
4 Lineare Gleichungssysteme

4.1 Problemstellung und Einführung

In diesem Kapitel betrachten wir direkte Verfahren zur Lösung von linearen Gleichungssystemen.

Lineares Gleichungssystem: Gesucht ist eine Lösung $x \in \mathbb{R}^n$ von

$$Ax = b \quad (4.1)$$

mit

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \in \mathbb{R}^{n \times n}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^n, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n. \quad (4.2)$$

Die hier besprochenen direkten Methoden liefern – rundungsfehlerfreie Rechnung vorausgesetzt – die Lösung von (4.1) in endlich vielen Rechenschritten. Bekanntlich ist (4.1) die Matrixschreibweise für

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i, \quad i = 1, \dots, n.$$

Lineare Gleichungssysteme treten in der Praxis als Hilfsproblem bei einer Vielzahl von Problemstellungen auf, z. B. bei der Lösung von Rand- und Randanfangswertaufgaben für gewöhnliche und partielle Differentialgleichungen (Schaltkreissimulation, elektromagnetische Felder, ...), in der Bildverarbeitung, usw. Schätzungen besagen, dass etwa 75% der Rechenzeit im technisch-wissenschaftlichen Bereich auf die Lösung von linearen Gleichungssystemen entfällt.

Wir erinnern zunächst an folgenden Sachverhalt.

Proposition 4.1.1. *Das lineare Gleichungssystem (4.1) hat eine Lösung genau dann, wenn gilt*

$$\text{rang}(A) = \text{rang}(A, b).$$

Hierbei ist bekanntlich für eine Matrix $B \in \mathbb{R}^{n \times m}$ der Rang definiert durch

$$\begin{aligned} \text{Rang}(B) &= \text{Maximalzahl } r \text{ der linear unabhängigen Zeilenvektoren} \\ &= \text{Maximalzahl } r \text{ der linear unabhängigen Spaltenvektoren.} \end{aligned}$$

Das lineare Gleichungssystem (4.1) hat eine eindeutige Lösung genau dann, wenn A invertierbar ist (oder gleichbedeutend: $\det(A) \neq 0$). Die eindeutige Lösung lautet dann

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

4.2 Das Gaußsche Eliminationsverfahren, Dreieckszerlegung einer Matrix

Das grundsätzliche Vorgehen der Gauß-Elimination ist aus der Linearen Algebra bekannt. Wir werden das Verfahren kurz wiederholen und zeigen, wie man daraus eine Dreieckszerlegung einer Matrix erhält. Zudem werden wir uns klarmachen, welchen Einfluss Rundungsfehler haben können und wie dieser Einfluss wirksam bekämpft werden kann.

Die Grundidee des Gaußschen Eliminationsverfahrens besteht darin, das Gleichungssystem (4.1) durch die elementaren Operationen

- Addition eines Vielfachen einer Gleichung zu einer anderen,
- Zeilenvertauschungen, d. h. Vertauschen von Gleichungen
- Spaltenvertauschungen, die einer Umnummerierung der Unbekannten entsprechen,

in ein Gleichungssystem der Form

$$Ry = c, \quad y_{\sigma_i} = x_i, \quad i = 1, \dots, n,$$

mit der durchgeführten Spaltenpermutation $(\sigma_1, \dots, \sigma_n)$ und einer oberen Dreiecksmatrix

$$R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ 0 & & r_{nn} \end{pmatrix}$$

zu überführen, das dieselben Lösungen wie (4.1) besitzt. (4.3) ist ein sogenanntes *gestaffeltes Gleichungssystem*, das man leicht durch Rückwärtssubstitution lösen kann, solange R invertierbar ist. Werden keine Spaltenvertauschungen durchgeführt, dann gilt $x = y$.

4.2.1 Lösung gestaffelter Gleichungssysteme

Gestaffelte Gleichungssysteme

$$Ry = c \tag{4.3}$$

mit einer oberen Dreiecksmatrix

$$R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ 0 & & r_{nn} \end{pmatrix}, \tag{4.4}$$

sowie

$$Lz = d \tag{4.5}$$

mit einer unteren Dreiecksmatrix

$$L = \begin{pmatrix} l_{11} & & 0 \\ \vdots & \ddots & \\ l_{n1} & \cdots & l_{nn} \end{pmatrix},$$

lassen sich offensichtlich leicht durch Rückwärts- bzw. Vorwärtssubstitution lösen:

Satz 4.2.1. Seien $R = (r_{ij}) \in \mathbb{R}^{n \times n}$ und $L = (l_{ij}) \in \mathbb{R}^{n \times n}$ invertierbare obere bzw. untere Dreiecksmatrizen und $c = (c_1, \dots, c_n)^T$, $d = (d_1, \dots, d_n)^T$ Spaltenvektoren. Dann lassen sich die Lösungen von (4.3) bzw. (4.5) folgendermaßen berechnen:

a) **Rückwärtssubstitution für obere Dreieckssysteme (4.3):**

$$y_i = \frac{c_i - \sum_{j=i+1}^n r_{ij} y_j}{r_{ii}}, \quad i = n, n-1, \dots, 1.$$

b) **Vorwärtssubstitution für untere Dreieckssysteme (4.5):**

$$z_i = \frac{d_i - \sum_{j=1}^{i-1} l_{ij} z_j}{l_{ii}}, \quad i = 1, 2, \dots, n.$$

Bemerkung 4.2.2. Der Aufwand für die Rückwärtssubstitution ist $O(n^2)$ an elementaren Rechenoperationen, falls nicht zusätzlich eine spezielle Besetztheitsstruktur vorliegt (Dünnbesetztheit, Bandstruktur).

4.2.2 Das Gaußsche Eliminationsverfahren

Wir erklären nun (die grundsätzliche Vorgehensweise sollte aus der Linearen Algebra bekannt sein), wie man mit dem Gaußschen Eliminationsverfahren ein gestaffeltes Gleichungssystem erhält. Statt mit den Gleichungen (4.1) zu arbeiten, ist es bequemer, die Operationen an der um die rechte Seite erweiterten Koeffizientenmatrix

$$(A, b) = \left(\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & b_n \end{array} \right)$$

durchzuführen.

Beim Gaußschen Eliminationsverfahren geht man nun wie folgt vor:

Grundkonzept des Gaußschen Eliminationsverfahrens

$$0. \text{ Initialisierung: } (A^{(1)}, b^{(1)}) = \left(\begin{array}{ccc|c} a_{11}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ \vdots & & \vdots & \vdots \\ a_{n1}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array} \right) := (A, b).$$

1. **Pivotsuche:** Suche eine Gleichung r , die von x_1 abhängt, also mit $a_{r1}^{(1)} \neq 0$ und vertausche sie mit der ersten Gleichung:

$$(A^{(1)}, b^{(1)}) = \left(\begin{array}{ccc|c} a_{11}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ \vdots & & \vdots & \vdots \\ a_{r1}^{(1)} & \cdots & a_{rn}^{(1)} & b_r^{(1)} \\ \vdots & & \vdots & \vdots \\ a_{n1}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array} \right) \rightsquigarrow \left(\begin{array}{ccc|c} a_{r1}^{(1)} & \cdots & a_{rn}^{(1)} & b_r^{(1)} \\ \vdots & & \vdots & \vdots \\ a_{11}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ \vdots & & \vdots & \vdots \\ a_{n1}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array} \right) \\ =: \left(\begin{array}{ccc|c} \tilde{a}_{11}^{(1)} & \cdots & \tilde{a}_{1n}^{(1)} & \tilde{b}_1^{(1)} \\ \vdots & & \vdots & \vdots \\ \tilde{a}_{n1}^{(1)} & \cdots & \tilde{a}_{nn}^{(1)} & \tilde{b}_n^{(1)} \end{array} \right) = (\tilde{A}^{(1)}, \tilde{b}^{(1)}).$$

Ist A invertierbar, dann existiert immer ein solches r , da wegen der Invertierbarkeit von A die erste Spalte nicht verschwinden kann.

2. **Elimination:** Subtrahiere geeignete Vielfache der ersten Gleichung von den übrigen Gleichungen derart, dass die Koeffizienten von x_1 in diesen Gleichungen verschwinden. Offensichtlich muss man hierzu jeweils das l_{i1} -fache mit

$$l_{i1} = \frac{\tilde{a}_{i1}^{(1)}}{\tilde{a}_{11}^{(1)}}$$

der ersten Gleichung von der i -ten Gleichung subtrahieren:

$$(\tilde{A}^{(1)}, \tilde{b}^{(1)}) \rightsquigarrow (A^{(2)}, b^{(2)}) = \left(\begin{array}{cccc|c} \tilde{a}_{11}^{(1)} & \tilde{a}_{12}^{(1)} & \cdots & \tilde{a}_{1n}^{(1)} & \tilde{b}_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array} \right) \\ =: \left(\begin{array}{ccc|c} \tilde{a}_{11}^{(1)} & \cdots & \tilde{a}_{1n}^{(1)} & \tilde{b}_1^{(1)} \\ 0 & & & \\ \vdots & \hat{A}^{(2)} & & \hat{b}^{(2)} \\ 0 & & & \end{array} \right).$$

3. **Iteration:** Wende für $k = 2, \dots, n-1$ Schritt 1. und 2. auf $(\hat{A}^{(k)}, \hat{b}^{(k)})$ an:

1_k. Wähle ein Pivotelement $a_{rk}^{(k)} \neq 0$, $k \leq r \leq n$, vertausche Zeile k und $r \rightsquigarrow (\tilde{A}^{(k)}, \tilde{b}^{(k)})$

2_k. Subtrahiere das l_{ik} -fache mit

$$l_{ik} = \frac{\tilde{a}_{ik}^{(k)}}{\tilde{a}_{kk}^{(k)}}$$

der k -ten Gleichung von der i -ten Gleichung, $i = k+1, \dots, n$.

$\rightsquigarrow (A^{(k+1)}, b^{(k+1)})$

Nach k Eliminationsschritten

$$(A, b) =: (A^{(1)}, b^{(1)}) \rightarrow (A^{(2)}, b^{(2)}) \rightarrow \dots \rightarrow (A^{(k+1)}, b^{(k+1)})$$

erhalten wir also eine Zwischenmatrix der Form

$$(A^{(k+1)}, b^{(k+1)}) = \left(\begin{array}{ccc|ccc} \tilde{a}_{11}^{(1)} & \dots & \tilde{a}_{1k}^{(1)} & \dots & \tilde{a}_{1n}^{(1)} & \tilde{b}_1^{(1)} \\ & \ddots & & & \vdots & \vdots \\ & & \tilde{a}_{kk}^{(k)} & \dots & \tilde{a}_{kn}^{(k)} & \tilde{b}_k^{(k)} \\ \hline & & 0 & & & \\ & & \vdots & \hat{A}^{(k+1)} & & \hat{b}^{(k+1)} \\ & & 0 & & & \end{array} \right).$$

Nach $n - 1$ Eliminationsschritten liegt somit ein gestaffeltes Gleichungssystem (4.3)

$$Rx = c, \quad \text{mit } R = A^{(n)}, \quad c = b^{(n)}$$

vor.

4.2.3 Pivotstrategie

Das Element $a_{rk}^{(k)}$, das in Schritt 1_k bestimmt wird, heißt *Pivotelement*. Theoretisch kann man bei der Pivotsuche jedes $a_{rk}^{(k)} \neq 0$ als Pivotelement wählen. Die Wahl kleiner Pivotelemente kann aber zu einer dramatischen Verstärkung von Rundungsfehlern führen.

Als Ausweg gibt es zwei typische Möglichkeiten:

- **Spaltenpivotsuche:** Wähle $k \leq r \leq n$ mit

$$|a_{rk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|.$$

- **Vollständige Pivotsuche:** Bestimme $k \leq r \leq n, k \leq s \leq n$ mit

$$|a_{rs}^{(k)}| = \max_{k \leq i, j \leq n} |a_{ij}^{(k)}|.$$

Hierbei sollten die Zeilen von A „equilibriert“ sein, also ihre Normen dieselbe Größenordnung haben.

Beispiel 4.2.3. Betrachte das Beispiel

$$\begin{pmatrix} 1 & 2 & -1 \\ 2 & -2 & 4 \\ 2 & 1 & -2 \end{pmatrix} x = \begin{pmatrix} 2 \\ 10 \\ -2 \end{pmatrix}$$

Dies liefert mit Spaltenpivotsuche

$$\begin{array}{ccc}
 \left(\begin{array}{ccc|c} 1 & 2 & -1 & 2 \\ 2 & -2 & 4 & 10 \\ 2 & 1 & -2 & -2 \end{array} \right) & \xrightarrow{\text{Spaltenpivotsuche}} & \left(\begin{array}{ccc|c} 2 & -2 & 4 & 10 \\ 1 & 2 & -1 & 2 \\ 2 & 1 & -2 & -2 \end{array} \right) \\
 & & \xrightarrow{\text{Elimination}} & \begin{array}{l} \underbrace{-(1/2)}_{=l_{21}} \cdot \text{Zeile 1} \\ \underbrace{-(1)}_{=l_{31}} \cdot \text{Zeile 1} \end{array} \left(\begin{array}{ccc|c} 2 & -2 & 4 & 10 \\ 0 & 3 & -3 & -3 \\ 0 & 3 & -6 & -12 \end{array} \right) \\
 & & \xrightarrow{\text{Spaltenpivotsuche+Elimination}} & \underbrace{-1}_{=l_{32}} \cdot \text{Zeile 2} \left(\begin{array}{ccc|c} 2 & -2 & 4 & 10 \\ 0 & 3 & -3 & -3 \\ 0 & 0 & -3 & -9 \end{array} \right)
 \end{array}$$

4.2.4 Praktische Implementierung des Gauß-Verfahrens – LR-Zerlegung

Bei der Realisierung auf einem Rechner speichert man in der Regel auch die verwendeten Multiplikatoren l_{ik} . Wir werden sehen, dass das Gaußsche Eliminationsverfahren dann „gratis“ eine Dreieckszerlegung (oder LR-Zerlegung) von A der Form

$$LR = PA \quad (4.6)$$

liefert. Hierbei ist $R \in \mathbb{R}^{n \times n}$ eine obere Dreiecksmatrix (4.4), $L \in \mathbb{R}^{n \times n}$ eine untere Dreiecksmatrix der Form

$$L = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & & \ddots & \ddots & \\ l_{n1} & & \cdots & l_{n,n-1} & 1 \end{pmatrix}, \quad (4.7)$$

und P eine Permutationsmatrix, die lediglich die Zeilen von A permutiert (siehe unten).

Wir erhalten die folgende Implementierung des Gauß-Verfahrens mit Spaltenpivotsuche:

Algorithmus 4.2.4. Gaußsches Eliminationsverfahren mit Spaltenpivotsuche

Setze $(A^{(1)}, b^{(1)}) = (A, b)$ und $L^{(1)} = 0 \in \mathbb{R}^{n \times n}$.

Für $k = 1, 2, \dots, n-1$:

1. **Spaltenpivotsuche:** Bestimme $k \leq r \leq n$ mit

$$|a_{rk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|.$$

Falls $a_{rk}^{(k)} = 0$: STOPP, A ist singulär.

Vertausche die Zeilen r und k von $(A^{(k)}, b^{(k)})$ und von $L^{(k)}$. Das Ergebnis sei formal mit $(\tilde{A}^{(k)}, \tilde{b}^{(k)})$, $\tilde{L}^{(k)}$ bezeichnet.

2. **Elimination:** Subtrahiere für $i = k + 1, \dots, n$ das l_{ik} -fache, $l_{ik} = \frac{\tilde{a}_{ik}^{(k)}}{\tilde{a}_{kk}^{(k)}}$, der k -ten Zeile von $(\tilde{A}^{(k)}, \tilde{b}^{(k)})$ von der i -ten Zeile und füge die Multiplikatoren l_{ik} in $\tilde{L}^{(k)}$ ein. Das Ergebnis sei formal mit $(A^{(k+1)}, b^{(k+1)})$ und $L^{(k+1)}$ bezeichnet.

Im Detail: Initialisiere $(A^{(k+1)}, b^{(k+1)}) := (\tilde{A}^{(k)}, \tilde{b}^{(k)})$, $L^{(k+1)} := \tilde{L}^{(k)}$.

Für $i = k + 1, \dots, n$:

$$\begin{aligned} l_{ik} &= \frac{\tilde{a}_{ik}^{(k)}}{\tilde{a}_{kk}^{(k)}}, \\ b_i^{(k+1)} &= \tilde{b}_i^{(k)} - l_{ik} \tilde{b}_k^{(k)}, \\ a_{ik}^{(k+1)} &= 0, \\ l_{ik}^{(k+1)} &= l_{ik} \quad (\text{Multiplikator speichern}). \end{aligned}$$

Für $j = k + 1, \dots, n$:

$$a_{ij}^{(k+1)} = \tilde{a}_{ij}^{(k)} - l_{ik} \tilde{a}_{kj}^{(k)}$$

Ergebnis: $R := A^{(n)}$, $c := b^{(n)}$, $L := I + L^{(n)}$ mit der Einheitsmatrix $I \in \mathbb{R}^{n \times n}$.

Für die Spaltenpivotsuche muss Schritt 1 in Algorithmus 4.2.4 wie folgt modifiziert werden:

Algorithmus 4.2.5. Gaußsches Eliminationsverfahren mit vollständiger Pivotsuche

Algorithmus 4.2.4 mit folgender Modifikation von Schritt 1:

1.' **Vollständige Pivotsuche:** Bestimme $k \leq r \leq n$, $k \leq s \leq n$ mit

$$|a_{rs}^{(k)}| = \max_{k \leq i, j \leq n} |a_{ij}^{(k)}|.$$

Falls $a_{rs}^{(k)} = 0$: STOPP, A ist singulär.

Vertausche die Zeilen r und k sowie die Spalten s und k von $(A^{(k)}, b^{(k)})$ und von $L^{(k)}$. Das Ergebnis sei formal mit $(\tilde{A}^{(k)}, \tilde{b}^{(k)})$, $\tilde{L}^{(k)}$ bezeichnet.

Achtung: Bei jeder Spaltenvertauschung müssen die Komponenten von x entsprechend umnummeriert werden, d. h. nach Lösen von (4.3) müssen die Komponenten des Ergebnisvektors x zurückgetauscht werden.

In der Regel wird vollständige Pivotsuche nur bei „fast singulären“ Matrizen angewandt, um den Rundungsfehlereinfluss minimal zu halten.

Das vom Verfahren gelieferte gestaffelte Gleichungssystem

$$Rx = c, \quad \text{mit} \quad R = A^{(n)}, \quad c = b^{(n)}$$

hat dieselbe Lösungsmenge wie das ursprüngliche System $Ax = b$.

Bei einer Implementierung auf dem Rechner kann man für die Speicherung aller $A^{(k)}$, $b^{(k)}$, $\tilde{A}^{(k)}$, $\tilde{b}^{(k)}$ die Felder verwenden, in denen A und b gespeichert waren. $L^{(k)}$ kann man anstelle der entstehenden Nullen im strikten unteren Dreieck platzsparend speichern.

Bemerkung 4.2.6. Der Rechenaufwand ist $O(n^3/3 - n/3)$ an elementaren Rechenoperationen, falls nicht zusätzlich eine spezielle Besetztheitsstruktur vorliegt.

4.2.5 Matrixdarstellung der Eliminationsschritte

Wir betrachten das Gaußsche Eliminationsverfahren mit Spaltenpivotsuche (Algorithmus 4.2.4). Formal lässt sich der Übergang $(A^{(k)}, b^{(k)}) \rightarrow (\tilde{A}^{(k)}, \tilde{b}^{(k)}) \rightarrow (A^{(k+1)}, b^{(k+1)})$ durch Multiplikation mit Matrizen darstellen. Tatsächlich gilt

$$\begin{aligned} (\tilde{A}^{(k)}, \tilde{b}^{(k)}) &= P_k(A^{(k)}, b^{(k)}) \quad (\text{Zeilenvertauschung}) \\ (A^{(k+1)}, b^{(k+1)}) &= L_k(\tilde{A}^{(k)}, \tilde{b}^{(k)}) = L_k P_k(A^{(k)}, b^{(k)}) \quad (\text{Elimination}) \end{aligned}$$

mit der *elementaren Permutationsmatrix* (vertausche Zeile k und r der Einheitsmatrix)

$$P_k = \begin{matrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ k \rightarrow & & & 1 & & & & \\ & & & 0 & & & 1 & \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ r \rightarrow & & & & & & 1 & 0 \\ & & & 1 & & & 0 & 1 \\ & & & & & & & \ddots \end{matrix} \quad (4.8)$$

und der *elementaren Eliminationsmatrix*

$$L_k = \begin{pmatrix} 1 & & & 0 \\ & \ddots & & \\ & & 1 & \\ & & -l_{k+1,k} & 1 \\ 0 & & \vdots & & \ddots \\ & & -l_{nk} & 0 & 1 \end{pmatrix} \quad (4.9)$$

Nach den $n - 1$ Schritten des Gaußschen Algorithmus erhalten wir somit

$$R = A^{(n)} = L_{n-1} P_{n-1} \cdots L_1 P_1 A.$$

Sind im Eliminationsverfahren keine Zeilenvertauschungen nötig, dann erhalten wir $R = A^{(n)} = L_{n-1} \cdots L_1 A$ und somit

$$A = L_1^{-1} \cdots L_{n-1}^{-1} R =: LR.$$

Man rechnet leicht nach, dass gilt

$$L = L_1^{-1} \cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & 0 \\ l_{21} & 1 & \\ l_{31} & l_{32} & 1 \\ \vdots & \ddots & \ddots \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{pmatrix} = I + L^{(n)}.$$

Mit den vom Gauß-Verfahren gelieferten Matrizen L und R gilt also ohne Pivotsuche

$$A = LR.$$

Allgemein gilt der folgende Satz.

Satz 4.2.7. Es sei $A \in \mathbb{R}^{n \times n}$ nichtsingulär. Dann gilt:

- i) Das Gaußsche Eliminationsverfahren aus Algorithmus 4.2.4 liefert eine untere Dreiecksmatrix L der Form (4.7) und eine obere Dreiecksmatrix R mit

$$LR = PA.$$

Hierbei ist $P = P_{n-1} \cdots P_1$ eine Permutationsmatrix, wobei jeweils P_k die Permutationsmatrix für die Zeilenvertauschung im k -ten Schritt ist.

- ii) Algorithmus 4.2.5 liefert eine Dreieckszerlegung

$$LR = PAQ$$

Hierbei ist P wie oben und $Q = Q_1 \cdots Q_{n-1}$, wobei jeweils Q_k die Permutationsmatrix für die Spaltenvertauschung im k -ten Schritt ist.

Beweis. Finden keine Zeilen- und Spaltenvertauschungen statt, dann haben wir die Behauptung bereits gezeigt.

Im allgemeinen Fall kann man zeigen, dass das Gauß-Verfahren mit Spaltenpivotsuche (bzw. vollständiger Pivotsuche) dasselbe Ergebnis L, R liefert wie wenn man das Gauß-Verfahren ohne Pivotsuche auf PA (bzw. PAQ) anwendet. \square

Bemerkung 4.2.8. Eine Dreieckszerlegung (4.6) (oder $LR = PAQ$) ist sehr nützlich, wenn man (4.1) für mehrere rechte Seiten lösen will. Tatsächlich gilt

$$Ax = b \iff PAQy = Pb, \quad x = Qy \iff L \underbrace{Ry}_{=:z} = Pb, \quad x = Qy,$$

wobei $Q = I$ bei Spaltenpivotsuche. Man erhält nun x durch folgende Schritte:

Vorwärts-Rückwärtssubstitution für Dreieckszerlegung:

Löse $Lz = Pb$ nach z durch Vorwärtssubstitution gemäß Satz 4.2.1.

Löse $Ry = z$ nach y durch Rückwärtssubstitution gemäß Satz 4.2.1.

Lösung: $x = Qy$.

Liegt also die Dreieckszerlegung vor, dann kann (4.1) für jede rechte Seite in $O(n^2)$ Operationen berechnet werden.

4.2.6 Matrizenklassen, die keine Pivotsuche erfordern

Es gibt einige wichtige Teilklassen von Matrizen, bei denen auf die Pivotsuche verzichtet werden kann:

- $A = A^T$ ist symmetrisch positiv definit, also

$$x^T A x > 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\}.$$

Wir gehen hierauf noch ein.

- A ist strikt diagonaldominant, d. h.

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

- A ist M-Matrix, d. h. es gilt

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

$$a_{ij} \leq 0, \quad i \neq j$$

$$D^{-1}(A - D), \quad D = \text{diag}(a_{11}, \dots, a_{nn}) \text{ hat lauter Eigenwerte vom Betrag } < 1.$$

4.3 Das Cholesky-Verfahren

Für allgemeine invertierbare Matrizen kann das Gauß-Verfahren ohne Pivotsuche zusammenbrechen und wir werden sehen, dass auch aus Gründen der numerischen Stabilität eine Pivotsuche ratsam ist. Für die wichtige Klasse der positiv definiten Matrizen ist jedoch das Gauß-Verfahren immer ohne Pivotsuche numerisch stabil durchführbar.

Definition 4.3.1. Eine reelle Matrix $A \in \mathbb{R}^{n \times n}$ heißt positiv definit, falls gilt

$$A = A^T, \quad x^T A x > 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\}.$$

und positiv semi-definit, falls gilt

$$A = A^T, \quad x^T A x \geq 0 \quad \forall x \in \mathbb{R}^n.$$

Allgemeiner heißt eine komplexe Matrix $A \in \mathbb{C}^{n \times n}$ positiv definit, falls gilt

$$A = A^H, \quad x^H A x > 0 \quad \forall x \in \mathbb{C}^n \setminus \{0\}.$$

und positiv semi-definit, falls gilt

$$A = A^H, \quad x^H A x \geq 0 \quad \forall x \in \mathbb{C}^n.$$

Hierbei ist $A^H = (\bar{a}_{ji})_{1 \leq i \leq n, 1 \leq j \leq n}$, wobei \bar{a}_{ji} die komplexe Konjugation bezeichnet.

Positiv definite Matrizen treten sehr oft in Anwendungen auf, etwa bei der numerischen Lösung von elliptischen (z. B. Laplace-Gleichung) und parabolischen (z. B. Wärmeleitungsgleichung) partiellen Differentialgleichungen.

Positive definite Matrizen sind invertierbar.

Eine effiziente Variante des Gaußschen Verfahrens für Gleichungssysteme mit positiv definiter Matrix wurde von Cholesky angegeben. Das Cholesky-Verfahren beruht auf der folgenden Beobachtung

Satz 4.3.2. Es sei $A \in \mathbb{R}^{n \times n}$ positiv definit. Dann gibt es genau eine untere Dreiecksmatrix L mit positiven Diagonaleinträgen $l_{ii} > 0$, so dass

$$LL^T = A \quad (\text{Cholesky-Zerlegung}).$$

Ferner besitzt A eine eindeutige Dreieckszerlegung

$$\tilde{L}\tilde{R} = A,$$

wobei $\tilde{L} = LD^{-1}$, $\tilde{R} = DL^T$ mit $D = \text{diag}(l_{11}, \dots, l_{nn})$. Sie wird vom Gauß-Verfahren ohne Pivotsuche geliefert.

Der Beweis kann durch vollständige Induktion nach n erfolgen, wir wollen ihn aber nicht ausführen.

Die Cholesky-Zerlegung $LL^T = A$ berechnet man durch Auflösen der $\frac{n(n+1)}{2}$ Gleichungen (aus Symmetriegründen muss nur das untere Dreieck mit Diagonale betrachtet werden)

$$a_{ij} = \sum_{k=1}^j l_{ik}l_{jk}, \quad \text{für } j \leq i, \quad i = 1, \dots, n. \quad (4.10)$$

Man kann hieraus die Elemente von L spaltenweise in der Reihenfolge

$$l_{11}, \dots, l_{n1}, l_{22}, \dots, l_{n2}, \dots, l_{nn}$$

berechnen. Für die erste Spalte von L (setze $j = 1$) ergibt sich

$$\begin{aligned} a_{11} &= l_{11}^2, \text{ also } l_{11} = \sqrt{a_{11}} \\ a_{i1} &= l_{i1}l_{11}, \text{ also } l_{i1} = a_{i1}/l_{11}. \end{aligned}$$

Sukzessives Auflösen nach l_{ij} , $i = j, \dots, n$ liefert den folgenden Algorithmus.

Algorithmus 4.3.3. Cholesky-Verfahren zur Berechnung der Zerlegung $LL^T = A$

Für $j = 1, \dots, n$

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}$$

Für $i = j + 1, \dots, n$:

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}}{l_{jj}}$$

Bemerkung 4.3.4. Das Cholesky-Verfahren hat einige schöne Eigenschaften:

- Da das Cholesky-Verfahren die Symmetrie ausnutzt, benötigt es neben n Quadratwurzeln nur noch $O(n^3/6)$ Operationen. Dies ist etwa die Hälfte der beim Gauß-Verfahren benötigten Operationen.

- Aus (4.10) folgt

$$|l_{ij}| \leq \sqrt{a_{ii}}, \quad j \leq i, \quad i = 1, \dots, n.$$

Die Elemente der Matrix L können daher nicht zu groß werden. Dies ist ein wesentlicher Grund für die numerische Stabilität des Cholesky-Verfahrens.

- Das Cholesky-Verfahren ist die effizienteste allgemeine Testmethode auf positive Definitheit. Man muss hierbei Algorithmus 4.3.3 nur wie folgt erweitern:

$$a = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2. \quad \text{Falls } a \leq 0: \text{ STOPP, } A \text{ nicht positiv definit.}$$

Sonst setze $l_{jj} = \sqrt{a}$.

4.4 Fehlerabschätzungen und Rundungsfehlereinfluss

Bei der Beschreibung der direkten Verfahren zur Lösung von Gleichungssystemen sind wir bisher davon ausgegangen, dass alle Ausgangsdaten exakt vorliegen und die Rechnung ohne Rundungsfehler durchgeführt wird. Dies ist jedoch unrealistisch, denn insbesondere bei großen Systemen können Rundungsfehler die Rechnung erheblich beeinflussen.

4.4.1 Fehlerabschätzungen für gestörte Gleichungssysteme

Wir studieren zunächst, wie stark sich die Lösung eines linearen Gleichungssystems bei Störung von Matrix und rechter Seite ändert. Vorgelegt sei ein lineares Gleichungssystem

$$Ax = b$$

und ein gestörtes System

$$(A + \Delta A)\tilde{x} = b + \Delta b$$

mit ΔA und Δb „klein“.

Frage: Wie klein ist $x - \tilde{x}$?

Diese Fragestellung ist von größter praktischer Bedeutung:

- Man kann abschätzen, wie sensitiv die Lösung bezüglich Störungen von Matrix und rechter Seite ist.
- Eine berechnete Näherungslösung (z. B. mit einer Implementierung des Gauß-Verfahrens) \tilde{x} von $Ax = b$ ist exakte Lösung des Systems

$$A\tilde{x} = b + \Delta b, \quad \text{mit dem Residuum } \Delta b = A\tilde{x} - b.$$

Man kann nun aus dem leicht berechenbaren Residuum $\Delta b = A\tilde{x} - b$ Schranken an den unbekannten Fehler $\|x - \tilde{x}\|$ ableiten.

Es stellt sich heraus, dass die sogenannte Konditionzahl einer Matrix diesen Störeinfluss beschreibt.

Zur Messung von $x - \tilde{x}$, Δb und ΔA benötigen wir einen „Längenbegriff“ für Vektoren und Matrizen.

Definition 4.4.1. Eine Vektornorm auf \mathbb{R}^n ist eine Abbildung $x \in \mathbb{R}^n \mapsto \|x\| \in [0, \infty[$ mit folgenden Eigenschaften:

- a) $\|x\| = 0$ nur für $x = 0$.
- b) $\|\alpha x\| = |\alpha| \|x\|$ für alle $\alpha \in \mathbb{R}$ und alle $x \in \mathbb{R}^n$.
- c) $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in \mathbb{R}^n$ (Dreiecksungleichung).

Nun sollen auch *Matrix-Normen* eingeführt werden. Sei hierzu $\|\cdot\|$ eine beliebige Norm auf \mathbb{R}^n . Dann können wir auf $\mathbb{R}^{n \times n}$ eine zugehörige Matrix-Norm definieren durch

$$\|A\| := \sup_{\|x\|=1} \|Ax\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} \quad (4.11)$$

für $A \in \mathbb{R}^{n \times n}$. Sie heißt die durch die Vektornorm $\|\cdot\|$ induzierte Matrix-Norm.

Sie hat wiederum die Eigenschaften

- a) $\|A\| = 0$ nur für $A = 0$.
- b) $\|\alpha A\| = |\alpha| \|A\|$ für alle $\alpha \in \mathbb{R}$ und alle $A \in \mathbb{R}^{n \times n}$.
- c) $\|A + B\| \leq \|A\| + \|B\|$ für alle $A, B \in \mathbb{R}^{n \times n}$ (Dreiecksungleichung).

Zusätzlich sichert (4.11) die nützlichen Ungleichungen

- d) $\|Ax\| \leq \|A\| \|x\|$ für alle $x \in \mathbb{R}^n$ und alle $A \in \mathbb{R}^{n \times n}$ (Verträglichkeitsbedingung).
- e) $\|AB\| \leq \|A\| \|B\|$ für alle $A, B \in \mathbb{R}^{n \times n}$ (Submultiplikativität).

Beispiele 4.4.2.

$$\begin{aligned} \|x\|_2 &= \sqrt{x^T x} \quad \text{induziert} \quad \|A\|_2 = \sqrt{\lambda_{\max}(A^T A)} \\ \|x\|_1 &= \sum_{i=1}^n |x_i| \quad \text{induziert} \quad \|A\|_1 = \max_{j=1, \dots, n} \sum_{i=1}^n |a_{ij}| \quad (\text{Spaltensummennorm}) \\ \|x\|_\infty &= \max_{i=1, \dots, n} |x_i| \quad \text{induziert} \quad \|A\|_\infty = \max_{i=1, \dots, n} \sum_{j=1}^n |a_{ij}| \quad (\text{Zeilensummennorm}) \end{aligned}$$

Wir sind nun in der Lage, die bereits erwähnte Konditionszahl einer Matrix einzuführen.

Definition 4.4.3. Sei $A \in \mathbb{R}^{n \times n}$ invertierbar und sei $\|\cdot\|$ eine induzierte Matrixnorm. Dann heißt die Zahl $\text{cond}(A) = \|A\| \|A^{-1}\|$ die Konditionszahl von A bezüglich der Matrixnorm.

Man kann nun folgendes zeigen.

Satz 4.4.4 (Störeinfluss von Matrix und rechter Seite). Sei $A \in \mathbb{R}^{n \times n}$ invertierbar, $b, \Delta b \in \mathbb{R}^n$, $b \neq 0$ und $\Delta A \in \mathbb{R}^{n \times n}$ mit $\|\Delta A\| < 1/\|A^{-1}\|$ mit einer beliebigen durch eine Norm $\|\cdot\|$ auf \mathbb{R}^n induzierten Matrixnorm $\|\cdot\|$. Ist x die Lösung von

$$Ax = b$$

und \tilde{x} die Lösung von

$$(A + \Delta A)\tilde{x} = b + \Delta b,$$

dann gilt

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \frac{\text{cond}(A)}{1 - \text{cond}(A)\|\Delta A\|/\|A\|} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right).$$

Beweis. Wir betrachten der Einfachheit halber nur den Fall $\Delta A = 0$. Dann liefert Subtraktion der gestörten und ungestörten Gleichung

$$A(\tilde{x} - x) = \Delta b,$$

also

$$\|\tilde{x} - x\| = \|A^{-1}\Delta b\| \leq \|A^{-1}\|\|\Delta b\|.$$

Wegen $\|b\| = \|Ax\| \leq \|A\|\|x\|$ folgt $1/\|x\| \leq \|A\|/\|b\|$ und somit

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \|A\|\|A^{-1}\| \frac{\|\Delta b\|}{\|b\|}. \quad \square$$

Die Konditionszahl bestimmt also die Sensitivität bezüglich Störungen von Matrix und rechter Seite.

4.4.2 Rundungsfehleranalyse für das Gauß-Verfahren

Durch eine elementare aber aufwendige Abschätzung der beim Gauß-Verfahren auftretenden Rundungsfehlerverstärkung erhält man folgendes Resultat.

Satz 4.4.5. Sei $A \in \mathbb{R}^{n \times n}$ invertierbar. Wendet man das Gauß-Verfahren auf einem Rechner mit Maschinengenauigkeit eps mit einer Pivot-Technik an, die $|l_{ij}| \leq 1$ sicherstellt (z. B. Spaltenpivotsuche oder vollständige Pivotsuche), dann errechnet man \bar{L}, \bar{R} mit

$$\bar{L}\bar{R} = PAQ + F, \quad |f_{ij}| \leq 2j\bar{a} \frac{\text{eps}}{1 - \text{eps}}.$$

Hierbei sind P, Q die aus der Pivotsuche resultierenden Permutationen und

$$\bar{a} = \max_k \bar{a}_k, \quad \bar{a}_k = \max_{i,j} |a_{ij}^{(k)}|. \quad (4.12)$$

Berechnet man mit Hilfe von \bar{L}, \bar{R} durch Vorwärts- und Rückwärtssubstitution eine Näherungslösung \tilde{x} von $Ax = b$, dann existiert eine Matrix E mit

$$(A + E)\tilde{x} = b, \quad |e_{ij}| \leq \frac{2(n+1)\text{eps}}{1 - n\text{eps}} (|\bar{L}||\bar{R}|)_{ij} \leq \frac{2(n+1)\text{eps}}{1 - n \cdot \text{eps}} n\bar{a}.$$

Hierbei bezeichnet $|\bar{L}| = (|\bar{l}_{ij}|)$, $|\bar{R}| = (|\bar{r}_{ij}|)$.

Beweis. Siehe Stoer [5]. □

Bemerkung 4.4.6. Mit Satz 4.4.4 kann man nun auch den relativen Fehler der Näherungslösung \tilde{x} abschätzen.

Einfluss der Pivot-Strategie

Die Größe von \bar{a} in (4.12) hängt von der Pivotstrategie ab. Man kann folgendes zeigen:

- **Spaltenpivotsuche:** $\bar{a}_k \leq 2^k \max_{i,j} |a_{ij}|$.
Diese Schranke kann erreicht werden, ist aber in der Regel viel zu pessimistisch. In der Praxis tritt fast immer $\bar{a}_k \leq 10 \max_{i,j} |a_{ij}|$ auf.
- **Spaltenpivotsuche bei Tridiagonalmatrizen:** $\bar{a}_k \leq 2 \max_{i,j} |a_{ij}|$.
- **Vollständige Pivotsuche:** $\bar{a}_k \leq f(k) \max_{i,j} |a_{ij}|$, $f(k) = k^{1/2} (2^1 3^{1/2} \dots k^{1/(k-1)})^{1/2}$.
 $f(n)$ wächst recht langsam. Es ist bislang kein Beispiel mit $\bar{a}_k \geq (k+1) \max_{i,j} |a_{ij}|$ entdeckt worden.

Beispiel 4.4.7. Betrachte die Hilbert-Matrix $H^n = (h_{ij}^n) \in \mathbb{R}^{n \times n}$ wobei

$$h_{ij}^n = \frac{1}{i+j-1}, \quad i, j \in \{1, \dots, n\}.$$

Diese Matrix ist bekanntermaßen schlecht konditioniert. So ist beispielsweise $\text{cond}(H^5) \approx 9.4 \cdot 10^5$ (bzgl. $\|\cdot\|_\infty$), $\|H^5\|_\infty \approx 2.3$ und $\|(H^5)^{-1}\|_\infty \approx 4.1 \cdot 10^5$. Bei Anwendung des Gauß-Verfahrens mit Spaltenpivotisierung ist $\bar{a} = 1$.

Für $n = 5$ und $\text{eps} = 10^{-16}$ ergibt sich mit Satz 4.4.5

$$|e_{ij}| \leq \frac{2(n+1)\text{eps}}{1-n \cdot \text{eps}} n\bar{a} = \frac{6 \cdot 10^{-15}}{1-5 \cdot 10^{-16}} \approx 6 \cdot 10^{-15}$$

und damit $\|E\|_\infty \leq 3 \cdot 10^{-14}$. Satz 4.4.4 liefert nun:

$$\begin{aligned} \frac{\|\tilde{x} - x\|_\infty}{\|x\|_\infty} &\leq \frac{\text{cond}(A)}{1 - \text{cond}(A)\|E\|_\infty/\|A\|_\infty} \frac{\|E\|_\infty}{\|A\|_\infty} = \frac{\|A^{-1}\|_\infty \|E\|_\infty}{1 - \|A^{-1}\|_\infty \|E\|_\infty} \\ &\approx \frac{4.1 \cdot 10^5 \cdot 3 \cdot 10^{-14}}{1 - 4.1 \cdot 10^5 \cdot 3 \cdot 10^{-14}} \approx 1.23 \cdot 10^{-8}. \end{aligned}$$

Durch Rundungsfehler „verliert“ man also etwa die Hälfte der Stellen. Für größere n wird der Rundungsfehler allerdings rapide größer und macht die Ergebnisse bald nicht mehr brauchbar. (Bemerkung: Satz 4.4.4 ist für größeres n nicht mehr anwendbar, weil $\|\Delta A\| > 1/\|A^{-1}\|$ gilt.)