“ \Rightarrow “: Das Verfahren konvergiere für beliebige Startvektoren. Sei λ_i ein beliebiger Eigenwert von M mit einem zugehörigen Eigenvektor $v^{(i)}$. Zu zeigen ist, dass $|\lambda_i| < 1$. Wir nehmen $|\lambda_i| \geq 1$ an und wählen als Startvektor $x^{(0)} := x + \epsilon v^{(i)}$ mit $\epsilon \neq 0$. Dann ist $d^{(0)} = \epsilon v^{(i)}$, und nach Formel (2.2) ist $d^{(k)} = \epsilon \lambda_i^k v^{(i)}$. Dies ist keine Nullfolge, was im Widerspruch zur Voraussetzung steht.

“ \Leftarrow “: Sei $\rho(M) < 1$. Zu zeigen ist die Konvergenz für beliebige Startvektoren.

Fall 1: M sei diagonalähnlich, d.h. mit Hilfe einer regulären Matrix T lasse sich M darstellen als

$$M = T \cdot \text{diag}(\lambda_1, \dots, \lambda_n) \cdot T^{-1}.$$

In diesem Fall sind die Spalten $v^{(i)} := T^{(i)}$, $i = 1, \dots, n$ von T Eigenvektoren von M zu den Eigenwerten λ_i und bilden eine Basis des \mathbb{R}^n . Sei $x^{(0)}$ ein beliebiger Startvektor der Fixpunktiteration. Der Fehler $d^{(0)}$ kann als Linearkombination der $v^{(i)}$ beschrieben werden:

$$d^{(0)} = \sum_{i=1}^n \alpha_i v^{(i)}.$$

Nach Formel (2.2) ist dann

$$d^{(k)} = \sum_{i=1}^n \alpha_i \lambda_i^k v^{(i)}.$$

Dies ist wegen $|\lambda_i| \leq \rho(M) < 1$ eine Nullfolge.

Fall 2: Ist M nicht diagonalähnlich, so ist sie doch ähnlich zu einer Matrix in Jordan-Diagonalform. Das heißt, dass mit einer geeigneten Matrix T gilt

$$M = T \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_p \end{pmatrix} T^{-1}$$

wobei jede der Teilmatrizen J_r entweder gleich einem der Eigenwerte λ_i (und damit eine 1×1 -Matrix) ist oder die Form eines Jordan-Blocks

$$J_r = \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}$$

annimmt. Wie in Fall 1 sei $v^{(i)}$ die Basis des \mathbb{R}^n welche aus den Spaltenvektoren von T zusammengesetzt wird, und $d^{(0)} = \sum_{i=1}^n \alpha_i v^{(i)}$. Zur Berechnung von $d^{(k)}$ müssen die Potenzen J_r^k berechnet werden. Bezeichnet d_{\max} die maximale Dimension aller Jordan-Blöcke, so kann durch Induktion gezeigt werden, dass für $k > d_{\max}$ jede Zeile der Matrix J_r^k höchstens die Elemente

$$\binom{k}{j} \cdot \lambda_i^{k-j}, \quad j = 0, \dots, d_{\max}$$

enthält. Diese sind durch $k^{d_{\max}} \cdot (\rho(M))^{k-d_{\max}}$ beschränkt. Hieraus folgt mit einer geeigneten Konstante c die Abschätzung

$$\|d^{(k)}\| \leq c \cdot k^{d_{\max}} \cdot (\rho(M))^k,$$

welche zeigt, dass $d^{(k)}$ eine Nullfolge ist. \square

3.2 Die wichtigsten Fixpunkt-Iterationsverfahren

Die spezielle Wahl des Fixpunktverfahrens hängt von der Art der Zerlegung $A = B - C$ ab. Die einzige praktische Forderung besteht zunächst nur darin, dass B^{-1} effizient berechenbar sein muss. In Frage kommen damit für B geeignete Diagonal- oder Dreiecksmatrizen.

Wir zerlegen zunächst A wie folgt in die untere Dreiecksmatrix A_L , die Diagonalmatrix A_D und die obere Dreiecksmatrix A_R :

$$A = \underbrace{\begin{pmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}}_{A_L} + \underbrace{\begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix}}_{A_D} + \underbrace{\begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & 0 & 0 \end{pmatrix}}_{A_R}$$

A – Jacobi- (auch: Gesamtschritt-) Verfahren:

Hier ist

$$B := A_D = \text{diag}(a_{11}, \dots, a_{nn}), \quad C := -A_L - A_R.$$

Die Inverse von B ist

$$B^{-1} = \text{diag}(1/a_{11}, \dots, 1/a_{nn}).$$

Die Iterationsvorschrift (2.1) lautet explizit

$$x_i^{(k+1)} = -\frac{1}{a_{ii}} \left(\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(k)} - b_i \right) \quad (3.3)$$

und wird in folgendem Algorithmus umgesetzt.

(3.2) Algorithmus (Jacobi-Verfahren):

- (S1) Für $i=1(1)n$:
 $x_i^{(k+1)} := b_i$
- (S2)_J Für $j=1(1)n$:
 Falls ($i \neq j$): $x_i^{(k+1)} := x_i^{(k+1)} - a_{ij} x_j^{(k)}$
- (S3) $x_i^{(k+1)} := x_i^{(k+1)} / a_{ii}$.

Die Konvergenz des Verfahrens kann für Matrizen A mit hinreichend großen Diagonalelementen leicht gezeigt werden. A heißt **strikt diagonal dominant**, wenn für $i = 1, \dots, n$ gilt

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$$

(3.3) Satz: Ist A strikt diagonal dominant, so konvergiert das Jacobi-Verfahren für beliebige Startvektoren gegen die Lösung von $Ax = b$.

Beweis: Die Jacobi-Iterationsmatrix $M = -A_D \cdot (A_L + A_R)$ hat die Elemente $m_{ii} = 0$ sowie $m_{ij} = -a_{ij}/a_{ii}$ für $i \neq j$. Wegen der Diagonaldominanz von A gilt für die Zeilensummen

$$|m_{ij}| \leq \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| < 1$$

und für die Zeilensummennorm $\|M\|_Z < 1$. Daher konvergiert $x^{(k)}$ bzgl. jeder zur Zeilensummennorm kompatiblen Vektornorm und somit bzgl. jeder Norm in \mathbb{R}^n . \circ

B – Gauß-Seidel- (auch: Einzelschritt-) Verfahren:

Setzt man in Formel (2.3) auf der rechten Seite alle bereits berechneten neuen Werte $x_j^{(k+1)}$ anstelle der alten Werte $x_j^{(k)}$ ein, so kommt man auf das Gauß-Seidel-Verfahren. Es ist

$$B := A_D + A_L, \quad C := -A_R.$$

Die Iterationsvorschrift lautet

$$x_i^{(k+1)} = -\frac{1}{a_{ii}} \left(\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} + \sum_{j=i+1}^n a_{ij} x_j^{(k)} - b_i \right).$$

Der Algorithmus muss wie folgt geändert werden.

(3.4) Algorithmus (Gauß-Seidel-Verfahren): Ersetze in (2.6) den Schritt $S2_J$ durch

$$\begin{aligned} (S2)_{GS} \quad & \text{Für } j=1(1)i-1: \\ & x_i^{(k+1)} := x_i^{(k+1)} - a_{ij} x_j^{(k+1)} \\ & \text{Für } j=i+1(1)n: \\ & x_i^{(k+1)} := x_i^{(k+1)} - a_{ij} x_j^{(k)} \end{aligned}$$

(3.5) Beispiel: Gelöst werden soll das Gleichungssystem

$$\begin{aligned} 5x_1 - x_2 - 2x_3 &= 2 \\ x_1 - 6x_2 + 3x_3 &= -2 \\ -2x_1 + x_2 + 4x_3 &= 3. \end{aligned}$$

Die Iterationsvorschrift für das Jacobi-Verfahren lautet

$$\begin{aligned} x_1^{(k+1)} &= (2 + x_2^{(k)} + 2x_3^{(k)}) / 5 \\ x_2^{(k+1)} &= (2 + x_1^{(k)} + 3x_3^{(k)}) / 6 \\ x_3^{(k+1)} &= (3 + 2x_1^{(k)} - x_2^{(k)}) / 4. \end{aligned}$$

Beim Gauß-Seidel-Verfahren wird wie folgt modifiziert.

$$\begin{aligned} x_1^{(k+1)} &= (2 + x_2^{(k)} + 2x_3^{(k)}) / 5 \\ x_2^{(k+1)} &= (2 + x_1^{(k+1)} + 3x_3^{(k)}) / 6 \\ x_3^{(k+1)} &= (3 + 2x_1^{(k+1)} - x_2^{(k+1)}) / 4. \end{aligned}$$

k	$x_{J,1}^{(k)}$	$x_{J,2}^{(k)}$	$x_{J,2}^{(k)}$	$x_{GS,1}^{(k)}$	$x_{GS,2}^{(k)}$	$x_{GS,3}^{(k)}$
2	0.7667	0.7750	0.8667	0.7667	0.8361	0.9243
4	0.9547	0.9534	0.9772	0.9826	0.9874	0.9980
6	0.9918	0.9914	0.9955	0.9992	0.9992	0.9998
8	0.9977	0.9984	0.9991	0.9999	0.9999	1.0000
10	0.9995	0.9995	0.9999	1.0000	1.0000	1.0000

Tabelle 1: Numerische Lösung des Beispiels [2.8]

Die Ergebnisse einiger Iterationen zum Startwert $x_0 = x_2 = x_3 = 0$ sind in Tabelle 1 wiedergegeben. Die exakte Lösung ist $x_1 = x_2 = x_3 = 1$.

Ohne Beweis geben wir Kriterien zur Konvergenz an.

(3.6) Satz: Das Gauß-Seidel-Verfahren konvergiert, wenn eine der beiden Bedingungen erfüllt ist.

- (i) A ist strikt diagonaldominant.⁴
- (ii) A ist positiv definit.⁵

(3.7) Beispiel: Für die strikt diagonaldominante Matrix

$$A = \begin{pmatrix} 0.7 & -0.4 \\ -0.2 & 0.5 \end{pmatrix}$$

ist $\rho(M_{GS}) = \rho(M_J)^2 = 8/35$. Das Gauß-Seidel-Verfahren konvergiert damit doppelt so schnell wie das Jacobi-Verfahren.

Den hier beobachteten Zusammenhang zwischen $\rho(M_J)$ und $\rho(M_{GS})$ werden wir nun für eine Klasse von Matrizen nachweisen.

(3.8) Definition: Es sei $A = A_L + A_D + A_R$ wie oben die Zerlegung der (komplexwertigen) Matrix A . A_D sei regulär. A heißt *konsistent geordnet*, falls die (komplexen) Eigenwerte von

$$C(\alpha) := -(\alpha A_D^{-1} A_L + \alpha^{-1} A_D^{-1} A_R), \quad \alpha \in \mathbb{C} \setminus \{0\}$$

⁴vgl. Korollar 4.18 in A. Meister, *Numerik linearer Gleichungssysteme*, Vieweg 1999.

⁵vgl. Satz 8.4 in P. Deuffhard/A. Hohmann, *Numerische Mathematik I*, de Gruyter 1993.

unabhängig von α sind.

(3.9) Beispiel: A habe die Blockstruktur

$$A = \begin{pmatrix} I & A_{12} \\ A_{21} & I \end{pmatrix}.$$

Dann ist

$$A_L = \begin{pmatrix} 0 & 0 \\ A_{21} & 0 \end{pmatrix} \quad \text{und} \quad A_R = \begin{pmatrix} 0 & A_{12} \\ 0 & 0 \end{pmatrix}$$

und es gilt

$$C(\alpha) = \begin{pmatrix} 0 & -\alpha^{-1}A_{12} \\ -\alpha A_{21} & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & -\alpha I \end{pmatrix} \begin{pmatrix} 0 & A_{12} \\ A_{21} & 0 \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & -\alpha^{-1}I \end{pmatrix}.$$

Man rechnet nun leicht nach, dass $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)^T$ genau dann Eigenvektor zu $C(\alpha)$ zum Eigenwert λ ist, wenn $\mathbf{w} = (\alpha\mathbf{v}_1, \mathbf{v}_2)^T$ Eigenvektor von $A - I$ zum Eigenwert λ ist. Die Eigenwerte von $C(\alpha)$ sind damit gleich den Eigenwerten von $A - I$, insbesondere also unabhängig von α . Damit ist A konsistent geordnet.

(3.10) Satz: A sei konsistent geordnet. $\mu \in \mathbb{C} \setminus \{0\}$ ist genau dann Eigenwert von M_{GS} , wenn $\lambda = \mu^{1/2}$ Eigenwert von M_J ist. Insbesondere ist $\rho(M_{GS}) = \rho(M_J)^2$.

Beweis: Zur Bestimmung der Eigenwerte von M_{GS} muss $\det(\mu I - M_{GS}(\omega))$ berechnet werden. Wegen $\det(I + A_D^{-1}A_L) = 1$ ist diese für $\mu \in \mathbb{C} \setminus \{0\}$ gleich der Determinante von

$$\begin{aligned} & (I + A_D^{-1}A_L)(\mu I - M_{GS}) \\ &= (\mu(I + A_D^{-1}A_L) + A_D^{-1}(A_D + A_L) \underbrace{(A_D + A_L)^{-1}A_R}_{M_{GS}} \\ &= +A_D^{-1}(\mu A_L + A_R) \\ &= \mu I + \mu^{1/2}A_D^{-1}(\mu^{1/2}A_L + \mu^{-1/2}A_R). \end{aligned}$$

Damit ist $\mu \in \mathbb{C} \setminus \{0\}$ genau dann Eigenwert von M_{GS} , wenn

$$\det(\mu I + \mu^{1/2}A_D^{-1}(\mu^{1/2}A_L + \mu^{-1/2}A_R)) = 0.$$

Dies wiederum ist genau dann der Fall, wenn $\mu^{1/2}$ Eigenwert von $-A_D^{-1}(\mu^{1/2}A_L + \mu^{-1/2}A_R)$ ist. Da A konsistent geordnet ist, stimmen die Eigenwerte von $-A_D^{-1}(\mu^{1/2}A_L + \mu^{-1/2}A_R)$ und von $-A_D^{-1}(A_L + A_R) = M_J$ überein. \circ

3.3 Relaxationsverfahren

Relaxationsverfahren sind Modifikationen der oben vorgestellten Iterationsverfahren. Zu ihrer Herleitung beschreiben wir die Iteration (2.1), $x^{(k+1)} = Mx^{(k)} + c$, in der Form

$$x^{(k+1)} = x^{(k)} + A_D^{-1}r^{(m)}, \quad (3.4)$$

wobei A_D wieder der Diagonalanteil von A ist und $r^{(m)}$ aus dem Iterationsverfahren hergeleitet werden muss. Durch Einführung eines *Relaxationsparameters* ω wandeln wir diese Gleichung um in das **Relaxationsschema**

$$x^{(k+1)} = x^{(k)} + \omega \cdot A_D^{-1}r^{(m)}. \quad (3.5)$$

Zunächst ist unmittelbar klar, dass im Falle der Konvergenz die Gleichungen (2.4) und (2.5) auf die selben Limites führen. Die Wahl $\omega = 1$ führt auf das ursprüngliche Verfahren zurück. Die Hoffnung bei dem modifizierten Ansatz besteht darin, dass ω geeignet gewählt werden kann zur Beschleunigung der Konvergenz. Die Anwendung des Ansatzes auf die oben beschriebenen Verfahren liefert

A – Das Jacobi-Relaxationsverfahren

Dies ergibt sich aus dem Jacobi-Verfahren:

$$x^{(k+1)} = -A_D^{-1}(A - A_D)x^{(k)} + A_D^{-1}b = x^{(k)} + A_D^{-1}(b - Ax^{(k)})$$

zu

$$x^{(k+1)} = x^{(k)} + \omega \cdot A_D^{-1}(b - Ax^{(k)}) = \left((1 - \omega)I - \omega \cdot A_D^{-1}(A_L + A_R) \right) x^{(k)} + \omega A_D^{-1}b.$$

Die Iterationsmatrix ist damit gegeben als

$$M_J(\omega) = (1 - \omega)I - \omega \cdot A_D^{-1}(A_L + A_R). \quad (3.6)$$

Die praktische Durchführung besteht darin, ausgehend von der k -ten Iterierten $x^{(k)}$ zunächst die nächste Jacobi-Iterierte $x_J^{(k+1)}$ nach Formel (2.3) zu berechnen und anschließend zu korrigieren durch $x^{(k+1)} := (1 - \omega)x^{(k)} + \omega x_J^{(k+1)}$.

(3.11) Satz: Die Jacobi-Iterationsmatrix M_J sei diagonalähnlich mit den (reellwertigen) Eigenwerten $\lambda_1, \dots, \lambda_n$ und es sei $\rho(M_J) < 1$. Dann ist der optimale Parameter für das Jacobi-Relaxationsverfahren gegeben durch

$$\omega_{opt} = \frac{2}{2 - \lambda_{max} - \lambda_{min}}$$

und es gilt

$$\rho(M_J(\omega_{opt})) = \frac{\lambda_{max} - \lambda_{min}}{2 - \lambda_{max} - \lambda_{min}}.$$

Beweis: O. B. d. A. können wir voraussetzen, dass die Eigenwerte in der folgenden Reihenfolge numeriert sind:

$$\lambda_1 \leq \dots \leq \lambda_n.$$

Aus (3.6) folgt, dass die Eigenwerte μ_i von $M_J(\omega)$ gegeben sind durch

$$\mu_i = \omega \cdot \lambda_i + (1 - \omega).$$

Die Menge $\{\mu_1, \mu_n\}$ enthält den kleinsten und größten Eigenwert von $M_J(\omega)$. Zur Bestimmung des Parameters ω^* , für welchen die Beträge dieser Werte gleich sind, setzen wir

$$\omega^* \cdot \lambda_1 + (1 - \omega^*) = -\omega^* \cdot \lambda_n - (1 - \omega^*)$$

und erhalten hieraus

$$\omega^* = \frac{2}{2 - \lambda_n - \lambda_1} > 0.$$

Damit ist

$$\rho(M_J(\omega^*)) = \frac{\lambda_n - \lambda_1}{2 - \lambda_n - \lambda_1}.$$

Die μ_i sind lineare, streng monoton fallende Funktionen von ω , und es ist $\mu_1(\omega^*) \leq 0$ und $\mu_n(\omega^*) \geq 0$. Damit wächst $|\mu_n(\omega)|$ für fallendes und $|\mu_1(\omega)|$ für wachsendes ω , und das Minimum von $\rho(M_J(\omega))$ wird angenommen für ω^* . \circ

B – Das SOR- (“successive overrelaxation”-) Verfahren

Ausgehend von der Beziehung der Iterierten des Gauß-Seidel-Verfahrens

$$(A_L + A_D)x^{(k+1)} = -A_Rx^{(k)} + b$$

erhalten wir durch elementare Umformungen

$$\begin{aligned} x^{(k+1)} &= -A_D^{-1}A_Lx^{(k+1)} - A_D^{-1}A_Rx^{(k)} + A_D^{-1}b \\ &= x^{(k)} + A_D^{-1} \left(b - A_Lx^{(k+1)} - (A_R + A_D)x^{(k)} \right) \end{aligned}$$

und hieraus das *Gauß-Seidel-Relaxationsverfahren*

$$x^{(k+1)} = x^{(k)} + \omega A_D^{-1} \left(b - A_L x^{(k+1)} - (A_R + A_D) x^{(k)} \right)$$

mit der Iterationsmatrix

$$\begin{aligned} M_{GS}(\omega) &= (I + \omega A_D^{-1} A_L)^{-1} \left((1 - \omega)I - \omega A_D^{-1} A_R \right) \\ &= (A_D + \omega A_L)^{-1} \left((1 - \omega)A_D - \omega A_R \right). \end{aligned}$$

Die Menge der möglichen Parameter ω kann wie folgt eingeschränkt werden.

(3.12) Satz: Es sei $a_{ii} \neq 0$ für $i = 1, \dots, n$. Dann ist

$$\rho(M_{GS}(\omega)) \geq |\omega - 1|.$$

Folglich konvergiert das Gauß-Seidel-Relaxationsverfahrens höchstens für $\omega \in (0, 2)$.

Beweis: Es seien $\lambda_1, \dots, \lambda_n$ die (komplexwertigen) Eigenwerte von M_{GS} . Dann gilt

$$\begin{aligned} \prod_{i=1}^n \lambda_i &= \det(M_{GS}(\omega)) = \det((A_D + \omega \cdot A_L)^{-1}) \det((1 - \omega) \cdot A_D - \omega \cdot A_R) \\ &= \det(A_D^{-1}) \det((1 - \omega) \cdot A_D) \\ &= (1 - \omega)^n \cdot \det(A_D^{-1}) \det(A_D) = (1 - \omega)^n. \end{aligned}$$

Damit ist $\max_{i=1, \dots, n} |\lambda_i| \geq |1 - \omega|$. $\quad \circ$

Aussagen über den optimalen Parameter sind im Allgemeinen schwer zu formulieren. Für konsistent geordnete Matrizen leiten wir ein Ergebnis her. Zunächst beweisen wir eine Verallgemeinerung von Satz (3.10).

(3.13) Satz: A sei konsistent geordnet. Es sei $\omega \in (0, 2)$. $\mu \in \mathbb{C} \setminus \{0\}$ ist genau dann Eigenwert von $M_{GS}(\omega)$, wenn $\lambda = (\mu + \omega - 1)/(\omega\mu^{1/2})$ Eigenwert von M_J ist.

Beweis: Zur Bestimmung der Eigenwerte von M_{GS} muss $\det(\mu I - M_{GS}(\omega))$ berechnet werden. Wegen $\det(I + \omega A_D^{-1} A_L) = 1$ ist diese für $\mu \in \mathbb{C} \setminus \{0\}$ gleich der Determinante von

$$(I + \omega A_D^{-1} A_L)(\mu I - M_{GS}(\omega))$$

$$\begin{aligned}
&= (\mu(I + \omega A_D^{-1} A_L) - A_D^{-1}(A_D + \omega A_L) \underbrace{(A_D + \omega A_L)^{-1}((1 - \omega)A_D - \omega A_R)}_{M_{GS}(\omega)}) \\
&= (\mu - (1 - \omega))I + \omega A_D^{-1}(\mu A_L + A_R) \\
&= (\mu - (1 - \omega))I + \omega \mu^{1/2} A_D^{-1}(\mu^{1/2} A_L + \mu^{-1/2} A_R).
\end{aligned}$$

Damit ist $\mu \in \mathbb{C} \setminus \{0\}$ genau dann Eigenwert von $M_{GS}(\omega)$, wenn

$$\det\left((\mu - (1 - \omega))I + \omega \mu^{1/2} A_D^{-1}(\mu^{1/2} A_L + \mu^{-1/2} A_R)\right) = 0.$$

Dies wiederum ist genau dann der Fall, wenn $(\mu - (1 - \omega))/\omega \mu^{1/2}$ Eigenwert von $-A_D^{-1}(\mu^{1/2} A_L + \mu^{-1/2} A_R)$ ist. Da A konsistent geordnet ist, stimmen die Eigenwerte von $-A_D^{-1}(\mu^{1/2} A_L + \mu^{-1/2} A_R)$ und von $-A_D^{-1}(A_L + A_R) = M_J$ überein. \circ

(3.14) Satz: A sei konsistent geordnet. Die Eigenwerte von M_J seien reell und es gelte

$$\rho := \rho(M_J) < 1.$$

Dann gilt:

- (i) Das Gauß-Seidel-Verfahren konvergiert für alle $\omega \in (0, 2)$.
- (ii) Der Spektralradius von $M_{GS}(\omega)$ wird minimal für

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2}},$$

und es ist

$$\rho(M_{GS}(\omega_{opt})) = \omega_{opt} - 1 = \frac{1 - \sqrt{1 - \rho^2}}{1 + \sqrt{1 - \rho^2}}.$$

Beweis: Seien $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ die Eigenwerte von M_J . μ ist genau dann Eigenwert von $M_{GS}(\omega)$, wenn

$$\lambda = \frac{\mu + \omega - 1}{\omega \mu^{1/2}} \in \{\lambda_1, \dots, \lambda_n\}$$

Da A konsistent geordnet ist, ist mit λ auch $-\lambda$ Eigenwert von M_J . Wir können also $\lambda > 0$ voraussetzen. Aus

$$\lambda^2 \omega^2 \mu = (\mu + \omega - 1)^2$$

erhalten wir die beiden Eigenwerte

$$\mu^\pm = \mu^\pm(\omega, \lambda) = \frac{1}{2}\lambda^2\omega^2 - (\omega - 1) \pm \lambda\omega\sqrt{\frac{1}{4}\lambda^2\omega^2 - (\omega - 1)}$$

Die Nullstellen von

$$g(\omega, \lambda) := \frac{1}{4}\lambda^2\omega^2 - (\omega - 1)$$

sind

$$\omega^\pm = \omega^\pm(\lambda) = \frac{2}{1 \pm \sqrt{1 - \lambda^2}}$$

von denen nur $\omega^+ \in (0, 2)$ ist. Für $\lambda \in [0, 1)$ und $\omega \in (0, 2)$ ist

$$\frac{\partial g}{\partial \omega}(\omega, \lambda) = \frac{1}{2}\lambda^2\omega - 1 < 0$$

Fall 1, $2 > \omega > \omega^+(\lambda)$: Die beiden Eigenwerte μ^\pm sind komplex und es ist

$$|\mu^+(\omega, \lambda)| = |\mu^-(\omega, \lambda)| = |\omega - 1| = \omega - 1$$

Fall 2, $\omega = \omega^+(\lambda)$: Es folgt $\lambda^2 = 4/\omega - 4/\omega^2$ und

$$|\mu^+(\omega, \lambda)| = |\mu^-(\omega, \lambda)| = \frac{1}{2}\lambda^2\omega^2 - (\omega - 1) = 2\omega - 2 - (\omega - 1) = \omega - 1$$

Fall 3, $0 < \omega < \omega^+(\lambda)$: $M_{GS}(\omega)$ hat die zwei reellen Eigenwerte

$$\mu^\pm(\omega, \lambda) = \underbrace{\frac{1}{2}\lambda^2\omega^2 - (\omega - 1)}_{>0} \pm \lambda\omega \underbrace{\sqrt{\frac{1}{4}\lambda^2\omega^2 - (\omega - 1)}}_{>0}$$

mit

$$\max\{|\mu^+(\omega, \lambda)|, |\mu^-(\omega, \lambda)|\} = \mu^+(\omega, \lambda)$$

Für $\lambda \in [0, 1)$ definiere

$$\mu(\omega, \lambda) = \begin{cases} \mu^+(\omega, \lambda) & \text{für } 0 < \omega < \omega^+(\lambda) \\ \omega - 1 & \text{für } \omega^+(\lambda) \leq \omega < 2 \end{cases}$$

Für $0 < \omega < \omega^+(\lambda)$ und $\lambda \in (0, 1)$ ist

$$\frac{\partial \mu}{\partial \lambda}(\omega, \lambda) = \underbrace{\lambda\omega^2}_{\geq 0} + \omega \underbrace{\sqrt{\frac{1}{4}\lambda^2\omega^2 - (\omega - 1)}}_{>0} + \frac{1}{2}\lambda\omega \underbrace{\frac{\frac{1}{2}\lambda\omega^2}{\sqrt{\frac{1}{4}\lambda^2\omega^2 - (\omega - 1)}}}_{>0} > 0$$

Wegen

$$\mu(\omega, \lambda) = \left(\frac{\omega\lambda}{2} + \sqrt{\frac{1}{4}\lambda^2\omega^2 - (\omega - 1)} \right)$$

ist

$$\frac{\partial\mu}{\partial\omega}(\omega, \lambda) = 2 \underbrace{\left(\frac{\omega\lambda}{2} + \sqrt{\frac{1}{4}\lambda^2\omega^2 - (\omega - 1)} \right)}_{>0} \underbrace{\left[\frac{\lambda}{2} + \frac{1}{2} \frac{\frac{1}{2}\lambda^2\omega - 1}{\sqrt{\frac{1}{4}\lambda^2\omega^2 - (\omega - 1)}} \right]}_{=:q(\omega, \lambda)}$$

Setze

$$q(\omega, \lambda) = \frac{1}{2\sqrt{\frac{1}{4}\lambda^2\omega^2 - (\omega - 1)}} \left(\underbrace{\lambda\sqrt{\frac{1}{4}\lambda^2\omega^2 - (\omega - 1)}}_{=:q_1(\omega, \lambda)} + \underbrace{\frac{1}{2}\lambda^2\omega - 1}_{=:q_2(\omega, \lambda)} \right)$$

Für $\lambda \in [0, 1)$ und $\omega \in (0, \omega^+(\lambda))$ ist $q_1(\omega, \lambda) > 0$ und $q_2(\omega, \lambda) < 0$. Außerdem liefert

$$[q_1(\omega, \lambda)]^2 = \frac{\omega^2\lambda^4}{4} + \lambda^2 - \omega\lambda^2 < \frac{\omega^2\lambda^4}{4} + 1 - \omega\lambda^2 = [q_2(\omega, \lambda)]^2$$

die Ungleichung

$$\frac{\partial\mu}{\partial\omega}(\omega, \lambda) < 0 \quad \text{für alle } \lambda \in [0, 1) \quad \text{und } \omega \in (0, \omega^+(\lambda))$$

Da außerdem $\mu(0, \lambda) = \mu(2, \lambda)$ ist, folgt $\rho(M_{GS}(\omega) < 1$ für beliebige $\omega \in (0, 2)$, und das Minimum von $|\mu(\omega, \lambda)|$ wird für $\omega_{opt} = \omega^+(\lambda)$ angenommen. Damit ist

$$\rho(M_{GS}(\omega_{opt})) = |\mu(\omega_{opt}, \rho(M_J))| = \omega_{opt}(\rho(M_J)) - 1 = \frac{2}{1 + \sqrt{1 - \rho^2}} - 1 \quad \circlearrowright$$

(3.15) Zahlenbeispiele: Sei A konsistent geordnet.

(a) Ist $\rho(M_J) = 0.1$, so ist $\omega_{opt} = 1.0025$, also $\rho(M_{GS}(\omega_{opt})) = 0.0025$. Wegen $0.0025 = 0.1^{2.60}$ bedeutet dies, dass ein SOR-Schritt die gleiche Qualität hat wie 2.60 Jacobi-Rechenschritte.

(b) Ist $\rho(M_J) = 0.5$, so ist $\omega_{opt} = 1.072$, also $\rho(M_{GS}(\omega_{opt})) = 0.072$. Ein SOR-Schritt entspricht 3.80 Jacobi-Schritten.

(c) Ist $\rho(M_J) = 0.9$, so ist $\omega_{opt} = 1.393$ und $\rho(M_{GS}(\omega_{opt})) = 0.393$. Die Fehlerverbesserung durch einen SOR-Schritt entspricht der Verbesserung durch 8.86 Jacobi-Schritte.

4 Gradientenverfahren

Die Lösung eines linearen Gleichungssystems mit einer positiv definiten Matrix A kann durch ein äquivalentes Extremwertproblem formuliert und gelöst werden. Dies wird in den beiden nächsten Abschnitten erläutert. Dort wird also immer vorausgesetzt werden, dass A positiv definit ist.

4.1 Lineare Gleichungssysteme als Extremwertprobleme

A sei positiv definit. Wir definieren das quadratische Funktional

$$q_A(\mathbf{x}) := \frac{1}{2} \cdot \mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{b}. \quad (4.1)$$

Da A positiv definit ist, ist A ähnlich zu einer Diagonalmatrix mit positiven Diagonalelementen. Daher bildet der Graph von q_A ein nach oben geöffnetes Paraboloid. $q_A(\cdot)$ hat damit ein eindeutig bestimmtes Minimum. Die Äquivalenz der Lösung des Gleichungssystems $A\mathbf{x} = \mathbf{b}$ mit der Bestimmung des Minimums von $q_A(\mathbf{x})$ wird im folgenden Satz bewiesen.

(4.1) Satz: Ist A positiv definit, so hat das Gleichungssystem $A\mathbf{x} = \mathbf{b}$ eine eindeutig bestimmte Lösung $\overset{\circ}{\mathbf{x}}$. An der Stelle $\overset{\circ}{\mathbf{x}}$ nimmt q_A sein eindeutig bestimmtes Minimum an, d.h.

$$q_A(\overset{\circ}{\mathbf{x}}) < q_A(\mathbf{x}) \quad \text{für alle } \mathbf{x} \in \mathbb{R}^n \setminus \{\overset{\circ}{\mathbf{x}}\}.$$

Beweis: Die Regularität von positiv definiten Matrizen und damit die eindeutige Lösbarkeit der zugehörigen Gleichungssysteme ist bekannt. Zu zeigen ist der Zusammenhang mit der Extremwerteigenschaft von q_A .

(a) Wir untersuchen zuerst den Fall, dass A eine Diagonalmatrix ist, $A = \text{diag}(d_1, \dots, d_n)$. Es gilt $d_i > 0$ für alle i . Das Funktional ist gegeben durch

$$q_A(\mathbf{x}) = \sum_{i=1}^n \left(\frac{1}{2} d_i x_i^2 - x_i b_i \right).$$

Das Minimum von q_A erhalten wir durch die Bedingung

$$\nabla_x q = (\partial_{x_1} q_A, \dots, \partial_{x_n} q_A)^T = 0.$$

Wegen

$$\partial_{x_j} q_A = \frac{\partial}{\partial x_j} \left(\frac{1}{2} d_j x_j^2 - x_j b_j \right) = d_j x_j - b_j = (A\mathbf{x} - \mathbf{b})_j$$

führt dies auf die Forderung $A\mathbf{x} = \mathbf{b}$, also auf die Lösung des Gleichungssystems.

(b) Betrachten wir nun den allgemeinen Fall. A ist ähnlich zu einer Diagonalmatrix $D = \text{diag}(d_1, \dots, d_n)^T$ mit einer orthogonalen Transformationsmatrix P , d.h. $A = P^T D P$ – oder umgekehrt $D = P A P^T$. Wie in (a) gilt $d_i > 0 \forall i$. Außerdem ist

$$q_A(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T P^T D P \mathbf{x} - \mathbf{x}^T \mathbf{b} = \frac{1}{2} (P\mathbf{x})^T D (P\mathbf{x}) - (P\mathbf{x})^T P \mathbf{b}.$$

Wir definieren $\tilde{\mathbf{x}} := P\mathbf{x}$ und

$$q_D(\tilde{\mathbf{x}}) := q \circ P^{-1}(\tilde{\mathbf{x}}) = \frac{1}{2} \tilde{\mathbf{x}}^T D \tilde{\mathbf{x}} - \tilde{\mathbf{x}}^T P \mathbf{b}.$$

Offenbar liegt das Minimum von q_A genau dann in \mathbf{x} , wenn das Minimum von q_D in $\tilde{\mathbf{x}} = P\mathbf{x}$ angenommen wird. Das Minimum von q_D wird nach (a) in dem Wert $\tilde{\mathbf{x}}$ angenommen, welcher Lösung des Gleichungssystems $D\tilde{\mathbf{x}} = P\mathbf{b}$ ist. Dann ist aber \mathbf{x} Lösung von $A\mathbf{x} = \mathbf{b}$. \circ

Wir wollen nun ein Iterationsverfahren zur Bestimmung des Minimums von q_A konstruieren. Hierzu legen wir zunächst einen Startwert $\mathbf{x}^{(0)}$ fest und bestimmen einen Richtungsvektor $\mathbf{p}^{(0)}$. In einem zweiten Schritt ermitteln wir diejenige Stelle $\mathbf{x}^{(1)}$ auf der Geraden

$$\mathcal{G}^{(0)} := \{\mathbf{x}^{(0)} - \alpha \cdot \mathbf{p}^{(0)} \mid \alpha \in \mathbb{R}\},$$

auf welcher die Funktion

$$q^{(0)}(\alpha) := q_A(\mathbf{x}^{(0)} + \alpha \cdot \mathbf{p}^{(0)})$$

ihr Minimum annimmt. Von hier aus wird die Suche in einer anderen Richtung $\mathbf{p}^{(1)}$ fortgesetzt. Zur Bestimmung von $\mathbf{x}^{(1)}$ hilft das folgende Lemma.

(4.2) Lemma: Gegeben seien zwei Vektoren \mathbf{x}, \mathbf{p} ($\mathbf{p} \neq 0$). Das Minimum der Funktion q_A auf der Geraden

$$\mathcal{G} := \{\mathbf{x} - \alpha \cdot \mathbf{p} \mid \alpha \in \mathbb{R}\}$$

wird angenommen für

$$\alpha_{min} = \frac{\mathbf{p}^T(A\mathbf{x} - \mathbf{b})}{\mathbf{p}^T A \mathbf{p}}.$$

Beweis: Wie oben bezeichnen wir

$$\begin{aligned} q(\alpha) &:= q_A(\mathbf{x} - \alpha \cdot \mathbf{p}) = \frac{1}{2}(\mathbf{x}^T - \alpha \mathbf{p}^T)A(\mathbf{x} - \alpha \mathbf{p}) - (\mathbf{x}^T - \alpha \mathbf{p}^T)\mathbf{b} \\ &= \frac{1}{2}\alpha^2 \cdot \mathbf{p}^T A \mathbf{p} - \alpha \cdot (\mathbf{p}^T A \mathbf{x} - \mathbf{p}^T \mathbf{b}) + \left(\frac{1}{2}\mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{b}\right). \end{aligned}$$

Der Wert α_{min} ist gegeben als Nullstelle der Ableitung

$$q'(\alpha) = \alpha \cdot \mathbf{p}^T A \mathbf{p} - \mathbf{p}^T(A\mathbf{x} - \mathbf{b}). \quad \circ$$

Die Iterationsvorschrift für das oben skizzierte allgemeine Verfahren lautet

(4.3) Iterationsschritt: Gegeben seine $\mathbf{x}^{(k)}$ und die Richtungen $\mathbf{p}^{(0)}, \dots, \mathbf{p}^{(k)}$.

S1 Definiere

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} - \alpha_{min} \cdot \mathbf{p}^{(k)} \quad \text{mit} \quad \alpha_{min} = \frac{(\mathbf{p}^{(k)})^T(A\mathbf{x}^{(k)} - \mathbf{b})}{(\mathbf{p}^{(k)})^T A \mathbf{p}^{(k)}}.$$

S2 Lege (ggf. in Abhängigkeit von den bisherigen Richtungen $\mathbf{p}^{(i)}, i = 1, \dots, k$) eine neue Richtung $\mathbf{p}^{(k+1)}$ fest.

Die Freiheit in der Ausgestaltung des Iterationsschritts liegt in der Wahl der neuen Richtung. Naheliegender wäre es, die Richtung $\mathbf{p}^{(k+1)}$ als die Richtung des steilsten Abstiegs von q_A im Punkt $\mathbf{x}^{(k+1)}$ zu wählen.

(4.4) Beispiel (Gradientenverfahren, Verfahren des steilsten Abstiegs): Gegeben sei die k -te Iterierte $\mathbf{x}^{(k)}$. Als neue Richtung $\mathbf{p}^{(k)}$ wird diejenige Richtung bestimmt, in welcher das Funktional q_A von $\mathbf{x}^{(k)}$ aus am schnellsten abfällt. Diese Richtung ist gegeben als Gradient von q_A , also

$$\mathbf{p}^{(k)} = \nabla q_A(\mathbf{x}^{(k)}) = A\mathbf{x}^{(k)} - \mathbf{b}.$$

Auf diese Art erhalten wir in der Tat ein konvergentes Verfahren. Allerdings ist die Konvergenz gegen die gesuchte Lösung unter Umständen sehr langsam (insbesondere, wenn für die Eigenwerte von A gilt $\lambda_{\max}/\lambda_{\min} \gg 1$). Daher ist diese Methode praktisch nicht interessant.

4.2 Das Verfahren der konjugierten Gradienten (cg-Verfahren)

Aus Gründen, welche unten näher ausgeführt werden, erweist es sich als sinnvoll, in jedem Iterationsschritt solche Richtungsvektoren zu wählen, welche “senkrecht“ auf den vorher gewählten Richtungen stehen – allerdings nicht in dem üblichen, durch das euklidische Skalarprodukt gegebenen Sinn, sondern bezüglich des durch A gegebenen Skalarprodukts (vgl. Abschnitt 1.3.1).

(4.5) Definition: Zur positiv definiten Matrix A definieren wir das Skalarprodukt

$$\langle \mathbf{x}, \mathbf{y} \rangle_A := \mathbf{x}^T A \mathbf{y}.$$

Die Richtungen $\mathbf{p}^{(0)}, \dots, \mathbf{p}^{(n-1)}$ heißen *zueinander konjugiert*, wenn $\langle \mathbf{p}^{(i)}, \mathbf{p}^{(j)} \rangle_A = 0$ für alle $i \neq j$. (Gelegentlich werden solche Vektoren auch als *A-orthogonal* bezeichnet.)

(Man beachte: Im Gegensatz hierzu bezeichnet $\langle \cdot, \cdot \rangle$ das übliche euklidische Skalarprodukt!)

Ein erstes Argument, welches für die Wahl zueinander konjugierter Richtungen spricht, ist, dass das Verfahren (4.3) im Gegensatz zu den Iterationsverfahren des Abschnitts 3 bereits *nach endlich vielen Schritten* gegen die Lösung konvergiert, wie der folgende Satz zeigt.

(4.6) Satz: Sind die Richtungen $\mathbf{p}^{(0)}, \dots, \mathbf{p}^{(n-1)}$ zueinander konjugiert gewählt, so konvergiert das Iterationsverfahren (4.3) für beliebige Startvektoren $\mathbf{x}^{(0)}$ *nach höchstens n Schritten* gegen die Lösung von $A\mathbf{x} = \mathbf{b}$.

Beweis: Wir untersuchen die Komponenten der Residuen $A\mathbf{x}^{(k)} - \mathbf{b}$ in Richtung der Vektoren $\mathbf{p}^{(j)}$.

(i), $j < k$: Wegen $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{p}^{(k)}$ ist

$$\langle \underbrace{A\mathbf{x}^{(k+1)} - \mathbf{b}}_{(k+1)\text{-tes Resid.}}, \mathbf{p}^{(j)} \rangle = \langle A\mathbf{x}^{(k)} - \mathbf{b}, \mathbf{p}^{(j)} \rangle - \alpha_k \underbrace{\langle \mathbf{p}^{(k)}, \mathbf{p}^{(j)} \rangle_A}_{=0} = \langle \underbrace{A\mathbf{x}^{(k)} - \mathbf{b}}_{k\text{-tes Resid.}}, \mathbf{p}^{(j)} \rangle.$$

Das bedeutet, dass sich im Verlauf der Iteration für $k > j$ diejenige Komponente des Residuums, welche in Richtung $\mathbf{p}^{(j)}$ zeigt, nicht mehr ändert.

(ii), $j = k$: Wegen

$$\alpha_j = \frac{\langle A\mathbf{x}^{(j)} - \mathbf{b}, \mathbf{p}^{(j)} \rangle}{\langle \mathbf{p}^{(j)}, \mathbf{p}^{(j)} \rangle_A}$$

ist

$$\langle A\mathbf{x}^{(j+1)} - \mathbf{b}, \mathbf{p}^{(j)} \rangle = \langle A\mathbf{x}^{(j)} - \mathbf{b}, \mathbf{p}^{(j)} \rangle - \alpha_j \langle \mathbf{p}^{(j)}, \mathbf{p}^{(j)} \rangle_A = 0.$$

(iii) Nach (i) und (ii) gilt für $j = 0, \dots, n-1$

$$\langle A\mathbf{x}^{(n)} - \mathbf{b}, \mathbf{p}^{(j)} \rangle \stackrel{(i)}{=} \langle A\mathbf{x}^{(n-1)} - \mathbf{b}, \mathbf{p}^{(j)} \rangle \stackrel{(i)}{=} \dots \stackrel{(i)}{=} \langle A\mathbf{x}^{(j+1)} - \mathbf{b}, \mathbf{p}^{(j)} \rangle \stackrel{(ii)}{=} 0.$$

Da $\mathbf{p}^{(0)}, \dots, \mathbf{p}^{(n-1)}$ eine Basis des \mathbb{R}^n bilden, muss gelten $A\mathbf{x}^{(n)} - \mathbf{b} = 0$. \square

Das zweite Argument für die Wahl konjugierter Richtungen besteht darin, dass solche Richtungen ohne allzu großen Rechenzeit- und Speicheraufwand bestimmt werden können. Betrachten wir jedoch zunächst zwei Beispiele, wie man es nicht machen sollte.

(4.7) Beispiele: (a) Da A symmetrisch ist, gibt es einen Satz $\mathbf{e}_1, \dots, \mathbf{e}_n$ von paarweise senkrecht aufeinander stehenden Eigenvektoren, d.h. $\mathbf{e}_i^T \mathbf{e}_j = 0$ für $i \neq j$. Damit sind diese Vektoren auch A -orthogonal, denn

$$\langle \mathbf{e}_i, \mathbf{e}_j \rangle_A = \mathbf{e}_i^T A \mathbf{e}_j = \lambda_j \mathbf{e}_i^T \mathbf{e}_j = 0.$$

Da die Bestimmung der Eigenvektoren in der Regel sehr aufwändig ist, ist diese Wahl praktisch unbrauchbar.

(b) Gegeben seien beliebige linear unabhängige Vektoren $\mathbf{y}_0, \dots, \mathbf{y}_{n-1}$. Mit Hilfe des Gram-Schmidt-Verfahrens (vgl. Abschnitt 2.2.1) können hieraus n zueinander konjugierte Richtungen $\mathbf{p}^{(0)}, \dots, \mathbf{p}^{(n-1)}$ konstruiert werden. (Übung!) Man überzeugt sich jedoch leicht, dass für diese Konstruktion alle Richtungsvektoren abgespeichert werden müssen. Darüber hinaus ist der Rechenaufwand sehr groß.

Ein Verfahren, mit dem die Bestimmung der Richtung $\mathbf{p}^{(k+1)}$ schnell und lediglich aus der Kenntnis von $\mathbf{p}^{(k)}$ möglich ist (die anderen Richtungen $\mathbf{p}^{(j)}, j < k$, werden nicht mehr benötigt und müssen daher nicht weiter abgespeichert werden), ist das *Verfahren der konjugierten Gradienten* (auch "cg-Verfahren", *cg=conjugate gradients*). Zu seiner

Herleitung nehmen wir an, dass die Richtungen $\mathbf{p}^{(0)}, \dots, \mathbf{p}^{(k)}$ sowie $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{p}^{(k)}$ bereits bestimmt seien. Das $(k+1)$ -te Residuum ist definiert durch $\mathbf{r}^{(k+1)} = \mathbf{b} - A\mathbf{x}^{(k+1)}$. Zur Festlegung der Richtung $\mathbf{p}^{(k+1)}$ wählen wir den Ansatz

$$\mathbf{p}^{(k+1)} := \mathbf{r}^{(k+1)} + \sum_{i=0}^k c_i \mathbf{p}^{(i)},$$

wobei die c_i so zu wählen sind, dass

$$\langle \mathbf{p}^{(k+1)}, \mathbf{p}^{(j)} \rangle_A = 0, \quad j = 0, \dots, k. \quad (4.2)$$

Sind $\mathbf{p}^{(0)}, \dots, \mathbf{p}^{(k)}$ zueinander konjugiert, so ist

$$\langle \mathbf{p}^{(k+1)}, \mathbf{p}^{(j)} \rangle_A = \langle \mathbf{r}^{(k+1)}, \mathbf{p}^{(j)} \rangle_A + c_j \langle \mathbf{p}^{(j)}, \mathbf{p}^{(j)} \rangle_A.$$

Ähnlich wie im Beweis des Satzes (4.6) die Orthogonalität von $\mathbf{r}^{(k+1)}$ und $\mathbf{p}^{(j)}$ für $j \leq k$ gezeigt wurde, kann auch gezeigt werden, dass $\mathbf{r}^{(k+1)}$ und $\mathbf{p}^{(j)}$ für $j < k$ A-orthogonal sind. Zur Erfüllung der Bedingungen (4.2) muss also gelten

$$c_0 = \dots = c_{k-1} = 0.$$

Für $j = k$ folgt

$$c_k = -\frac{\langle \mathbf{r}^{(k+1)}, \mathbf{p}^{(k)} \rangle_A}{\langle \mathbf{p}^{(k)}, \mathbf{p}^{(k)} \rangle_A}.$$

Das führt auf das folgende numerische Verfahren.

(4.8) Algorithmus (cg-Verfahren):

S1 Wähle Startvektor $\mathbf{x}^{(0)}$ und berechne $\mathbf{p}^{(0)} = \mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$.

S2 Für $k = 0, 1, 2, \dots$ berechne

(a) $\alpha_k = -\langle \mathbf{p}^{(k)}, \mathbf{r}^{(k)} \rangle / \langle \mathbf{p}^{(k)}, \mathbf{p}^{(k)} \rangle_A;$

(b) $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{p}^{(k)};$

(c) $\mathbf{r}^{(k+1)} = \mathbf{b} - A\mathbf{x}^{(k+1)};$

falls $\|\mathbf{r}^{(k+1)}\|_2^2 < \epsilon$: Stop;

(d) $\beta_k = -\langle \mathbf{p}^{(k)}, \mathbf{r}^{(k+1)} \rangle_A / \langle \mathbf{p}^{(k)}, \mathbf{p}^{(k)} \rangle_A;$

(e) $\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta_k \mathbf{p}^{(k)}.$

Die numerische Effizienz des Verfahrens kann durch folgende Beobachtung verbessert

werden.

(4.9) Bemerkung: Genauere Analysen zeigen, dass sich α_k und β_k schreiben lassen in der Form

$$\alpha_k = -\frac{\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle}{\langle \mathbf{p}^{(k)}, A\mathbf{p}^{(k)} \rangle}, \quad \beta_k = -\frac{\langle \mathbf{r}^{(k+1)}, \mathbf{r}^{(k+1)} \rangle}{\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle}.$$

Damit erfordert ein cg-Iterationsschritt als komplexere Operationen

- eine Matrix-Vektor-Multiplikation $A\mathbf{p}^{(k)}$ und
- zwei Skalarprodukte $\langle \mathbf{r}^{(k+1)}, \mathbf{r}^{(k+1)} \rangle$ und $\langle \mathbf{p}^{(k)}, A\mathbf{p}^{(k)} \rangle$.

Mit Hilfe der durch das Skalarprodukt $\langle \cdot, \cdot \rangle_A$ definierten Norm $\|\mathbf{v}\|_A = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle_A}$ können für die Iterierten der oben besprochenen Verfahren Fehlerabschätzungen gefunden werden. Beispielsweise gilt

(4.10) Satz: (a) A sei symmetrisch und positiv definit. $\mathbf{x}^{(k)}$ sei die k -te Iterierte des Gradientenverfahrens; der zugehörige Fehlervektor sei definiert durch $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - A^{-1}\mathbf{b}$. Dann gilt (mit der Konditionszahl $\kappa_2(A)$ bzgl. der euklidischen Norm)⁶

$$\|\mathbf{e}^{(k)}\|_A \leq \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1} \right)^k \cdot \|\mathbf{e}^{(0)}\|_A.$$

(b) A sei symmetrisch und positiv definit. $\mathbf{x}^{(k)}$ sei die k -te Iterierte des cg-Verfahrens mit dem Fehlervektor $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - A^{-1}\mathbf{b}$. Dann gilt⁷

$$\|\mathbf{e}^{(k)}\|_A \leq 2 \cdot \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^k \cdot \|\mathbf{e}^{(0)}\|_A.$$

Abschätzungen dieser Art lassen sich einsetzen, um Schranken für die Anzahl benötigter Iterationsschritte zu finden, z.B.⁸

(4.11) Folgerung: Bei

$$k \geq \frac{1}{2} \sqrt{\kappa_2(A)} \cdot \ln(2/\epsilon)$$

⁶vgl. A. Meister, Numerik linearer Gleichungssysteme, Satz 4.54

⁷vgl. P.Deuffhard und A. Hohmann, Numerische Mathematik I, Satz 8.17

⁸vgl. P.Deuffhard/A. Hohmann, Numerische Mathematik I, Folgerung 8.18

Iterationen des *cg*-Verfahrens wird der Approximationsfehler (bzgl. der Norm $\|\cdot\|_A$) gegenüber dem Startfehler um mindestens den Faktor ϵ verringert, d.h.

$$\|\mathbf{e}^{(k)}\|_A \leq \epsilon \cdot \|\mathbf{e}^{(0)}\|_A.$$

5 QR-Zerlegungen

Für eine Reihe von Anwendungsproblemen (Eigenwert-, lineare Ausgleichsprobleme) sind *QR*-Zerlegungen numerisch günstiger als die *LU*-Zerlegung des Gaußschen Eliminationsverfahrens. Hierbei sind R eine obere Dreiecksmatrix und Q eine Orthogonalmatrix. Die Überlegenheit dieser Zerlegung bezüglich der numerischen Stabilität hängen eng mit den spezifischen Eigenschaften orthogonaler Matrizen zusammen; so lässt die Multiplikation mit einer Orthogonalmatrix die (euklidische) Länge von und die Winkel zwischen Vektoren invariant. Wegen

$$\langle \mathbf{x}, \mathbf{y} \rangle = \langle Q^T Q \mathbf{x}, \mathbf{y} \rangle = \langle Q \mathbf{x}, Q \mathbf{y} \rangle$$

gilt nämlich

$$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\langle Q \mathbf{x}, Q \mathbf{x} \rangle} = \|Q \mathbf{x}\|_2$$

sowie für den Winkel $\angle(\mathbf{x}, \mathbf{y})$ zwischen \mathbf{x} und \mathbf{y}

$$|\cos \angle(\mathbf{x}, \mathbf{y})| = \frac{|\langle \mathbf{x}, \mathbf{y} \rangle|}{\|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2} = \frac{|\langle Q \mathbf{x}, Q \mathbf{y} \rangle|}{\|Q \mathbf{x}\|_2 \cdot \|Q \mathbf{y}\|_2} = |\cos \angle(Q \mathbf{x}, Q \mathbf{y})|.$$

Zudem gilt: Ist die *QR*-Zerlegung von A gegeben, so lässt sich die Lösung von $Ax = B$ auf die Lösung eines gestaffelten Systems zurückführen. Es gilt nämlich

$$Ax = b \iff Rx = Q^T b.$$

5.1 Zur Existenz und Eindeutigkeit von *QR*-Zerlegungen

Die Aufgabe besteht darin, eine Matrix Q zu konstruieren mit folgenden Eigenschaften:

- (i) Die Spalten $\mathbf{q}_{(1)}, \dots, \mathbf{q}_{(n)}$ von Q sind Einheitsvektoren und paarweise orthogonal.
- (ii) A lässt sich darstellen als Produkt von Q mit einer geeigneten Dreiecksmatrix R .

Aus (ii) lässt sich unmittelbar ableiten, dass sich die j -te Spalte $\mathbf{a}_{(j)}$ von A als Linearkombination der ersten j Spalten von Q schreiben lässt:

$$\mathbf{a}_{(j)} = \sum_{i=1}^j r_{ij} \mathbf{q}_{(i)}. \quad (5.1)$$

Wegen der Orthogonalität der $\mathbf{q}_{(j)}$ ist

$$r_{ij} = \langle \mathbf{a}_{(j)}, \mathbf{q}_{(i)} \rangle. \quad (5.2)$$

Mit den Hilfsmitteln der linearen Algebra kann man leicht zeigen, dass sich dann auch die j -te Spalte $\mathbf{q}_{(j)}$ von Q als Linearkombination

$$\mathbf{q}_{(j)} = \sum_{i=1}^j s_{ij} \mathbf{a}_{(i)}$$

mit geeigneten Koeffizienten s_{ij} schreiben lassen muss. Damit spannen die Vektoren $\mathbf{a}_{(1)}, \dots, \mathbf{a}_{(j)}$ und $\mathbf{q}_{(1)}, \dots, \mathbf{q}_{(j)}$ den selben Vektorraum auf, es gilt also

$$\text{span}(\mathbf{a}_{(i)} | i = 1, \dots, j) = \text{span}(\mathbf{q}_{(i)} | i = 1, \dots, j)$$

Aus dieser Überlegung und der Vorschrift (i) lässt sich nun das *Gram-Schmidt-Verfahren* zur Konstruktion der $\mathbf{q}_{(j)}$ herleiten.

Zunächst erkennt man, dass

$$\mathbf{q}_{(1)} = \pm \frac{1}{\|\mathbf{a}_{(1)}\|} \cdot \mathbf{a}_{(1)}$$

(das Vorzeichen kann nach Belieben festgelegt werden) sowie

$$r_{11} = \pm \|\mathbf{a}_{(1)}\|.$$

Sind $\mathbf{q}_{(1)}, \dots, \mathbf{q}_{(k)}$ bestimmt, so werden $\mathbf{q}_{(k+1)}$ und die $(k+1)$ -te Spalte $r_{(k+1)}$ von R bestimmt durch die folgenden Schritte.

- (i) Wähle als Ansatz für den nicht-normierten Vektor

$$\mathbf{q}_{(k+1)}^0 := \mathbf{a}_{(k+1)} + \sum_{i=1}^k \sigma_{i,k+1} \mathbf{q}_{(i)}$$

und bestimme die $\sigma_{i,k+1}$ aus der Forderung der Orthogonalität von $\mathbf{q}_{(k+1)}^0$ zu $\mathbf{q}_{(j)}$, $i = 1, \dots, k$. Dies führt auf die k linearen Gleichungen für $\sigma_{i,k+1}$,

$$\langle \mathbf{a}_{(k+1)}, \mathbf{q}_{(j)} \rangle + \sum_{i=1}^k \sigma_{i,k+1} \cdot \langle \mathbf{q}_{(i)}, \mathbf{q}_{(j)} \rangle = 0, \quad j = 1, \dots, k.$$

Aus der Orthonormalität der \mathbf{q}_i folgt

$$\sigma_{j,k+1} = -\langle \mathbf{a}_{(k+1)}, \mathbf{q}_{(j)} \rangle = -r_{j,k+1} \quad (\text{vgl. (2.9)}).$$

- (ii) Bestimme $\lambda_{(k+1)} \in \mathbb{R}$ so, dass

$$\mathbf{q}_{(k+1)} := \lambda_{(k+1)} \cdot \mathbf{q}_{(k+1)}^0$$

die euklidische Norm 1 hat.

- (iii) Berechne

$$r_{k+1,k+1} := \langle \mathbf{a}_{(k+1)}, \mathbf{q}_{(k+1)} \rangle.$$

Die algorithmische Umsetzung dieser vier Schritte ist in Skizze 1 beschrieben.

5.2 Givens-Rotationen

Givens-Rotationen beruhen auf der Beobachtung, dass Drehungen im \mathbb{R}^2 durch orthogonale Matrizen beschrieben werden. Die Multiplikation eines Vektors $\mathbf{x} \in \mathbb{R}^2$ mit der *Drehmatrix*

$$\Omega := \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix}$$

	Für $k = 1, \dots, n$
$S(i(a))$	[Berechnung der r_{jk} , $j = 1(1)k - 1$] Für $j = 1(1)k - 1$ $r_{jk} = 0$ Für $i = 1(1)n$ $r_{jk} = r_{jk} + a_{ik}q_{ij}$
$S(i(b))$	[Berechnung von $\mathbf{q}_{(k)}^0$] Für $j = 1(1)n$ $q_{jk} = a_{jk}$ Für $i = 1(1)k - 1$ $q_{jk} = q_{jk} - r_{ik}q_{ji}$
$S(ii)$	[Berechnung von $\lambda_{(k)}$ und $\mathbf{q}_{(k)}$] $\lambda = 0$ Für $j = 1(1)n$ $\lambda = \lambda + q_{jk}q_{jk}$ $\lambda = 1/\sqrt{\lambda}$ Für $j = 1(1)n$ $q_{jk} = \lambda q_{jk}$
$S(iii)$	[Berechnung von r_{kk}] $r_{kk} := 0$ Für $j = 1(1)n$ $r_{kk} = r_{kk} + a_{jk}q_{jk}$

Skizze 1: QR-Zerlegung nach dem Gram-Schmidt-Verfahren

bewirkt eine Drehung von \mathbf{x} im Uhrzeigersinn um den Winkel ϕ . Entsprechend bewirkt die Matrix

$$\Omega_{kl} := \begin{pmatrix} I & & & \\ & c & s & \\ & & I & \\ & -s & c & \\ & & & I \end{pmatrix} \begin{matrix} \leftarrow k \\ \\ \leftarrow l \\ \end{matrix}$$

mit $c^2 + s^2 = 1$ eine Drehung in der kl -Ebene um den Winkel ϕ , welcher gegeben ist durch

$$\cos(\phi) = c, \quad \sin(\phi) = s.$$

Solche Drehmatrizen können nun wie bei der Gauß-Elimination dazu benutzt werden, um sukzessive alle Subdiagonalelemente in Nullelemente umzuwandeln. Betrachten wir die Wirkung von Ω_{kl} auf die k -te und die l -te Komponente von \mathbf{x} (alle anderen Komponenten bleiben unverändert). Es ist

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x_k \\ x_l \end{pmatrix} = \begin{pmatrix} cx_k + sx_l \\ -sx_k + cx_l \end{pmatrix}.$$

Um die zweite Komponente auf Null zu transformieren, muss $c/s = x_k/x_l$ gewählt werden. Zusammen mit der Forderung $c^2 + s^2 = 1$ folgt

$$s = \frac{\pm x_k}{\sqrt{x_k^2 + x_l^2}}, \quad c = \frac{\pm x_l}{\sqrt{x_k^2 + x_l^2}}$$

Zur Vermeidung eines Exponentenüberlaufs berechnet man c und s am günstigsten wie folgt.

$$\text{Falls } |x_l| > |x_k|: \quad \tau := x_k/x_l, \quad s := 1/\sqrt{1 + \tau^2}, \quad c := s\tau;$$

$$\text{Falls } |x_l| \leq |x_k|: \quad \tau := x_l/x_k, \quad c := 1/\sqrt{1 + \tau^2}, \quad s := s\tau.$$

Durch Multiplikation von A mit einer Matrix der Form

$$\Omega := \Omega_{n-1,n} \cdot (\Omega_{n-2,n} \Omega_{n-2,n-1}) \cdots (\Omega_{n1} \cdots \Omega_{21}),$$

wobei die Koeffizienten von Ω_{kl} wie oben bestimmt werden, wird A in eine obere Dreiecksmatrix R verwandelt. Umgekehrt gilt

$$A = QR$$

$Q = I$ [Einheitsmatrix]

Für $k = 1, \dots, n-1$

Für $j = k+1(1)n$

Falls $a_{jk} \neq 0$:

Falls $|a_{jk}| > |a_{kk}|$:

$$\tau = a_{kk}/a_{jk} \quad s = 1/\sqrt{1+\tau^2} \quad c = s\tau$$

Andernfalls:

$$\tau = a_{jk}/a_{kk} \quad c = 1/\sqrt{1+\tau^2} \quad s = c\tau$$

$$x_1 = a_{kk} \quad x_2 = a_{jk}$$

$$a_{kk} = cx_1 + sx_2 \quad a_{jk} = -sx_1 + cx_2$$

$$x_1 = a_{kj} \quad x_2 = a_{jj}$$

$$a_{kj} = cx_1 + sx_2 \quad a_{jj} = -sx_1 + cx_2$$

$$x_1 = q_{kk} \quad x_2 = q_{kj}$$

$$q_{kk} = cx_1 + sx_2 \quad a_{kj} = -sx_1 + cx_2$$

$$x_1 = q_{jk} \quad x_2 = q_{jj}$$

$$q_{jk} = cx_1 + sx_2 \quad q_{jj} = -sx_1 + cx_2$$

Skizze 2: QR-Zerlegung mit Hilfe von Givens-Rotationen

mit

$$Q = \Omega^{-1} = (\Omega_{21}^{-1} \cdots \Omega_{n1}^{-1}) \cdots (\Omega_{n-2,n-1}^{-1} \Omega_{n-2,n}^{-1}) \cdot \Omega_{n-1,n}^{-1}.$$

Hierzu beachte man, dass die Inversen der Drehmatrizen leicht berechnet werden können durch

$$\begin{pmatrix} I & & & \\ & c & s & \\ & & I & \\ & -s & c & \\ & & & I \end{pmatrix}^{-1} = \begin{pmatrix} I & & & \\ & c & -s & \\ & & I & \\ & s & c & \\ & & & I \end{pmatrix}.$$

Ein Algorithmus zur Berechnung von Q und zur Umwandlung von A in R ist in Skizze 2 dargestellt. Die QR -Zerlegung mittels Givens-Rotationen benötigt bei vollbesetzter Matrix $\sim 4n^3/3$ Multiplikationen gegenüber $n^3/3$ bei der Gauß-Elimination. Günstiger fällt der Vergleich bei dünn-besetzten Matrizen aus; z.B. sind für eine Hessenberg-Matrix nur $n - 1$ Givens-Rotationen nötig.

5.3 Householder-Reflexionen

Eine weitere Gruppe von orthogonalen Transformationen sind Spiegelungen. Es sei $\mathbf{v} \in \mathbb{R}^n$ ein Einheitsvektor und \mathbf{E} die auf \mathbf{v} senkrechte Ebene. Jeder Vektor \mathbf{x} kann zerlegt werden in den Anteil

$$\mathbf{x}_\perp := \langle \mathbf{x}, \mathbf{v} \rangle \cdot \mathbf{v},$$

welcher senkrecht auf \mathbf{E} steht, und den Anteil

$$\mathbf{x}_\parallel := \mathbf{x} - \mathbf{x}_\perp,$$

welcher innerhalb \mathbf{E} verläuft. Die Spiegelung von \mathbf{x} an der Ebene \mathbf{E} ist gegeben durch die Abbildung

$$\mathbf{x} \mapsto \mathbf{x} - 2\mathbf{x}_\perp.$$

Wegen

$$\langle \mathbf{x}, \mathbf{v} \rangle \cdot \mathbf{v} = \mathbf{v}(\mathbf{v}^T \mathbf{x}) = (\mathbf{v}\mathbf{v}^T)\mathbf{x}$$

wird sie beschrieben durch die Matrix (*Householder-Reflexion*)

$$Q = I - 2\mathbf{v}\mathbf{v}^T.$$

Man überzeugt sich leicht von den folgenden Eigenschaften von Q .

- (5.1) Lemma:** (i) Q ist symmetrisch.
(ii) Q ist orthogonal.
(iii) Q ist eine Involution, d.h. $Q^2 = I$.

Die Anwendung von Q auf einen Vektor \mathbf{y} ergibt

$$\mathbf{y} \mapsto Q\mathbf{y} = (I - 2\mathbf{v}\mathbf{v}^T)\mathbf{y} = \mathbf{y} - 2\langle \mathbf{v}, \mathbf{y} \rangle \cdot \mathbf{v}.$$

Soll \mathbf{y} auf ein Vielfaches des ersten Einheitsvektors \mathbf{e}_1 gespiegelt werden, d.h.

$$\alpha\mathbf{e}_1 = \mathbf{y} - 2\langle \mathbf{v}, \mathbf{y} \rangle \cdot \mathbf{v},$$

so folgt

$$|\alpha| = \|\mathbf{y}\|_2 \quad \text{und} \quad \mathbf{v} \in \text{span}(\mathbf{y} - \alpha\mathbf{e}_1),$$

und die Spiegelungsebene ist gegeben durch den (nicht-normierten) Normalenvektor

$$\mathbf{v} = \mathbf{y} \pm \|\mathbf{y}\|_2 \cdot \mathbf{e}_1.$$

Um Auslöschung bei der Berechnung von \mathbf{v} zu vermeiden und damit Rundungsfehler möglichst gering zu halten, ist es ratsam, den Normalenvektor zu definieren durch

$$\mathbf{v} = \mathbf{y} + \text{sign}(y_1) \cdot \|\mathbf{y}\|_2 \cdot \mathbf{e}_1.$$

Für die QR -Zerlegung mittels Householder-Reflexionen werden nacheinander für die Spalten $1 \dots n-1$ die Elemente unterhalb des Diagonalelements auf 0 transformiert. Dies geschieht jeweils durch eine einzige Spiegelung. Wurde beispielsweise A in k Schritten transformiert auf die Gestalt

$$A^{(k)} = \begin{pmatrix} * & \cdots & \cdots & \cdots & * \\ & \ddots & & & \vdots \\ & & * & \cdots & * \\ & & 0 & & \\ & & \vdots & T^{(k+1)} & \\ & & 0 & & \end{pmatrix}$$

(mit einer i.a. vollbesetzten $(n-k) \times (n-k)$ -Matrix $T^{(k+1)}$), so wird im nächsten Schritt $A^{(k)}$ multipliziert mit einer orthogonalen Matrix der Form

$$Q_{k+1} = \left(\begin{array}{c|c} I_k & 0 \\ \hline 0 & \tilde{Q}_{k+1} \end{array} \right)$$

wobei \tilde{Q}_{k+1} so gewählt ist, dass in \mathbb{R}^{n-k} der erste Spaltenvektor von $T^{(k+1)}$ auf den ersten Einheitsvektor transformiert wird.

6 Eigenwert- und Ausgleichsprobleme

Im folgenden setzen wir voraus, dass A n linear unabhängige Eigenvektoren $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n)}$ zu den Eigenwerten $\lambda_1, \dots, \lambda_n$ besitzt. Gesucht sind Verfahren zur Bestimmung eines oder mehrerer Eigenwerte (und zugehöriger Eigenvektoren).

6.1 Vektoriterationen

A – Das Iterationsverfahren nach von Mises. Mit diesem Verfahren kann der betragsgrößte Eigenwert bestimmt werden. Vorausgesetzt sei, dass dieser Eigenwert eindeutig bestimmt sei. Denken wir uns die Eigenwerte der Größe nach sortiert, so soll also gelten

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$$

Wir wählen einen beliebigen Startvektor $\mathbf{x}^{(0)}$ und bestimmen den führenden Eigenwert samt zugehörigem Eigenvektor durch fortgesetzte Anwendung der Matrix A :

$$\mathbf{x}^{(k+1)} := A\mathbf{x}^{(k)}.$$

Hierzu beobachten wir: Da die Eigenvektoren linear unabhängig sind, bilden sie eine Basis des \mathbb{R}^n . Damit lässt sich $\mathbf{x}^{(0)}$ als Linearkombination schreiben:

$$\mathbf{x}^{(0)} = c_1 \mathbf{v}^{(1)} + \dots + c_n \mathbf{v}^{(n)}.$$

Durch Induktion folgt dann

$$\mathbf{x}^{(k)} = c_1 \lambda_1^k \mathbf{v}^{(1)} + \dots + c_n \lambda_n^k \mathbf{v}^{(n)}.$$

Für große k dominiert aufgrund der Voraussetzungen der erste Summand. Für die *normierte* k -te Iterierte

$$\mathbf{y}^{(k)} := \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|}$$

gilt

(6.1) Satz: Ist $c_1 \neq 0$, so konvergiert $\mathbf{y}^{(k)}$ für $k \rightarrow \infty$ gegen einen normierten Eigenvektor zum Eigenwert λ_1 .

Beweis: Aus den Voraussetzungen folgt

$$\mathbf{x}^{(k)} = c_1 \lambda_1^k \mathbf{v}^{(1)} + \cdots + c_n \lambda_n^k \mathbf{v}^{(n)} = c_1 \lambda_1^k \underbrace{\left(\mathbf{v}^{(1)} + \sum_{i=2}^n \frac{c_i}{c_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k \mathbf{v}^{(n)} \right)}_{=: \mathbf{z}_k}.$$

Wegen $|\lambda_i/\lambda_1| \leq |\lambda_2/\lambda_1| < 1$ konvergiert \mathbf{z}_k gegen $\mathbf{v}^{(1)}$ und es gilt

$$\mathbf{y}^{(k)} = \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|} = \pm \frac{\mathbf{z}^{(k)}}{\|\mathbf{z}^{(k)}\|} \longrightarrow \pm \frac{\mathbf{v}^{(1)}}{\|\mathbf{v}^{(1)}\|}. \quad \square$$

(6.2) Bemerkung: Nachteile des oben aufgeführten Verfahrens sind:

- (i) Berechnet werden können nur die betragsgrößten Eigenwerte.
- (ii) Die Konvergenzgeschwindigkeit beträgt $|\lambda_2/\lambda_1|$, ist also sehr gering, falls $|\lambda_1| \approx |\lambda_2|$.

B – Inverse Vektoriteration. Zur Abschwächung des Nachteils (2.13)(i) kann das Verfahren leicht modifiziert werden.

- (a) Der *betragskleinste* Eigenwert von A ist der *betragsgrößte* Eigenwert von A^{-1} . Dieser kann mit den selben Argumenten wie oben bestimmt werden durch die Vektoriteration

$$\mathbf{x}^{(k+1)} := A^{-1}\mathbf{x}^{(k)}$$

also durch iterative Lösung des linearen Gleichungssystems

$$A\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}.$$

- (b) Ist (durch ungenaue Berechnungen, Messungen oder andere Informationen) bekannt, dass sich in der Nähe des Wertes ξ ein Eigenwert $\bar{\lambda}$ befindet, so kann dieser wie oben bestimmt werden. Ist nämlich ξ hinreichend nahe bei $\bar{\lambda}$, so ist $\bar{\lambda}$ der *betragskleinste* Eigenwert von $(A - \xi \cdot I)$ und kann ermittelt werden durch die Vektoriteration

$$(A - \xi \cdot I)\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}.$$

6.2 Ein QR-Algorithmus für symmetrische EW-Probleme

Das Ziel ist es nun, simultan *alle* Eigenwerte der Matrix A zu bestimmen. Wir müssen im Folgenden voraussetzen, dass A symmetrisch ist. Die Eigenwertbestimmung erfolgt in zwei Schritten. In einem ersten Schritt wird A durch Ähnlichkeitstransformationen “möglichst nahe“ an eine Diagonalgestalt gebracht. (Wäre eine zu A ähnliche Diagonalmatrix bekannt, so wären gleichzeitig auch alle Eigenwerte von A bekannt.) Konkret wird A mit Hilfe von Householder-Transformationen auf Tridiagonalgestalt gebracht. Im zweiten Schritt hilft uns eine QR -Zerlegung der Tridiagonalmatrix, mittels eines Iterationsverfahrens alle Eigenwerte und Eigenvektoren zu bestimmen.

Erster Schritt: Tridiagonalisierung von A

Im Folgenden bezeichnen wir mit $a^{(1)}$ die erste Spalte von A . Wegen der Symmetrie von A ist die erste Zeile von A die Transponierte von $a^{(1)}$. Wollen wir $a^{(1)}$ in einen Vektor der Besetzungsstruktur $(*, *, 0, \dots, 0)^T$ verwandeln, so können wir dies mittels

Householder-Reflexion mit einer Transformationsmatrix der Gestalt

$$P_1 = \left(\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \overline{P}_1 & \\ 0 & & & \end{array} \right)$$

tun. In diesem Fall ändert sich die erste Zeile von A nicht. Multiplizieren wir nun die so erhaltene Matrix mit der transponierten Transformationsmatrix P_1^T von rechts, so verändert sich die erste Zeile in der selben Weise, wie sich vorher die erste Spalte verändert hat. Wir erhalten also folgendes Muster:

$$A = \left(\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline a_{21} & * & \cdots & * \\ \vdots & \vdots & & \vdots \\ a_{n1} & * & \cdots & * \end{array} \right) \xrightarrow{P_1 \cdot} \left(\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline * & * & \cdots & * \\ 0 & \vdots & & \vdots \\ \vdots & * & \cdots & * \end{array} \right) \xrightarrow{\cdot P_1^T} \left(\begin{array}{c|ccc} a_{11} & * & 0 & \cdots \\ \hline * & * & \cdots & * \\ 0 & \vdots & & \vdots \\ \vdots & * & \cdots & * \end{array} \right)$$

Die so entstehende Matrix ist offensichtlich wieder symmetrisch. Ebenso können wir mit den weiteren Spalten und Zeilen vorgehen. So verwenden wir für die Transformation der zweiten Spalte eine Matrix der Form

$$P_2 = \left(\begin{array}{c|ccc} I_2 & & & \\ \hline & & & \\ & & \overline{P}_2 & \end{array} \right)$$

sowie deren Transponierte zur Modifikation der zweiten Zeile. Schließlich erhalten wir eine symmetrische Tridiagonalmatrix T der Form $T = PAP^T$ mit einer Orthogonalmatrix P , welche sich als Produkt von Householder-Reflexionen darstellen lässt: $P = P_{n-2} \cdots P_1$. A und T besitzen die selben Eigenwerte.

Zweiter Schritt: QR-Schritt

Startend mit der Matrix

$$A_0 := A$$

kann nun folgendes Iterationsschema definiert werden.

(6.3) Iterationsschema: Gegeben sei die Tridiagonalmatrix A_k .

(i) Bestimme mittels Givens-Rotationen die QR -Zerlegung von A_k :

$$A_k =: Q_k R_k.$$

(ii) Bestimme die nächste Iterierte durch

$$A_{k+1} := R_k Q_k.$$

Wie die folgenden Bemerkungen zeigen, behalten die Matrizen A_k wichtige Eigenschaften der Ausgangsmatrix A bei.

(6.4) Bemerkungen: (a) A_k und A_{k+1} sind ähnlich, denn:

$$A_k = Q_k R_k = Q_k (R_k Q_k) Q_k^T = Q_k A_{k+1} Q_k^T. \quad (6.1)$$

Es folgt, dass A zu allen A_k ähnlich ist.

(b) Alle A_k sind symmetrisch, denn durch Induktion folgt: ist A_k symmetrisch, so ist

$$(A_{k+1})^T = (A_{k+1})^T Q_k^T Q_k = Q_k^T R_k^T Q_k^T Q_k = Q_k^T A_k^T Q_k = Q_k^T A_k Q_k = A_{k+1}$$

(nach (2.10)).

(c) Alle A_k sind Tridiagonalmatrizen.

Beweis von (c): A_k sei symmetrisch und tridiagonal. A_{k+1} kann mittels Givens-Rotationen wie folgt konstruiert werden. Die bei der QR -Zerlegung von A_k berechnete orthogonale Matrix Q_k ist ein Produkt von Givens-Rotationen:

$$Q_k = \Omega_{n-1,n} \cdots \Omega_{12}$$

und ist tridiagonal. Damit hat $R_k = Q_k^T A_k$ die folgende Besetzungsstruktur.

$$R_k = \begin{pmatrix} * & * & * & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & * & \\ & & & \ddots & * & \\ & & & & * & \\ & & & & & * \end{pmatrix}$$

Außerdem ist nach (2.10) A von der Form

$$A = QA_kQ^T$$

mit einer geeigneten orthogonalen Matrix Q . Q enthält für $k \rightarrow \infty$ die Eigenvektoren, denn allgemein gilt: Ist A ähnlich zu einer Diagonalmatrix $D = \text{diag}(d_{11}, \dots, d_{nn})$ mit einer Ähnlichkeitsmatrix T , d.h.

$$A = TDT^{-1},$$

so ist die i -te Spalte von T ein Eigenvektor von A zum Eigenwert d_{ii} .

Dem zufolge müssen die Spalten von Q Approximationen der Eigenvektoren von A sein. Hieraus ergibt sich der folgende Algorithmus zur Approximation der Eigenwerte und Eigenvektoren.

(6.6) Algorithmus (symm. EW-Problem):

- (i) Wandle mittels Householder-Reflexionen A in eine Tridiagonalmatrix T um:

$$A \longrightarrow T = PAP^T.$$

- (ii) Bestimme mittels Givens-Rotationen eine Orthogonalmatrix Ω derart, dass

$$\Lambda \approx \Omega T \Omega^T = (\Omega P)A(\Omega P)^T.$$

Dann enthält Λ die Eigenwerte von A und die Spalten von $\Omega \cdot P$ sind Approximationen der Eigenvektoren.

6.3 Lineare Ausgleichsprobleme

A – Normalgleichungen

Eine Anwendung von QR -Zerlegungen ist ihre numerisch stabile Lösung linearer Ausgleichsprobleme. Wir geben zunächst ein einfaches Beispiel.

(6.7) Beispiel: Durch die Messpunkte (x_i, f_i) , $i = 1, \dots, 5$, soll eine Ausgleichsgerade $f(x)$ gelegt werden. Die Messpunkte seien $(0.8, 1.2)$, $(2.4, 2.1)$, $(3.0, 1.6)$, $(3.7, 2.3)$ und

(5.1, 2.4). Der Ansatz

$$f(x) = ax + b$$

führt auf das überbestimmte lineare Gleichungssystem für die 2 Unbekannten a und b ,

$$\begin{pmatrix} 0.8 & 1 \\ 2.4 & 1 \\ 3.0 & 1 \\ 3.7 & 1 \\ 5.1 & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1.2 \\ 2.1 \\ 1.6 \\ 2.3 \\ 2.4 \end{pmatrix}.$$

Eine Möglichkeit, dieses System in ein reguläres Gleichungssystem überzuführen ist die Multiplikation (von links) mit der Transponierte A^T der Systemmatrix. Dies führt auf die *Normalgleichungen*

$$\begin{pmatrix} 55.1 & 15 \\ 15 & 5 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 31.55 \\ 9.6 \end{pmatrix}$$

mit der Lösung $(a, b)^T = (0.272, 1.103)^T$.

Gegeben sei das überbestimmte Gleichungssystem

$$A\mathbf{x} = \mathbf{b}$$

mit der $m \times n$ -Matrix A ($m > n$). Im Allgemeinen besitzt dieses System keine Lösung; es gibt aber in der Regel einen Näherungsvektor $\tilde{\mathbf{x}}$, für welchen der Fehler $\|A\tilde{\mathbf{x}} - \mathbf{b}\|_2$ minimiert wird. ($\tilde{\mathbf{x}}$ heißt dann *Lösung des Ausgleichsproblems*.) Dieser ergibt sich nach der *Methode der kleinsten Quadrate* durch Aufstellen und Lösen der *Normalgleichungen*

$$A^T A\mathbf{x} = A^T \mathbf{b}.$$

(6.8) Satz: $A \in \mathbb{R}^{m \times n}$ ($m > n$) habe maximalen Rang n . Sei $\mathbf{b} \in \mathbb{R}^m$. $\mathbf{x} \in \mathbb{R}^n$ minimiert genau dann das Funktional $\|\mathbf{b} - A\mathbf{x}\|_2$, wenn \mathbf{x} Lösung der Normalgleichung $A^T A\mathbf{x} = A^T \mathbf{b}$ ist.

Beweis: Die Matrix $A^T A$ ist positiv definit, also regulär. Damit hat die Normalgleichung

genau eine Lösung. Definiere das Funktional $\Phi(x) = 0.5\|b - Ax\|_2^2$. Wegen $\Phi(x) \rightarrow \infty$ für $\|x\| \rightarrow \infty$ besitzt Φ (mindestens) ein globales Minimum. Die Richtungsableitung von Φ ist

$$\begin{aligned} \frac{\partial}{\partial v} \Phi(x) &= \lim_{t \rightarrow 0} \frac{\Phi(x + tv) - \Phi(x)}{t} \\ &= \lim_{t \rightarrow 0} \frac{1}{2t} \left[(b - Ax - tAv)^T (b - Ax - tAv) - (b - Ax)^T (b - Ax) \right] \\ &= (Ax - b)^T Av \end{aligned}$$

Damit ist $\nabla \Phi(x) = (Ax - b)^T A$. Eine notwendige Bedingung für das Minimum von Φ ist $\nabla \Phi(x) = 0$. \circ

Hat A den maximalen Rang n , so ist $A^T A$ symmetrisch und positiv definit. Damit kann das Gleichungssystem z.B. mit Hilfe der Cholesky-Zerlegung gelöst werden.

B – Zur Kondition des Ausgleichsproblems

Die relative Kondition κ eines Problems zu den Eingabedaten b und den Ausgabedaten x ist definiert als (kleinste) Schranke für die Abschätzung

$$\frac{\Delta x}{x} \leq \kappa \cdot \frac{\Delta b}{b}$$

Wir beginnen mit der Konditionsanalyse einer orthogonalen Projektion auf einen Unterraum.

(6.9) Lemma: Sei $P : \mathbb{R}^n \rightarrow V$ die orthogonale Projektion auf einen Unterraum V des \mathbb{R}^n . Für die Eingabe b bezeichne θ den Winkel zwischen b und V , d.h.

$$\sin(\theta) = \frac{\|b - Pb\|_2}{\|b\|_2}$$

Dann gilt für die relative Kondition der Berechnung von Pb

$$\kappa = \frac{1}{\sin(\theta)} \|P\|_2$$

Beweis: Nach dem Satz von Pythagoras ist

$$\|Pb\|_2^2 = \|b\|_2^2 - \|b - Pb\|_2^2$$

und daher

$$\frac{\|Pb\|_2^2}{\|b\|_2^2} = 1 - \sin^2(\theta) = \cos^2(\theta)$$

Es folgt

$$\frac{\|P\Delta b\|_2}{\|Pb\|_2} \cdot \frac{\|b\|_2}{\|\Delta b\|_2} = \frac{1}{\cos(\theta)} \cdot \frac{\|P\Delta b\|_2}{\|\Delta b\|_2} \leq \frac{1}{\cos(\theta)} \cdot \|P\|_2 \quad \circ$$

Die Kondition κ_2 einer Matrix $A \in \mathbb{R}^{m \times n}$ ist in Analogie zur Aussage über die Kondition regulärer Matrizen definiert durch

$$\kappa_2(A) = \frac{\max_{\|x\|_2=1} \|Ax\|_2}{\min_{\|x\|_2=1} \|Ax\|_2}$$

(6.10) Lemma: Für eine Matrix $A \in \mathbb{R}^{m \times n}$ von maximalem Rang $p = n$ gilt

$$\kappa_2(A^T A) = \kappa_2(A)^2$$

Beweis: Die Matrix $A^T A$ ist symmetrisch und positiv definit. Es gilt

$$\begin{aligned} \kappa_2(A)^2 &= \frac{\max_{\|x\|_2=1} \|Ax\|_2^2}{\min_{\|x\|_2=1} \|Ax\|_2^2} \\ &= \frac{\max_{\|x\|_2=1} \langle A^T A x, x \rangle}{\min_{\|x\|_2=1} \langle A^T A x, x \rangle} = \frac{\lambda_{\max}}{\lambda_{\min}} = \kappa_2(A^T A) \quad \circ \end{aligned}$$

Damit kommen wir zum zentralen Ergebnis.

(6.11) Satz: Sei $A \in \mathbb{R}^{m \times n}$, $m \geq n$, eine Matrix von vollem Spaltenrang, $b \in \mathbb{R}^m$ und $x \neq 0$ die eindeutige Lösung des linearen Ausgleichsproblems

$$\|b - Ax\|_2 = \min$$

θ bezeichne den Winkel zwischen b und dem Bildraum $R(A)$ von A , d.h.

$$\sin(\theta) = \frac{\|b - Ax\|_2}{\|b\|_2} = \frac{\|r\|_2}{\|b\|_2}$$

Dann gilt für die relative Kondition von x in der Euklidischen Norm

(a) bezüglich Störungen in b

$$\kappa \leq \frac{\kappa_2(A)}{\cos(\theta)}$$

(b) bezüglich Störungen in A

$$\kappa \leq \kappa_2(A) + \kappa_2(A)^2 \tan(\theta)$$

Beweis: (a) Die exakte Lösung der Normalgleichungen ist

$$x = (A^T A)^{-1} A^T b$$

und es ist

$$\frac{\|(A^T A)^{-1} A^T \Delta b\|_2}{\|(A^T A)^{-1} A^T b\|_2} \cdot \frac{\|b\|_2}{\|\Delta b\|_2} \leq \frac{\|A\|_2 \|(A^T A)^{-1} A^T\|_2 \|b\|_2}{\|A\|_2 \|x\|_2}$$

Man rechnet leicht nach, dass für eine Matrix A mit vollem Spaltenrang gilt

$$\kappa_2(A) = \frac{\max_{\|x\|_2=1} \|Ax\|_2}{\min_{\|x\|_2=1} \|Ax\|_2} = \|A\|_2 \|(A^T A)^{-1} A^T\|_2$$

Die Aussage folgt nun wie in Lemma (6.9).

(b) Wir betrachten eine kleine Störung $\Delta A = t \cdot C$. Für kleine t erhalten wir die gestörte Lösung näherungsweise durch

$$x + \Delta x = \underbrace{((A + tC)^T (A + tC))^{-1} (A + tC)^T}_{=: \Phi(t)} b = x + \Phi'(0)b$$

Aus

$$(A + tC)^T (A + tC) \Phi(t) = (A + tC)^T$$

folgt $\Phi'(0)$. Rest: Übung. \circ

C – Lösung mittels QR-Zerlegung

Man überlegt sich leicht, dass die Berechnung von $A^T A$ einen Aufwand von $\mathcal{O}(n^2 m)$ Rechenoperationen erfordert. Die Cholesky-Zerlegung erfordert etwa $\mathcal{O}(n^3)$ Operationen.

Ist $m \gg n$, so liegt der Hauptaufwand bei der Berechnung von $A^T A$. Hinzu kommt, dass sich bei der Gauß-Elimination die Kondition des Ausgangsproblems deutlich verschlechtern kann. Diese Gründe motivieren, warum man versucht, Lösungen des Ausgleichsproblems auf andere Weise zu konstruieren. Als Lösungsmethoden bieten sich wieder QR -Zerlegungen an. Das Ziel ist hierbei, A mit Hilfe einer orthogonalen Transformation Q auf die folgende Form zu transformieren:

$$Q^T A = \begin{pmatrix} * & \cdots & * \\ & \ddots & \vdots \\ & & * \end{pmatrix} = \begin{pmatrix} R \\ 0 \end{pmatrix} \quad (6.2)$$

mit einer oberen $n \times n$ -Dreiecksmatrix R . Die Begründung für diesen Schritt liefert der folgende Satz.

(6.12) Satz: Die $m \times n$ -Matrix A habe den maximalen Rang n ; Q sei eine orthogonale $m \times m$ -Matrix derart, dass $Q^T A$ die Form (2.11) hat. Dann ist die Lösung $\tilde{\mathbf{x}}$ des linearen Ausgleichsproblems

$$\|A\mathbf{x} - \mathbf{b}\|_2 \stackrel{!}{=} \min$$

gegeben als $\tilde{\mathbf{x}} = R^{-1}\mathbf{b}_1$, wobei \mathbf{b}_1 der Vektor der ersten n Komponenten von $Q^T\mathbf{b}$ ist.

Beweis: Für den Fehler $\|A\mathbf{x} - \mathbf{b}\|_2$ gilt mit $Q^T\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2)^T$

$$\|A\mathbf{x} - \mathbf{b}\|_2^2 = \|Q^T(A\mathbf{x} - \mathbf{b})\|_2^2 = \left\| \begin{pmatrix} R\mathbf{x} - \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \right\|_2^2 = \|R\mathbf{x} - \mathbf{b}_1\|_2^2 + \|\mathbf{b}_2\|_2^2 \geq \|\mathbf{b}_2\|_2^2. \quad \square$$

Eine Möglichkeit zur Lösung des Ausgleichsproblems besteht in der Zerlegung von A in der Form (2.11) mittels Householder-Transformationen.

6.4 Singulärwertzerlegungen und Pseudoinverse

A – Singulärwerte

Es ist bekannt, dass symmetrische Matrizen ähnlich zu Diagonalmatrizen sind; genauer: ist A symmetrisch, so existieren eine reguläre Matrix T (welche als Orthogonalmatrix

gewählt werden kann) und eine Diagonalmatrix $\Lambda = \text{diag}(\lambda_i, i = 1, \dots, n)$ derart, dass $A = T\Lambda T^{-1}$. Die Diagonalelemente von Λ sind gerade die Eigenwerte von A .

Dieses Ergebnis lässt sich nicht auf beliebige Matrizen übertragen. Dagegen lässt sich eine schwächere Aussage verallgemeinern, wie aus dem folgenden Satz hervorgeht.

(6.13) Satz: A sei eine reelle $m \times n$ -Matrix. Dann gibt es eine orthogonale $m \times m$ -Matrix U und eine orthogonale $n \times n$ -Matrix derart, dass

$$U^T A V = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_p), \quad p = \min(m, n)$$

mit nichtnegativen λ_i . Die Zerlegung kann so gewählt werden, dass

$$\lambda_1 \geq \dots \geq \lambda_p \geq 0.$$

Statt eines vollständigen **Beweises** wollen wir uns damit begnügen, den ersten Schritt der Diagonalisierung durchzuführen, also eine Darstellung

$$U^T A V = \left(\begin{array}{c|ccc} \lambda & 0 & \dots & 0 \\ \hline 0 & & & \\ \vdots & & B & \\ 0 & & & \end{array} \right)$$

zu finden. Es sei

$$\lambda := \|A\|_2 = \max_{\|\mathbf{v}\|_2=1} \|A\mathbf{v}\|_2.$$

Dieses Maximum wird angenommen (da die Menge der Einheitsvektoren beschränkt und abgeschlossen ist). Damit gibt es Einheitsvektoren $\mathbf{v} \in \mathbb{R}^n$ und $\mathbf{u} \in \mathbb{R}^m$ mit $A\mathbf{v} = \lambda\mathbf{u}$. Wir erweitern $\{\mathbf{v}\}$ zu einer Orthonormalbasis $\{\mathbf{v} = \mathbf{v}_{(1)}, \mathbf{v}_{(2)}, \dots, \mathbf{v}_{(n)}\}$ in \mathbb{R}^n und $\{\mathbf{u}\}$ zu einer Orthonormalbasis $\{\mathbf{u} = \mathbf{u}_{(1)}, \mathbf{u}_{(2)}, \dots, \mathbf{u}_{(m)}\}$ in \mathbb{R}^m und bilden die Orthogonalmatrizen $V := (\mathbf{v}_{(1)}, \dots, \mathbf{v}_{(n)})$ und $U := (\mathbf{u}_{(1)}, \dots, \mathbf{u}_{(m)})$. Definiere $A_1 := U^T A V$. Mit dem ersten Einheitsvektor \mathbf{e}_1 (in \mathbb{R}^n bzw. in \mathbb{R}^m) gilt

$$A_1 \mathbf{e}_1 = U^T A V \mathbf{e}_1 = U^T A \mathbf{v}_{(1)} = \lambda \cdot U^T \mathbf{u}_{(1)} = \lambda \cdot \mathbf{e}_{(1)}.$$

Damit ist die erste Spalte von A_1 gleich $\lambda \cdot \mathbf{e}_{(1)}$ und A_1 von der Form

$$A_1 = \left(\begin{array}{c|c} \lambda & \omega^T \\ \hline 0 & B \end{array} \right)$$

mit einem geeigneten Vektor $\omega \in \mathbb{R}^{n-1}$. Es bleibt zu zeigen, dass $\omega = 0$.

Da A_1 aus A durch Multiplikation mit Orthogonalmatrizen hervorgegangen ist, gilt $\|A_1\|_2 = \|A\|_2$. Außerdem ist

$$A_1 \begin{pmatrix} \lambda \\ \omega \end{pmatrix} = \begin{pmatrix} \lambda^2 + \omega^T \omega \\ * \end{pmatrix}$$

und damit

$$\left\| A_1 \begin{pmatrix} \lambda \\ \omega \end{pmatrix} \right\|_2^2 \geq (\lambda^2 + \omega^T \omega)^2 = (\lambda^2 + \|\omega\|_2^2)^2,$$

sowie

$$\left\| A \begin{pmatrix} \lambda \\ \omega \end{pmatrix} \right\|_2^2 \leq \lambda^2 \cdot \left\| \begin{pmatrix} \lambda \\ \omega \end{pmatrix} \right\|_2^2 = \lambda^2 \cdot (\lambda^2 + \|\omega\|_2^2).$$

Es folgt $\|\omega\|_2 = 0$, also $\omega = 0$. \square

Die im Satz beschriebene Zerlegung $U^T A V = \Lambda$ heißt **Singulärwertzerlegung** von A . Die Werte $\lambda_1, \dots, \lambda_p$ heißen die **Singulärwerte** von A . Sie sind (bis auf Vertauschung) eindeutig bestimmt. Wir wollen kurz auf die Bedeutung der Singulärwerte eingehen.

(6.14) Bemerkung: (a) Für die Spalten $\mathbf{u}_{(i)}$ von U und $\mathbf{v}_{(i)}$ von V gilt

$$A \mathbf{v}_{(i)} = \lambda_i \mathbf{u}_{(i)} \quad \text{und} \quad A^T \mathbf{u}_{(i)} = \lambda_i \mathbf{v}_{(i)}.$$

(b) Aus (a) folgt: $\mathbf{u}_{(i)}$ und $\mathbf{v}_{(i)}$ sind Eigenvektoren von $A^T A$ bzw. $A A^T$ zum EW λ_i^2 .

(c) Ist $\lambda_1 \geq \dots \geq \lambda_r > \lambda_{r+1} = \dots = \lambda_p = 0$, so ist $\text{rang}(A) = r$,

$$\text{kern}(A) = \text{span}(\mathbf{v}_{(r+1)}, \dots, \mathbf{v}_{(n)}) \quad \text{und} \quad \text{bild}(A) = \text{span}(\mathbf{u}_{(1)}, \dots, \mathbf{u}_{(r)}).$$

(d) Die euklidische Norm ist gleich dem größten Singulärwert, d.h.

$$\|A\|_2 = \lambda_1.$$

(e) Die Frobeniusnorm ist

$$\|A\|_F = \sqrt{\lambda_1^2 + \dots + \lambda_p^2}.$$

(f) Die Kondition bzgl. der Euklidischen Norm ist

$$\kappa_2(A) = \lambda_1 / \lambda_p.$$

B – Numerische Berechnung der Singulärwerte

Eine Grundidee zur *effizienten numerischen Berechnung* besteht zunächst darin, A in Bidiagonalform

$$PAQ = \begin{pmatrix} B \\ 0 \end{pmatrix} \quad (m \geq n) \quad \text{bzw.} \quad = \begin{pmatrix} B & 0 \end{pmatrix} \quad (m < n)$$

zu bringen (mit geeigneten Orthogonalmatrizen P und Q) und dann den QR -Algorithmus auf die symmetrische Tridiagonalmatrix $B^T B$ (bzw. BB^T) anzuwenden. Wir beschränken uns hier auf den Fall $m \geq n$.

(6.15) Lemma: Für jede Matrix $A \in \mathbb{R}^{m \times n}$ ($m \geq n$) existieren orthogonale Matrizen $P \in \mathbb{R}^{m \times m}$ und $Q \in \mathbb{R}^{n \times n}$ derart, dass

$$PAQ = \begin{pmatrix} B \\ 0 \end{pmatrix} \quad \text{mit} \quad B = \begin{pmatrix} * & * & & & \\ & \ddots & \ddots & & \\ & & \ddots & * & \\ & & & \ddots & * \end{pmatrix}$$

Beweis: Die Konstruktion erfolgt (ähnlich wie die Tridiagonalisierung einer symmetrischen Matrix, vgl. Abschnitt 6.2) durch Householder-Transformationen von links und rechts zur Elimination der Zeilen und Spalten. \circ

Führen wir nun den ersten Givens-Eliminationsschritt des QR -Algorithmus auf $A = B^T B$ aus,

$$A \rightarrow \Omega_{12} B^T B \Omega_{12}^T = \underbrace{(B \Omega_{12}^T)^T}_{\tilde{B}^T} \underbrace{B \Omega_{12}^T}_{\tilde{B}}$$

so erhalten wir für \tilde{B} die Matrix

$$\tilde{B} = B \Omega_{12}^T = \begin{bmatrix} * & * & & & \\ \oplus & * & * & & \\ & & \ddots & \ddots & \\ & & & \ddots & * \\ & & & & * \end{bmatrix}$$

C – Pseudoinverse

(6.17) Definition: Die *Pseudoinverse* einer Matrix $A \in \mathbb{R}^{m \times n}$ ist diejenige Matrix $A^+ \in \mathbb{R}^{n \times m}$, für die gilt: Für beliebige $b \in \mathbb{R}^m$ ist $x = A^+b$ der kleinste Vektor, welcher den Abstand $\|b - Ax\|$ minimiert, d.h.

$$A^+b \in (\text{kern}(A))^\perp \quad \text{und} \quad \|b - AA^+b\| = \min$$

Ohne Beweis¹⁰ geben wir folgende Eigenschaften an.

(6.18) Satz: Die Pseudoinverse ist eindeutig charakterisiert durch folgende vier Eigenschaften (“*Penrose-Axiome*”):

- (i) $(A^+A)^T = A^+A$
- (ii) $(AA^+)^T = AA^+$
- (iii) $A^+AA^+ = A^+$
- (iv) $AA^+A = A$

Mit Hilfe der Singulärwertzerlegung lässt sich die Pseudoinverse berechnen.

(6.19) Folgerung: Sei $U^TAV = \Sigma$ die Singulärwertzerlegung einer Matrix $A \in \mathbb{R}^{m \times n}$ mit $\text{Rang}(A) = p$ und

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_p, 0, \dots, 0) \in \mathbb{R}^{m \times n}$$

Dann ist die Pseudoinverse A^+ von A gegeben durch

$$A^+ = V\Sigma^+U^T \quad \text{mit} \quad \Sigma^+ = \text{diag}(\sigma_1^{-1}, \dots, \sigma_p^{-1}, 0, \dots, 0) \in \mathbb{R}^{n \times m}$$

Beweis: Offenbar ist

$$\begin{aligned} \Sigma\Sigma^+ &= \text{diag}(\underbrace{1, \dots, 1}_{p\text{-mal}}, 0, \dots, 0) \in \mathbb{R}^{m \times m} \\ \Sigma^+\Sigma &= \text{diag}(\underbrace{1, \dots, 1}_{p\text{-mal}}, 0, \dots, 0) \in \mathbb{R}^{n \times n} \end{aligned}$$

¹⁰vgl. Deuffhard/Hohmann, Satz 3.16

Damit rechnet man leicht nach, dass Σ^+ die Penrose-Axiome erfüllt und daher die Pseudoinverse von Σ ist. Hieraus folgt

$$\begin{aligned}
 A^+A &= V\Sigma^+U^T U\Sigma V^T = \underbrace{\text{diag}(1, \dots, 1, 0, \dots, 0)}_{n \times n} = (A^+A)^T \\
 AA^+ &= U\Sigma V^T V\Sigma^+U^T = \underbrace{\text{diag}(1, \dots, 1, 0, \dots, 0)}_{m \times m} = (AA^+)^T \\
 (A^+A)A^+ &= \text{diag}(1, \dots, 1, 0, \dots, 0)A^+ = V\text{diag}(1, \dots, 1, 0, \dots, 0)V^T V\Sigma^+U^T \\
 &= V\Sigma^+U^T = A^+ \\
 (AA^+)A &= \text{diag}(1, \dots, 1, 0, \dots, 0)A = U\text{diag}(1, \dots, 1, 0, \dots, 0)U^T U\Sigma V^T \\
 &= U\Sigma V^T = A \quad \circ
 \end{aligned}$$