3 Direkte Lösung linearer Gleichungssysteme

Lineare Gleichungssysteme (LGS) sind zwar ein verhältnismäßig einfaches Problem, bedürfen aber gerade für große Dimensionen sehr guter numerischer Lösungsverfahren. Hinzu kommt, dass LGS in extrem vielen Anwendungen vorkommen. Oft kann man das Wissen über die „Herkunft“ eines Gleichungssystems zu dessen schnellem Lösen nutzen.

Beispiel 3.0.1 (Schwingungsgleichung) Gegeben sei eine elastische Saite der Länge 1, die an beiden Enden fixiert ist. Die Saite wird nun durch eine äußere Kraft f ausgelenkt (angezupft). Wir wollen die Auslenkung u der Saite aus ihrer Ruhelage als Funktion von $x \in [0,1]$ berechnen. Die

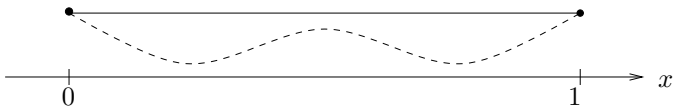


Abb. 3.1: Elastische, an beiden Enden fixierte Saite.

gesuchte Auslenkung $u : [0,1] \to \mathbb{R}$ ist Lösung des folgenden linearen Randwertproblems zweiter Ordnung

$$-u''(x) + \lambda(x)u(x) = f(x)\ ,\ x \in (0,1), \qquad u(0) = u(1) = 0 \tag{3.1}$$

mit gegebenen $f : (0,1) \to \mathbb{R}$ und $\lambda : (0,1) \to \mathbb{R}$. Genaueres zur Modellierung findet man z.B. in [Arendt/Urban].
Wir wollen (3.1) näherungsweise mit Hilfe eines numerischen Verfahrens lösen. Dazu unterteilen wir $[0,1]$ in Teilintervalle gleicher Länge. Die Anzahl der Intervalle sei $N > 1, N \in \mathbb{N}$ und $h = \frac{1}{N}$ die Schrittweite. Dann setzt man $x_i := ih\ ,\ i = 0, \ldots, N \quad (x_0 = 0, x_N = 1)$, die x_i werden

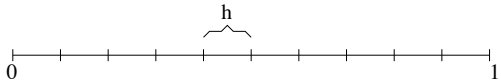


Abb. 3.2: Unterteilung des Intervalls in Teilintervalle der Länge $h > 0$.

als **Knoten** bezeichnet. Die **Schrittweite** ist $h := x_{i+1} - x_i$ für alle i, man spricht von einem **äquidistanten Gitter**. Wir wollen die Lösung an den Knoten x_i approximieren und ersetzen hierzu (wie aus der Analysis bekannt) die zweite Ableitung durch den zentralen Differenzenquotienten

$$u''(x_i) \approx \frac{1}{h^2}\big(u(x_{i-1}) - 2u(x_i) + u(x_{i+1})\big) =: D_c^2 u(x_i).$$

Es gilt bekanntlich $\|u'' - D_c^2 u\| = \mathcal{O}(h^2)$, falls $u \in C^4[0,1]$. Damit erhält man für die Näherung $u_i \approx u(x_i)$ also folgende Bedingungen ($\lambda_i = \lambda(x_i), f_i = f(x_i)$):

$$\begin{cases} \frac{1}{h^2}(-u_{i-1} + 2u_i - u_{i+1}) + \lambda_i u_i = f_i,\ 1 \le i \le N-1, \\ u_0 = u_N = 0, \end{cases}$$

also ein lineares Gleichungssystem der Form

$$\underbrace{\begin{bmatrix} (2+h^2\lambda_1) & -1 & & 0 \\ -1 & (2+h^2\lambda_2) & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & (2+h^2\lambda_{n-1}) \end{bmatrix}}_{=:A_h} \underbrace{\begin{bmatrix} u_1 \\ \vdots \\ \vdots \\ u_{n-1} \end{bmatrix}}_{=:u_h} = \underbrace{\begin{bmatrix} h^2 f_1 \\ \vdots \\ \vdots \\ h^2 f_{n-1} \end{bmatrix}}_{=:f_h},$$

d.h. $A_h u_h = f_h$. Für $N \to \infty$ ($h \to 0$) konvergiert die „**diskrete Lösung**“ u_h gegen die Lösung u von (3.1). Allerdings wächst die Dimension der Matrix A_h mit kleiner werdendem h. Bei mehrdimensionalen Problemen führt dies leicht zu sehr großen LGS. Wir nennen diese Matrix auch **Standardmatrix**.

3.1 Einführung, Cramersche Regel

Wir beginnen mit dem klassischen Verfahren zur Lösung eines linearen Gleichungssystems (LGS), der Gaußschen[1] Eliminationsmethode.

Zu lösen ist ein System von n linearen Gleichungen mit n Unbekannten $x_1, \ldots, x_n \in \mathbb{R}$

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array} \tag{3.2}$$

oder kurz

$$Ax = b, \tag{3.3}$$

wobei $A \in \mathbb{R}^{n\times n}$ eine reelle $n \times n$-Matrix (also insbesondere eine quadratische Matrix) ist und $b, x \in \mathbb{R}^n$ reelle (Spalten-)Vektoren sind.

Wann ist ein lineares Gleichungssystem überhaupt lösbar? Aus der Linearen Algebra (siehe z.B. [Wille]) kennen wir das folgende Resultat, das die Lösbarkeit mit Hilfe der Determinante der Matrix A charakterisiert.

Satz 3.1.1 (Lösbarkeit) *Sei $A \in \mathbb{R}^{n\times n}$ mit $\det A \neq 0$ und $b \in \mathbb{R}^n$. Dann existiert genau ein $x \in \mathbb{R}^n$, so dass $Ax = b$.*

Falls $\det A \neq 0$, so lässt sich die Lösung $x = A^{-1}b$ mit der **Cramerschen Regel** berechnen, d.h.

$$x_i = \frac{1}{\det A} \begin{vmatrix} a_{11} & \ldots & b_1 & \ldots & a_{1n} \\ a_{21} & \ldots & b_2 & \ldots & a_{2n} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \ldots & b_n & \ldots & a_{nn} \end{vmatrix} = \frac{D_i}{D} \qquad (i = 1, \ldots, n).$$

Dabei geht die im Zähler stehende Determinante D_i dadurch aus $D := \det A$ hervor, dass man die i-te Spalte der Matrix A durch den Vektor b der rechten Seite ersetzt.

[1] **Carl Friedrich Gauß**, 1777 - 1855. Lagrange hatte 1759 die Methode schon vorweggenommen und in China war sie schon vor dem ersten Jahrhundert bekannt. Näheres zu Gauß, Lagrange und weiteren Mathematikern findet man im Internet unter `www-groups.dcs.st-andrews.ac.uk/~history`.

Man beachte hier die Verbindung von Existenz- und Eindeutigkeitsaussage mit dem Rechenverfahren, was einen „guten" Algorithmus ausmacht. Dieser braucht dabei nicht unbedingt optimal zu sein!
Wenn wir die Lösung eines linearen Gleichungssystems mit Hilfe der Cramerschen Regel bestimmen wollen, müssen wir die verschiedenen auftretenden Determinanten berechnen.

Die Determinanten von $n \times n$-Matrizen lassen sich mittels der **Leibnizschen Darstellung** berechnen, d.h.

$$\det A := \sum_{\pi} (-1)^{j(\pi)} a_{1i_1}\, a_{2i_2} \cdots a_{ni_n}\,,$$

wobei die Summe über alle möglichen $n!$ Permutationen π der Zahlen $1, 2, \ldots, n$ zu berechnen ist. Der Wert des Ausdrucks $(-1)^{j(\pi)}$ ergibt sich aus der Anzahl $j(\pi)$ der Inversionen der Permutation $\pi = \begin{pmatrix} 1 & 2 & \ldots & n \\ i_1 & i_2 & \ldots & i_n \end{pmatrix}$.

Bemerkung 3.1.2 (Rechenoperationen) *Im Folgenden werden die Verknüpfungen Multiplikation, Addition, Division und Subtraktion in ihrem Rechenaufwand nicht unterschieden und unter dem Begriff* **Gleitkommaoperation** *zusammengefasst (1 Gleitkommaoperation $\simeq$ 1 FLOP[2]). Systematische Multiplikationen mit ± 1 bleiben im Allgemeinen unberücksichtigt. Anderweitige Operationen wie z.B. Wurzelziehen werden gesondert betrachtet.*

Satz 3.1.3 (Aufwand Leibnizsche Darstellung) *Sei $A \in \mathbb{R}^{n\times n}$. Der Aufwand zur Berechnung von $\det A$ mit der Leibnizschen Darstellung, d.h. als Summe über alle Permutationen der Menge $\{1, \ldots, n\}$, beträgt*

$$\text{FLOP}(\det A) = n\, n! - 1\,.$$

Beweis. Der Aufwand zur Berechnung von $\det A$ für $A \in \mathbb{R}^{n\times n}$ in der Leibnizschen Darstellung (unter Vernachlässigung der Multiplikation mit $(-1)^{j(\pi)}$) ergibt sich wie folgt:

$$\begin{aligned} \text{FLOP}(\det A) &= \text{„}(n! - 1) \text{ Additionen } + n! \text{ Produkte mit } n \text{ Faktoren "} \\ &= n! - 1 + n!(n-1) = n\, n! - 1\,. \end{aligned}$$

Damit ist die Aussage des Satzes bewiesen. □

Alternativ lässt sich die Determinante rekursiv mit Hilfe des **Laplaceschen Entwicklungssatzes** berechnen. Dieser lautet für eine quadratische Matrix $A \in \mathbb{R}^{n\times n}$

$$\det A = \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \det(A_{1j}),$$

wobei A_{1j} diejenige Matrix ist, die aus A durch Streichen der ersten Zeile und der j-ten Spalte entsteht.

Satz 3.1.4 (Aufwand Laplacescher Entwicklungssatz) *Sei $A \in \mathbb{R}^{n\times n}$ ($n \geq 2$). Der Aufwand zur Berechnung von $\det A$ mit dem Laplaceschen Entwicklungssatz beträgt*

$$\text{FLOP}(\det A) = \sum_{k=0}^{n} \frac{n!}{k!} - 2 < e\, n! - 2,$$

wobei $e = \exp(1)$ die Eulersche Zahl bezeichnet.

[2] **FLOP** (Floating point operation) ist nicht zu verwechseln mit **FLOP/s** oder **flops** (floating point operations per second), welches als Maßeinheit für die Geschwindigkeit von Computersystemen verwendet wird.

Beweis. Die Ungleichung ist klar. Die Gleichung lässt sich induktiv beweisen:

Induktionsanfang ($n = 2$): Die Determinante der Matrix $A = (a_{ij}) \in \mathbb{R}^{2\times 2}$ lautet $\det(A) = a_{11}a_{22} - a_{21}a_{12}$, d.h. der Aufwand beträgt 2 Multiplikationen und 1 Addition, also

$$\text{FLOP}\left(\det(\mathbb{R}^{2\times 2})\right) = 3 = \frac{2!}{0!} + \frac{2!}{1!} + \frac{2!}{2!} - 2\,.$$

Induktionsschritt ($n \to n+1$): Für $n \geq 2$ gilt

$$\begin{aligned}
\text{FLOP}\left(\det(\mathbb{R}^{(n+1)\times(n+1)})\right) &= \text{„}n \text{ Additionen } + (n+1) \text{ Multiplikationen} \\
&\quad + (n+1) \text{ Berechnungen von } \det(\mathbb{R}^{n\times n})\text{“} \\
&= n + (n+1) + (n+1)\cdot \text{FLOP}(\det(\mathbb{R}^{n\times n})) \\
&\overset{\text{IV}}{=} 2n + 1 + (n+1)\cdot\left(\sum_{k=0}^{n}\frac{n!}{k!} - 2\right) \\
&= 2n + 1 + \sum_{k=0}^{n}\frac{(n+1)!}{k!} - 2(n+1) \\
&= \sum_{k=0}^{n}\frac{(n+1)!}{k!} - 1 = \sum_{k=0}^{n+1}\frac{(n+1)!}{k!} - 2,
\end{aligned}$$

womit der Satz bewiesen ist. □

Beispiel 3.1.5 Ein einfaches Beispiel soll zeigen, dass man die Determinante überhaupt nur für sehr kleine allgemeine Matrizen der Dimension $n \ll 23$ mit dem Laplaceschen Entwicklungssatz berechnen kann.
Ein Jahr hat $365 \cdot 24 \cdot 60 \cdot 60 \approx 3 \cdot 10^7$ Sekunden. Geht man nun von einem schnellen Rechner[3] mit 3000 TFlops aus, so benötigt man zur Berechnung der Determinante einer 21×21-Matrix mit dem Laplaceschen Entwicklungssatz

$$\frac{\text{Anzahl der Operationen}}{\text{Gleitkommaoperationen pro Sekunde}}\,[s] = \frac{\sum_{k=0}^{21}\frac{21!}{k!} - 2}{3000\cdot 10^{12}}\,[s] \approx 46293\,[s] \approx 12.9\,[h]$$

bzw. für eine 25×25 -Matrix

$$\frac{\sum_{k=0}^{25}\frac{25!}{k!} - 2}{3000\cdot 10^{12}}\,[s] \approx 1.406\cdot 10^{10}\,[s] \approx 445.7 \text{ Jahre}\,.$$

Bemerkung 3.1.6 *Die Steigerung der Leistungsfähigkeit moderner Computer (Moore'sches Gesetz) ist bei weitem geringer als das Wachstum der Problemgröße mit steigendem n. Das Hoffen auf immer schnellere Rechner hilft nicht, um immer größere Probleme lösen zu können. Zum Vergleich: Im Jahr 2008 war der Supercomputer JUGENE vom Forschungszentrum Jülich weltweit Platz sechs und in Europa Platz eins mit 180 Flops. Für $n = 21$ brauchte man damals 214.3h, also immerhin eine Beschleunigung um den Faktor 16 in 4 Jahren. Trotzdem bleibt $n = 25$ auch in naher Zukunft unerreichbar.*

[3] Der weltweit auf Platz vier rangierende und zugleich schnellste europäische Supercomputer SuperMUC am Leibniz Rechenzentrum Garching bei München hat ca. 3100 TFlops (Stand Juni 2012). Ein aktueller PC hat eine Leistung von ca. 300 GFlops. T = Tera = 10^{12}, G = Giga = 10^9.

Bemerkung 3.1.7 *Bei der Cramerschen Regel ist zum Lösen eines linearen Gleichungssystems $Ax = b$ mit $A \in \mathbb{R}^{n\times n}$ und $b \in \mathbb{R}^n$ die Berechnung von $n+1$ Determinanten und n Quotienten notwendig. Der Aufwand zur Lösung eines linearen Gleichungssystems lässt sich somit zusammenfassen, wobei wir auf die Komplexität des Gauß-Verfahrens (siehe Seite 39) erst später eingehen werden.*

Cramersche Regel		Gauß-Elimination
Leibniz	Laplace	
$n\,(n+1)! - 1$	$\sum_{k=0}^{n} \frac{(n+1)!}{k!} - n - 2$	$\frac{4n^3 + 9n^2 - n}{6}$

Tab. 3.1: Aufwand für die Lösung eines linearen Gleichungssystems mit verschiedenen direkten Verfahren.

Bemerkung 3.1.8 *Für das Lösen eines LGS mit 21 Unbekannten mit der Cramerschen Regel und dem Laplaceschen Entwicklungssatz benötigt man auf einem der schnellsten Rechner mehr als 10 Tage, d.h. $22 \cdot 12.9\ [h] = 283.8\ [h] = 11.825\ [d]$ (22 verschiedene Determinanten im Zähler und eine im Nenner, vgl. Beispiel 3.1.5). Ein System mit 24 Unbekannten ist mit einem heutigen Supercomputer und der Cramerschen Regel nicht in einem Menschenleben zu lösen.*

Beispiel 3.1.9 (Vergleich Rechenaufwand) Aufwand zum Lösen eines linearen Gleichungssystems $Ax = b$ via Cramerscher Regel und Gauß-Verfahren.

Für einige n sei die Anzahl der notwendigen FLOPs wiedergegeben.

	Cramer/Leibniz	Cramer/Laplace	Gauß
n	$= n\,(n+1)! - 1$	$= \sum_{k=0}^{n} \frac{(n+1)!}{k!} - n - 2$	$= (4n^3 + 9n^2 - n)/6$
$n = 2$	11	11	11
$n = 3$	71	59	31
$n = 4$	479	319	66
$n = 5$	3599	1949	120
$n = 8$	2903039	986399	436
$n = 10$	399167999	108505099	815

Im Folgenden werden wir nun Verfahren beschreiben, die bereits für $n \geq 3$ effektiver als die Cramersche Regel sind. Mit Hilfe dieser Verfahren lassen sich auch Determinanten in polynomieller Zeit bestimmen. Daher wird die Cramersche Regel im Allgemeinen nur für $n = 2,\ 3$ verwendet. Der Vorteil der Cramerschen Regel ist jedoch die explizite Schreibweise, d.h. sie ist eine Formel für alle Fälle. Man spart sich bei ähnlichem Rechenaufwand (d.h. $n = 2, 3$) aufwendige Fallunterscheidungen (z.B. Pivotwahl) im Programm.

3.2 Gestaffelte Systeme

Betrachten wir zuerst besonders einfach zu lösende Spezialfälle. Am einfachsten ist sicherlich der Fall einer diagonalen Matrix A.

Diagonalmatrix

Man bezeichnet eine quadratische Matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ als **Diagonalmatrix**, falls $a_{ij} = 0$ für $i \neq j$, $i, j = 1, \ldots, n$ gilt. Häufig schreibt man eine Diagonalmatrix A mit Diagonaleinträgen $a_{11}, \ldots, a_{nn}$ auch einfach $A = \text{diag}(a_{11}, \ldots, a_{nn})$. Die Anwendung der Inversen einer Diagonalmatrix A mit Diagonalelementen $a_{ii} \neq 0$, $i = 1, \ldots, n$, auf einen Vektor b lässt sich als Pseudocode wie folgt schreiben:

Algorithmus 3.2.1: Lösen von $Ax = b$ mit Diagonalmatrix A

Sei $A \in \mathbb{R}^{n\times n}$ eine invertierbare Diagonalmatrix und $b \in \mathbb{R}^n$

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

for $j = 1, \ldots, n$

$x_j = b_j / a_{jj}$

end

Output $x = (x_1, \cdots, x_n)^T$

Eine einfache Matlab Realisierung ist im Folgenden dargestellt.

Man löse $Ax = b$ mit

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{pmatrix}, \; b = \begin{pmatrix} 5 \\ 3 \\ 2 \end{pmatrix}.$$

MATLAB-Beispiel:

```
>> A=diag([1,2,4]);
>> b=[5;3;2];
>> for j=1:3, x(j)=b(j)/A(j,j); end
>> x
x =
    5.0000    1.5000    0.5000
```

Dreiecksmatrix

Der nächstschwierigere Fall ist der einer **Dreiecksmatrix** A. Man spricht von einer (quadratischen) oberen Dreiecksmatrix $A = (a_{ij})$, falls $a_{ij} = 0$ für $i > j$, $i, j = 1, \ldots, n$ und von einer unteren Dreiecksmatrix $A = (a_{ij})$, falls $a_{ij} = 0$ für $i < j, i, j = 1, \ldots, n$, gilt. Häufig verwenden wir auch R für eine obere Dreiecksmatrix und L für eine untere Dreiecksmatrix.
Betrachten wir nun das „gestaffelte“ Gleichungssystem

$$\begin{array}{ccccccccc} r_{11}x_1 & + & r_{12}x_2 & + & \cdots & + & r_{1n}x_n & = & z_1 \\ & & r_{22}x_2 & + & \cdots & + & r_{2n}x_n & = & z_2 \\ & & & & \ddots & & \vdots & & \vdots \\ & & & & & & r_{nn}x_n & = & z_n \end{array} \tag{3.4}$$

oder in Matrix-Vektor-Schreibweise

$$Rx = z, \tag{3.5}$$

wobei $R = (r_{ij}) \in \mathbb{R}^{n\times n}$ gilt. Offenbar erhalten wir x durch sukzessive Auflösung des „gestaffelten“ Gleichungssystems, beginnend mit der n-ten Zeile:

$$\begin{array}{lcll} x_n & := & z_n/r_{nn} & , \text{ falls } \quad r_{nn} \neq 0 \\ x_{n-1} & := & (z_{n-1} - r_{n-1,n}x_n)/r_{n-1,n-1} & , \text{ falls } \quad r_{n-1,n-1} \neq 0 \\ \vdots & & \vdots & \vdots \\ x_k & := & (z_k - \sum_{i=k+1}^{n} r_{ki}x_i)/r_{kk} & , \text{ falls } \quad r_{kk} \neq 0 \\ \vdots & & \vdots & \vdots \\ x_1 & := & (z_1 - r_{12}x_2 - \cdots - r_{1n}x_n)/r_{11} & , \text{ falls } \quad r_{11} \neq 0. \end{array} \tag{3.6}$$

Für die obere Dreiecksmatrix R gilt, dass

$$\det R = r_{11} \cdot r_{22} \cdot \cdots \cdot r_{nn} \tag{3.7}$$

und daher

$$\det R \neq 0 \Longleftrightarrow r_{ii} \neq 0 \quad (i = 1, \ldots, n). \tag{3.8}$$

Der angegebene Algorithmus ist also wiederum genau dann anwendbar, wenn $\det R \neq 0$ (d.h., wenn R regulär ist), also unter der Bedingung des Existenz- und Eindeutigkeitssatzes (Satz 3.1.1).

Analog zum obigen Vorgehen lässt sich auch ein gestaffeltes lineares Gleichungssystem der Form

$$Lx = z$$

mit einer unteren Dreiecksmatrix $L \in \mathbb{R}^{n\times n}$ lösen. In diesem Fall beginnt man in der ersten Zeile mit der Berechnung von x_1 und arbeitet sich dann bis zur letzten Zeile zur Bestimmung von x_n vor.

Bemerkung 3.2.1 (Vorwärts-, Rückwärtssubstitution) *Das Lösen eines LGS mit oberer Dreiecksmatrix nennt man auch Rückwärtseinsetzen/-substitution (Index läuft „rückwärts“ von n nach 1), bzw. mit einer unteren Dreiecksmatrix auch Vorwärtseinsetzen/-substitution (Index läuft „vorwärts“ von 1 nach n).*

Satz 3.2.2 (Rechenaufwand Ax und $A^{-1}b$ – Dreiecksmatrix) *Sei $A \in \mathbb{R}^{n\times n}$ eine reguläre obere oder untere Dreiecksmatrix und $x, b \in \mathbb{R}^n$. Dann gilt*

$$\text{FLOP}(Ax) = n^2 \quad \textit{und} \quad \text{FLOP}(A^{-1}b) = n^2\,.$$

Bei der Berechnung von $A^{-1}b$ ist im Gegensatz zu Ax die Reihenfolge, in der die Zeilen „abgearbeitet” werden, festgelegt.

Beweis. Es genügt den Fall einer regulären oberen Dreiecksmatrix $R \in \mathbb{R}^{n\times n}$ zu betrachten. Für den Rechenaufwand zur Lösung von $Rx = b$ (oder die Anwendung von R^{-1} auf einen Vektor b) ergibt sich:

i) für die i-te Zeile: je $(n - i)$ Additionen und Multiplikationen sowie eine Division

ii) insgesamt für die Zeilen n bis 1: Betrachten wir zuerst die Summenformel

$$\sum_{i=1}^{n}(n - i) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = \frac{n^2 - n}{2}\,.$$

Mit i) erhält man insgesamt einen Aufwand von $2\,\frac{n^2-n}{2} + n = n^2$ Operationen.

Der zweite Teil der Aussage bleibt Ihnen als Übungsaufgabe überlassen. □

Man beachte allerdings, dass man allgemein für die Berechnung von $A^{-1}b$ zunächst die Inverse A^{-1} berechnen muß. Für ganz allgemeine reguläre Matrizen ist dies sehr aufwändig, weswegen wir unter der Schreibweise $A^{-1}b$ stets die Lösung des LGS mit gegebener rechten Seite b verstehen wollen und für diesen Fall schnelle Verfahren konstruieren wollen.

Aufgabe 3.2.3 Es sei $A \in \mathbb{R}^{n\times n}$ eine obere (oder untere) Dreiecksmatrix und $x \in \mathbb{R}^n$. Man zeige, dass das Matrix-Vektor-Produkt Ax mit n^2 Operationen berechnet werden kann.

Eine Matlab Realisierung zur Bestimmung von $R^{-1}b$, bei der die Matrix R zeilenweise durchlaufen wird, und ein Anwendungsbeispiel sind im Folgenden dargestellt.

MATLAB-Funktion: Rinvb.m

```
function x = Rinvb(R,b)
% compute solution of R * x = b with upper triangular matrix R
n = size(R,1);
x = zeros(n,1);
for j = n:-1:1
  for k=j+1:n
    b(j) = b(j) - R(j,k) * x(k);
  end
  x(j) = b(j) / R(j,j);
end
```

MATLAB-Beispiel:

Man löse $Rx = b$ mit

$$R = \begin{pmatrix} 4 & 1 & 1 \\ 0 & 3 & 2 \\ 0 & 0 & 1 \end{pmatrix}, \; b = \begin{pmatrix} 9 \\ 12 \\ 3 \end{pmatrix}.$$

```
>> R = [4,1,1;0,3,2;0,0,1];
>> b = [9;12;3];
>> x = Rinvb(R,b);
>> x'
ans =
     1     2     3
```

Eine alternative Realisierung, bei der die Matrix R spaltenweise durchlaufen wird, ist mit `Rinvb2.m` gegeben.

MATLAB-Funktion: Rinvb2.m

```
function x = Rinvb2(R,b)
% compute solution of R * x = b with upper triangular matrix R
n = size(R,1);
x = zeros(n,1);
for j = n:-1:1
  x(j) = b(j) / R(j,j);
  for k=1:j-1
    b(k) = b(k) - R(k,j) * x(j);
  end
end
```

Was ist der Unterschied? Machen Sie sich dies klar!

3.3 Gaußsche Eliminationsmethode

Gäbe es nun zu einer beliebigen Matrix A eine Zerlegung

$$A = L \cdot R\,, \tag{3.9}$$

so könnte ein beliebiges lineares Gleichungssystem $Ax = b$ durch eine Rückwärts- und Vorwärtssubstitution mittels der beiden Schritte

i) löse $Lz = b$,

ii) löse $Rx = z$,

gelöst werden. Die folgende Gaußsche Eliminationsmethode liefert gerade eine solche Zerlegung. Betrachten wir ein allgemeines lineares Gleichungssystem $Ax = b$ ($A \in \mathbb{R}^{n\times n}$ regulär, $b \in \mathbb{R}^n$)

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array} \tag{3.10}$$

und versuchen dies in ein gestaffeltes System umzuformen.
Durch die folgenden drei Äquivalenzoperationen

1. Vertauschung von Zeilen,

2. Multiplikation einer Zeile mit einem Skalar $\neq 0$,

3. Addition eines Vielfachen einer Zeile zu einer anderen,

wird die Lösungsmenge des Gleichungssystems (3.10) nicht verändert.
Wir setzen zunächst $a_{11} \neq 0$ voraus. Um (3.10) nun in ein gestaffeltes System umzuformen, muss die erste Zeile nicht verändert werden. Die restlichen Zeilen werden so modifiziert, dass in einem ersten Schritt die Koeffizienten vor x_1 verschwinden, d.h. die Variable x_1 aus den Gleichungen in

den Zeilen 2 bis n eliminiert wird.
So entsteht ein System der Art

$$\begin{array}{ccccccccc} a_{11}^{(1)}x_1 & + & a_{12}^{(1)}x_2 & + & \cdots & + & a_{1n}^{(1)}x_n & = & b_1^{(1)} \\ & & a_{22}^{(2)}x_2 & + & \cdots & + & a_{2n}^{(2)}x_n & = & b_2^{(2)} \\ & & \vdots & & & & \vdots & & \vdots \\ & & a_{n2}^{(2)}x_2 & + & \cdots & + & a_{nn}^{(2)}x_n & = & b_n^{(2)}\,, \end{array} \tag{3.11}$$

wobei $a_{11}^{(1)} := a_{11}, \ldots, a_{1n}^{(1)} := a_{1n}$ und $b^{(1)} := b_1$. Haben wir dies erreicht, so können wir das selbe Verfahren auf die letzten $(n-1)$ Zeilen anwenden und so rekursiv ein gestaffeltes System erhalten. Mit den Quotienten

$$l_{i1} = a_{i1}/a_{11} \qquad (i = 2, 3, \ldots, n) \tag{3.12}$$

sind die Elemente in (3.11) gegeben durch

$$\begin{aligned} a_{ik}^{(2)} &= a_{ik} - l_{i1}a_{1k}\,, \\ b_i^{(2)} &= b_i - l_{i1}b_1. \end{aligned} \tag{3.13}$$

Damit ist der erste Eliminationsschritt unter der Annahme $a_{11} \neq 0$ ausführbar. Die Matrix, die sich aus diesem ersten Eliminationsschritt ergibt, bezeichnen wir mit $A^{(2)}$.

Wenden wir auf diese Restmatrix die Eliminationsvorschrift erneut an, so erhalten wir eine Folge

$$A = A^{(1)} \to A^{(2)} \to \cdots \to A^{(n)} =: R$$

von Matrizen der speziellen Gestalt

$$A^{(k)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \ldots & & & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \ldots & & & a_{2n}^{(2)} \\ & & \ddots & & & \\ & & & a_{kk}^{(k)} & \ldots & a_{kn}^{(k)} \\ & & & \vdots & & \vdots \\ & & & a_{nk}^{(k)} & \ldots & a_{nn}^{(k)} \end{pmatrix} \tag{3.14}$$

mit einer $(n-k+1, n-k+1)$-Restmatrix, auf die wir den Eliminationsschritt

$$\boxed{\begin{array}{lcll} l_{ik} & := & a_{ik}^{(k)}/a_{kk}^{(k)} & \text{für} \quad i = k+1, \ldots, n \\ a_{ij}^{(k+1)} & := & a_{ij}^{(k)} - l_{ik}a_{kj}^{(k)} & \text{für} \quad i, j = k+1, \ldots, n \\ b_i^{(k+1)} & := & b_i^{(k)} - l_{ik}b_k^{(k)} & \text{für} \quad i = k+1, \ldots, n \end{array}} \tag{3.15}$$

ausführen können, wenn das **Pivotelement** (Dreh- und Angelpunkt) $a_{kk}^{(k)}$ nicht verschwindet. Da jeder Eliminationsschritt eine lineare Operation auf den Zeilen von A ist, lässt sich der Übergang von $A^{(k)}$ und $b^{(k)}$ zu $A^{(k+1)}$ und $b^{(k+1)}$ als Multiplikation mit einer Matrix $L_k \in \mathbb{R}^{n\times n}$ von links darstellen, d.h.

$$A^{(k+1)} = L_k A^{(k)}, \quad b^{(k+1)} = L_k b^{(k)}. \tag{3.16}$$

Die Matrix

$$L_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -l_{n,k} & & 1 \end{pmatrix} = I - \begin{pmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{n,k} \end{pmatrix} \cdot e_k^T =: I - \ell_k \cdot e_k^T , \qquad (3.17)$$

wobei e_k der k-te Einheitsvektor sei, hat die Eigenschaft, dass die Inverse L_k^{-1} aus L_k durch einen Vorzeichenwechsel in den Einträgen l_{ik} $(i > k)$ entsteht und für das Produkt der L_k^{-1} gilt

$$L := L_1^{-1} \cdot \ldots \cdot L_{n-1}^{-1} = \begin{pmatrix} 1 & 0 & \ldots & \ldots & 0 \\ l_{21} & 1 & \ddots & & \vdots \\ l_{31} & l_{32} & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ l_{n1} & \ldots & & l_{n,n-1} & 1 \end{pmatrix} . \qquad (3.18)$$

Zusammengefasst erhalten wir auf diese Weise das zu $Ax = b$ äquivalente gestaffelte System $Rx = z$ mit der oberen Dreiecksmatrix

$$R = L^{-1}A \qquad \text{und der rechten Seite} \qquad z = L^{-1}b. \qquad (3.19)$$

Das **Gaußsche Eliminationsverfahren** schreibt sich dann wie folgt:
Sei $A \in \mathbb{R}^{n\times n}$ und $b \in \mathbb{R}^n$. Berechne nacheinander

i) $L \cdot R = A$ (Zerlegung mit R obere und L untere Dreiecksmatrix),

ii) $Lz = b$ (Vorwärtseinsetzen bzw. -substitution),

iii) $Rx = z$ (Rückwärtseinsetzen bzw. -substitution).

Definition 3.3.1 (unipotente Matrix) *Eine untere oder obere Dreiecksmatrix, deren Diagonalelemente alle gleich eins sind, heißt unipotent.*

Definition 3.3.2 (Gaußsche Dreieckszerlegung, LR-Zerlegung) *Die oben genannte Darstellung $A = L \cdot R$ der Matrix A als Produkt einer unipotenten unteren Dreiecksmatrix L und einer oberen Dreiecksmatrix R heißt* **Gaußsche Dreieckszerlegung** *oder* **LR-Zerlegung** *von A.*

Satz 3.3.3 *Existiert die LR-Zerlegung einer quadratischen regulären Matrix A, so sind L und R eindeutig bestimmt.*

Beweis. Übung. □

Satz 3.3.4 (Rechenaufwand Gaußsche Dreieckszerlegung, Gaußsche Elimination)
Sei $A \in \mathbb{R}^{n\times n}$ regulär. Existiert die LR-Zerlegung, so gilt

$$\text{FLOP}(LR - Zerlegung) = \frac{4n^3 - 3n^2 - n}{6} .$$

Des Weiteren sei $b \in \mathbb{R}^n$. Dann gilt für die Gaußsche Elimination

$$\text{FLOP}(A^{-1}b) = \frac{4n^3 + 9n^2 - n}{6} ,$$

d.h. die Lösung von $Ax = b$ kann mit $(4n^3 + 9n^2 - n)/6$ FLOPs berechnet werden.

Bemerkung 3.3.5 *Die Aufwandsberechnung unterscheidet sich teilweise in der Literatur. Neuerdings werden + und · Operationen nicht mehr unterschieden!*

Beweis von Satz 3.3.4. Für den Rechenaufwand der Gaußschen Dreieckszerlegung gilt Folgendes:

i) für den k-ten Eliminationsschritt:
je $(n-k)$ Divisionen zur Berechnung der ℓ_{ik} und je $(n-k)^2$ Multiplikationen und Subtraktionen zur Berechnung der $a_{ij}^{(k+1)}$, d.h. $2\,(n-k)^2 + (n-k)$ Operationen

ii) für alle $n-1$ Eliminationsschritte erhalten wir durch Umindizierung und mit Summenformeln aus Analysis 1

$$\begin{aligned} & \sum_{k=1}^{n-1} \left((n-k) + 2\,(n-k)^2 \right) = \sum_{k=1}^{n-1} \left(k + 2\,k^2 \right) = \frac{n(n-1)}{2} + \frac{(2n-1)n(n-1)}{3} \\ = \; & \frac{2\,(2n-1)(n-1)n + 3n(n-1)}{6} = \frac{n(n-1)(4n-2+3)}{6} \\ = \; & \frac{n(n-1)(4n+1)}{6} = \frac{4n^3 - 3n^2 - n}{6} \end{aligned}$$

iii) insgesamt benötigt man zum Lösen von $Ax = b$ mittels Gauß-Elimination, d.h. LR-Zerlegung und Vorwärts-, Rückwärtssubstitution, d.h. mit dem Lösen von zwei Gleichungssystemen (siehe Satz 3.2.2)

$$\frac{4n^3 - 3n^2 - n}{6} + 2n^2 = \frac{4n^3 + 9n^2 - n}{6} \quad \text{Operationen.}$$

Somit sind die Behauptungen des Satzes bewiesen. □

Das Speicherschema für die Gauß-Elimination orientiert sich an der Darstellung von $A^{(k)}$. In die erzeugten Nullen des Eliminationsschritts können die l_{ik} eingetragen werden. Da die Elemente mit den Werten 0 oder 1 natürlich nicht eigens gespeichert werden müssen, kann die LR-Zerlegung mit dem Speicherplatz der Matrix A bewerkstelligt werden:

$$A \to \left(\begin{array}{c|ccc} a_{11}^{(1)} & \dots & \dots & a_{1n}^{(1)} \\ \hline l_{21} & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & \vdots & & \\ l_{n1} & a_{n2}^{(2)} & & a_{nn}^{(2)} \end{array} \right) \to \dots \to \left(\begin{array}{ccccc} a_{11}^{(1)} & \dots & & & a_{1n}^{(1)} \\ l_{21} & a_{22}^{(2)} & & & a_{2n}^{(2)} \\ \vdots & \ddots & & & \\ \vdots & & \ddots & & \\ l_{n1} & \dots & \dots & l_{n,n-1} & a_{nn}^{(n)} \end{array} \right)$$

$$\Longrightarrow L = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ l_{21} & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & 0 \\ l_{n1} & \dots & \dots & l_{n,n-1} & 1 \end{pmatrix}, \quad R = \begin{pmatrix} a_{11}^{(1)} & \dots & \dots & \dots & a_{1n}^{(1)} \\ 0 & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & \dots & 0 & a_{nn}^{(n)} \end{pmatrix}.$$

3.4 Pivot-Strategien

Bisher haben wir uns noch nicht mit der Frage beschäftigt, ob eine LR-Zerlegung immer existiert und ob eine Vertauschung von Zeilen bei $Ax = b$ zu einem anderen Ergebnis führt, wenn der Algorithmus mit endlicher Rechengenauigkeit durchgeführt wird. (Bei exakter Arithmetik sind beide

Ergebnisse gleich, bzw. Vertauschung von Zeilen in A führt zur gleichen Lösung). Betrachten wir hierzu zwei Beispiele:

Beispiel 3.4.1 i) **Permutation liefert LR–Zerlegung:** Sei $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Wie wir leicht sehen, gilt für die Eigenwerte $\lambda_{1,2} = \pm 1$ und somit $\det A \neq 0$. Falls eine Zerlegung existieren würde, gäbe es $a, b, c, d \in \mathbb{R}$ mit

$$A = \begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix} \begin{pmatrix} b & c \\ 0 & d \end{pmatrix} = \begin{pmatrix} b & c \\ ab & ac+d \end{pmatrix} .$$

Ein einfacher Vergleich der Koeffizienten zeigt jedoch (zuerst nach b und dann nach a), dass keine LR–Zerlegung existiert, obwohl nach Vertauschen der Zeilen trivialerweise eine LR–Zerlegung sofort gegeben ist. Die Frage, die sich nun stellt, lautet: Kann man für jede reguläre Matrix A durch Vertauschen ihrer Zeilen eine Matrix finden, für die dann eine LR–Zerlegung existiert?

ii) **Permutation liefert höhere Genauigkeit:** Berechnen wir die Lösung des folgenden linearen Gleichungssystems ($\epsilon > 0$)

$$\begin{aligned} \epsilon x_1 + x_2 &= 1 && (3.20) \\ x_1 + x_2 &= 2\,. && (3.21) \end{aligned}$$

Bei exakter Arithmetik erhalten wir

$$\tfrac{1}{\epsilon - 1} \begin{pmatrix} 1 & -1 \\ -1 & \epsilon \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \tfrac{1}{1-\epsilon} \begin{pmatrix} 1 \\ 1 - 2\epsilon \end{pmatrix}$$

d.h. $x_1 = \frac{1}{1-\epsilon}$, $x_2 = \frac{1-2\epsilon}{1-\epsilon}$. Für $2\epsilon <$ Rechengenauigkeit (d.h. $1 \pm 2\epsilon$ und 1 werden im Rechner durch dieselbe Zahl dargestellt) gilt nun

$$x_1 = 1, \quad x_2 = 1.$$

Für die Gauß-Elimination erhalten wir, wenn wir den Koeffizienten vor x_1 in (3.21) eliminieren, d.h. das $1/\epsilon$-fache von (3.20) subtrahieren

$$\begin{aligned} \epsilon x_1 + x_2 &= 1 \\ \left(1 - \tfrac{1}{\epsilon}\right) x_2 &= 2 - \tfrac{1}{\epsilon}\,. \end{aligned}$$

Somit ergibt sich für $2\epsilon <$ Rechengenauigkeit aus der letzten Gleichung $x_2 = 1$ und damit aus der ersten Gleichung $x_1 = 0$. Vertauschen wir jedoch 1. und 2. Zeile (bzw. eliminieren den Koeffizienten vor x_1 in (3.20)) so erhalten wir

$$\begin{aligned} x_1 + x_2 &= 2 \\ (1 - \epsilon)\, x_2 &= 1 - 2\epsilon\,. \end{aligned}$$

Dies liefert $x_1 = x_2 = 1$ bei $2\epsilon <$ Rechengenauigkeit.

MATLAB-Beispiel:

Dieses scheinbar theoretische Ergebnis aus Beispiel 3.4.1 können wir direkt in Matlab umsetzen. Da die Recheneinheiten heutiger Rechner meist 2 Stellen genauer rechnen als unsere bisherigen theoretischen Untersuchungen vermuten lassen, muss hier $\epsilon \leq$ Maschinengenauigkeit$/8$ gelten.

```
>> e = eps/8;  % eps/2 gen\"ugt nicht!
>> x(2) = (2 - 1/e) / (1 - 1/e);
>> x(1) = (1 - x(2)) / e
x =
     0     1
>> x(2) = (1 - 2*e) / (1 - e);
>> x(1) = 2 - x(2)
x =
     1     1
```

Das Vertauschen kann folglich nötig sein. Zum einen, damit eine LR-Zerlegung überhaupt existiert, zum anderen, um den „Rechenfehler“ bedingt durch Rundungsfehler möglichst klein zu halten. Daraus leiten sich besondere Strategien zur Wahl des Pivotelements ab.

Betrachten wir nun die Gauß-Elimination mit **Spaltenpivotstrategie**, welche theoretisch nur dann nicht zielführend sein kann, falls die Matrix A singulär ist.

Algorithmus 3.4.1: Gauß-Elimination mit Spaltenpivotstrategie

a) Wähle im Eliminationsschritt $A^{(k)} \to A^{(k+1)}$ ein $p \in \{k, \dots, n\}$, so dass

$$|a_{pk}^{(k)}| \geq |a_{jk}^{(k)}| \qquad \text{für } j = k, \dots, n\,.$$

Die Zeile p soll die Pivotzeile werden.

b) Vertausche die Zeilen p und k

$$A^{(k)} \to \widetilde{A}^{(k)} \quad \text{mit} \quad \widetilde{a}_{ij}^{(k)} = \begin{cases} a_{kj}^{(k)} & \text{falls} \quad i = p \\ a_{pj}^{(k)} & \text{falls} \quad i = k \\ a_{ij}^{(k)} & \text{sonst.} \end{cases}$$

Nun gilt

$$|l_{ik}| = \left| \frac{\widetilde{a}_{ik}^{(k)}}{\widetilde{a}_{kk}^{(k)}} \right| = \left| \frac{\widetilde{a}_{ik}^{(k)}}{a_{pk}^{(k)}} \right| \leq 1.$$

c) Führe den nächsten Eliminationsschritt angewandt auf $\widetilde{A}^{(k)}$ aus,

$$\widetilde{A}^{(k)} \to A^{(k+1)}.$$

Bemerkung 3.4.2 *Anstelle der Spaltenpivotstrategie mit Zeilentausch kann man auch eine Zeilenpivotstrategie mit Spaltentausch durchführen. Beide Strategien benötigen im schlimmsten Fall $\mathcal{O}(n^2)$ zusätzliche Operationen. In der Praxis werden im Gegensatz zum oben genannten Algorithmus die Zeile bzw. Spalte nicht umgespeichert, sondern besondere Indexvektoren verwendet (siehe Matlabfunktion* `mylu.m`*). Die Kombination von Spalten- und Zeilenpivotstrategie führt zur vollständigen Pivotsuche, bei der die gesamte Restmatrix nach dem betragsgrößten Eintrag durchsucht wird. Wegen des Aufwands $\mathcal{O}(n^3)$ wird dies jedoch so gut wie nie angewandt.*

MATLAB-Funktion: mylu.m

```
function [L,R,P] = mylu(A)
n = size(A,1);                          % get leading dimension of A
p = 1 : n;                              % pivot element vector
for k = 1 : n-1                         % consider k-th column
  [m,mptr] = max(abs(A(p(k:end),k)));   % find pivot element
  tmp = p(k);                           % interchange in vector p
  p(k) = p(k-1+mptr);
  p(k-1+mptr) = tmp;

  for j = k+1 : n                       % modify entries in
    A(p(j),k) = A(p(j),k)/A(p(k),k);    % compute  l_jk, store in A
    for i = k+1 : n                     % (n-k-1)*(n-k-1) submatrix
      A(p(j),i) = A(p(j),i) ...
                  - A(p(j),k)*A(p(k),i);
    end
  end
end
L = tril(A(p,:),-1)+eye(n);             % these lines could be
R = triu(A(p,:));                       % neglected, all information
P = eye(n);                             % already in A and p
P(:,p) = P;
```

Satz 3.4.3 *Es sei $A \in \mathbb{R}^{n\times n}$ regulär. Dann existiert vor dem k-ten Eliminationsschritt des Gauß-Algorithmus stets eine Zeilen-/Spaltenpermutation derart, dass das k-te Diagonalelement von Null verschieden ist. Bei Zeilenpermutation, d.h. Spaltenpivotisierung, sind alle Einträge von L vom Betrag kleiner oder gleich eins.*

Beweis. Nach Voraussetzung gilt $\det A \neq 0$. Angenommen, es sei $a_{11} = 0$. Dann existiert eine Zeilenvertauschung, so dass das erste Pivotelement $a_{11}^{(1)}$ von Null verschieden und das betragsgrößte Element in der Spalte ist, d.h.

$$0 \neq |a_{11}^{(1)}| \geq |a_{i1}^{(1)}| \qquad \text{für} \quad i = 1, \ldots, n,$$

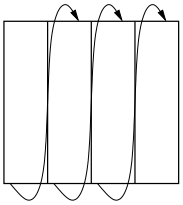
denn andernfalls wäre die Determinante von A in Widerspruch zur Voraussetzung gleich Null. Die Zeilenvertauschung hat dabei nur einen Vorzeichenwechsel in der Determinante zur Folge. Die Überlegungen für den ersten Eliminationsschritt übertragen sich sinngemäß auf die folgenden, reduzierten Systeme, bzw. ihre zugehörigen Determinanten. Diese Schlussfolgerungen gelten analog bei Zeilenpivotisierung. □

Eine unmittelbare Folge aus der Beweisführung des letzten Satzes ist das folgende Lemma.

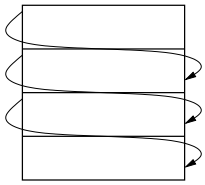
Lemma 3.4.4 *Erfolgen im Verlauf des Gauß-Algorithmus total m Zeilenvertauschungen (Spaltenvertauschungen), so ist die Determinante von A gegeben durch*

$$\det A = (-1)^m \prod_{k=1}^{n} r_{kk}.$$

Bemerkung 3.4.5 *Numerisch sind Spalten- und Zeilenpivotisierungen äquivalent. Die Auswahl hängt von der Speichermethode der Matrix A ab, d.h. man favorisiert die Spaltenpivotisierung, wenn die Einträge der Matrix spaltenweise im Speicher abgelegt sind, z.B. bei den Programmiersprachen Matlab, Fortran und die Zeilenpivotisierung u.a. bei C, denn hier sind die Matrizen als Folge von Zeilenvektoren abgelegt.*



Spaltenpivotisierung, bzw.



Zeilenpivotisierung.

Die genannten Pivotisierungen lassen sich formal durch die Multiplikation mit einer geeigneten Permutationsmatrix P beschreiben. Diese ist eine quadratische Matrix, welche in jeder Zeile und in jeder Spalte genau eine Eins und sonst Nullen enthält. Die Determinante ist $\det P = \pm 1$ und die Inverse ist durch $P^{-1} = P^T$ gegeben. Die Zeilenvertauschungen bei Spaltenpivotisierung entsprechen dabei der Multiplikation $P \cdot A$, analog lassen sich Spaltenpermutationen bei Zeilenpivotisierung durch $A \cdot P$ ausdrücken. Aus dem letzten Satz ergibt sich folgendes Resultat.

Satz 3.4.6 (Existenz einer Zerlegung $PA = LR$) *Zu jeder regulären Matrix A existiert eine Permutationsmatrix P, so dass $P \cdot A$ in ein Produkt $L \cdot R$ zerlegbar ist, d.h.*

$$P \cdot A = L \cdot R\,.$$

Ist nun immer eine Pivotisierung notwendig? Wir nennen im Folgenden ein Kriterium, welches leicht zu überprüfen ist.

Definition 3.4.7 (Diagonaldominanz) *Eine Matrix heißt diagonaldominant, falls gilt*

$$|a_{ii}| \geq \sum_{\substack{k=1 \\ k \neq i}}^{n} |a_{ik}| \qquad \forall\, i = 1, \ldots, n\,.$$

Sie heißt strikt diagonaldominant, falls für alle i „$>$" anstatt „$\geq$" gilt.

Satz 3.4.8 *Es sei $A \in \mathbb{R}^{n \times n}$ eine diagonaldominante und reguläre Matrix. Dann existiert eine Zerlegung*

$$A = L \cdot R.$$

Beweis. Einen Beweis dieser Aussage (in leicht abgewandelter Form) findet man z.B. bei [Schwarz]. Nach Voraussetzung ist entweder $|a_{11}| \geq \sum_{k=2}^{n} |a_{1k}| > 0$ oder es gilt $\sum_{k=2}^{n} |a_{1k}| = 0$ und $|a_{11}| > 0$; dies folgt aus der Regularität von A. Somit ist $a_{11} \neq 0$ ein zulässiges Pivotelement für den ersten Eliminationsschritt. Man muss nun zeigen, dass sich die Eigenschaft der diagonalen Dominanz auf das reduzierte Gleichungssystem überträgt. □

Zum Schluss dieses Abschnittes betrachten wir noch folgendes Beispiel zur konkreten Berechnung der LR-Zerlegung einer Matrix A.

Beispiel 3.4.9 (Berechnung von $PA = LR$) Sei

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

Diese Matrix ist offensichtlich nicht diagonaldominant, also wenden wir die Gauß-Elimination mit Spaltenpivotisierung an. Hierzu sei $P(i,j)$ diejenige Permutationsmatrix, die aus der Einheitsmatrix durch Vertauschen der i-ten und j-ten Zeile entsteht. Das Vertauschen der ersten mit der zweiten Zeile von A entspricht der Multiplikation der Matrix $P_1 := P(1,2)$ mit A von links, also

$$\widetilde{A}^{(1)} = P_1 A = \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ 1 & 1 & 1 & 1 \\ -2 & 1 & -2 & 1 \end{pmatrix}.$$

Der erste Schritt der LR-Zerlegung liefert

$$L_1 = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ -\frac{1}{2} & & 1 & \\ 1 & & & 1 \end{pmatrix}, \qquad L_1 P_1 A = \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ 0 & 2 & -1 & \frac{3}{2} \\ 0 & -1 & 2 & 0 \end{pmatrix}.$$

Bei der nächsten Spaltenpivotisierung sind keine Zeilenvertauschungen notwendig, also $P_2 = I$. Es folgt

$$L_2 = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & -1 & 1 & \\ & \frac{1}{2} & & 1 \end{pmatrix}, \qquad L_2 P_2 L_1 P_1 A = \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ 0 & 0 & 0 & \frac{7}{2} \\ 0 & 0 & \frac{3}{2} & -1 \end{pmatrix}.$$

Um nun auf eine obere Dreiecksgestalt zu kommen, müssen lediglich noch die dritte und vierte Zeile vertauscht werden, also formal $P_3 = P(3,4)$, $L_3 = I$ und damit

$$L_3 P_3 L_2 P_2 L_1 P_1 A = \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ 0 & 0 & \frac{3}{2} & -1 \\ 0 & 0 & 0 & \frac{7}{2} \end{pmatrix} =: R\,.$$

Aber wie sehen nun P und L aus, sodass

$$P \cdot A = L \cdot R \quad ?$$

Wir betrachten hierzu im allgemeinen Fall mit $A \in \mathbb{R}^{n\times n}$ invertierbar für $k = 1, \ldots, n-1$ die Matrizen

$$\widetilde{L}_k := P_n \cdot \ldots \cdot P_{k+1} L_k P_{k+1} \cdot \ldots \cdot P_n\,,$$

wobei $P_n := I$ gesetzt wird und $P_1, \ldots, P_{n-1}$ analog zu Beispiel 3.4.9 gewählt werden. Dann gilt

$$\begin{aligned} \widetilde{L}_k &= P_n \cdot \ldots \cdot P_{k+1} L_k P_{k+1} \cdot \ldots \cdot P_n \\ &\overset{(3.17)}{=} P_n \cdot \ldots \cdot P_{k+1} (I - \ell_k e_k^T) P_{k+1} \cdot \ldots \cdot P_n \\ &= I - \underbrace{P_n \cdot \ldots \cdot P_{k+1} \ell_k}_{=:\widetilde{\ell}_k} \underbrace{e_k^T P_{k+1} \cdot \ldots \cdot P_n}_{=e_k^T} \\ &= I - \widetilde{\ell}_k e_k^T\,. \end{aligned}$$

Da die Multiplikation mit den Matrizen $P_{k+1}, \ldots, P_n$ von links bzw. rechts nur Zeilen- bzw. Spaltenvertauschungen innerhalb der Zeilen bzw. Spalten $k+1, \ldots, n$ bewirkt, besitzt die Matrix $\widetilde{L}_k$ dieselbe Struktur wie L_k. Aus der Definition der $\widetilde{L}_k$ folgt nun

$$\widetilde{L}_{n-1} \cdot \ldots \cdot \widetilde{L}_1 P_{n-1} \cdot \ldots \cdot P_1 = L_{n-1} P_{n-1} \cdot \ldots \cdot L_2 P_2 L_1 P_1 \,.$$

Nach Konstruktion der L_k und P_k gilt schließlich

$$\widetilde{L}_{n-1} \cdot \ldots \cdot \widetilde{L}_1 P_{n-1} \cdot \ldots \cdot P_1 A = R \,.$$

Somit ergibt sich mit

$$L := \left(\widetilde{L}_{n-1} \cdot \ldots \cdot \widetilde{L}_1 \right)^{-1} = \widetilde{L}_1^{-1} \cdot \ldots \cdot \widetilde{L}_{n-1}^{-1}$$

und

$$P := P_{n-1} \cdot \ldots \cdot P_1$$

eine Zerlegung $PA = LR$.

Bemerkung 3.4.10 *Man beachte, dass mit den obigen Überlegungen ein alternativer, konstruktiver Beweis des Satzes 3.4.6 vorliegt.*

Führen wir nun Beispiel 3.4.9 weiter fort:

Beispiel 3.4.11 Zu der Matrix A aus Beispiel 3.4.9 sollen die Matrizen L und P der Zerlegung $PA = LR$ bestimmt werden. Wir erhalten

$$P = P_3 P_2 P_1 = P(3,4)\, I\, P(1,2) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$\widetilde{L}_1 = P_3 P_2 L_1 P_2 P_3 = P(3,4) \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ -\frac{1}{2} & & 1 & \\ 1 & & & 1 \end{pmatrix} P(3,4) = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 1 & & 1 & \\ -\frac{1}{2} & & & 1 \end{pmatrix},$$

$$\widetilde{L}_2 = P_3 L_2 P_3 = P(3,4) \begin{pmatrix} 1 & & & \\ & 1 & & \\ & -1 & 1 & \\ & \frac{1}{2} & & 1 \end{pmatrix} P(3,4) = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & \frac{1}{2} & 1 & \\ & -1 & & 1 \end{pmatrix},$$

$$\widetilde{L}_3 = P_4 L_3 P_4 = I \,,$$

$$L = \widetilde{L}_1^{-1} \widetilde{L}_2^{-1} \widetilde{L}_3^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 1 & 0 & 1 \end{pmatrix}$$

und somit

$$PA = \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ -2 & 1 & -2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ 0 & 0 & \frac{3}{2} & -1 \\ 0 & 0 & 0 & \frac{7}{2} \end{pmatrix} = LR \,.$$

3.5 Nachiteration

Die oben diskutierten Pivotstrategien schließen offenbar nicht aus, dass die so berechnete Lösung x immer noch „ziemlich ungenau" ist. Wie kann man nun x ohne großen Aufwand verbessern?

Häufig lässt sich dies durch eine **Nachiteration** mit Hilfe einer expliziten Auswertung des **Residuums** $r := b - A\widetilde{x}$ erreichen. Dabei bezeichne $\widetilde{x}$ die durch ein numerisches Verfahren berechnete Näherung an x. Ausgehend von $\widetilde{x}$ soll die exakte Lösung mittels des Korrekturansatzes

$$x = \widetilde{x} + s$$

ermittelt werden. Der Korrekturvektor ist so zu bestimmen, dass die Gleichungen erfüllt sind, d.h.

$$Ax - b = A(\widetilde{x} + s) - b = A\widetilde{x} + As - b = 0.$$

Der Korrekturvektor s ergibt sich somit als Lösung von

$$As = b - A\widetilde{x} = r. \tag{3.22}$$

Bei der numerischen Lösung dieser Korrekturgleichung erhalten wir im Allgemeinen eine wiederum fehlerhafte Korrektur $\widetilde{s} \neq s$. Trotzdem erwarten wir, dass die Näherungslösung

$$\widetilde{x} + \widetilde{s}$$

„besser" ist als $\widetilde{x}$.
Das Gleichungssystem (3.22) unterscheidet sich nur in der rechten Seite von dem ursprünglichen Problem $Ax = b$, sodass die Berechnung des Korrekturvektors s relativ wenig Aufwand erfordert, da beispielsweise bei Anwendung einer Nachiteration in Kombination mit der Gauß-Elimination schon die LR-Zerlegung der Koeffizientenmatrix A bekannt ist.

Bemerkung 3.5.1 *In der Praxis genügen bei Spalten- oder Zeilenpivotisierung meist wenige Nachiterationen, um eine Lösung auf eine dem Problem angepasste Genauigkeit zu erhalten.*

3.6 Cholesky-Verfahren

Wir wollen nun die Gauß-Elimination auf die eingeschränkte Klasse von Gleichungssystemen mit symmetrisch positiv definiten Matrizen anwenden. Es wird sich herausstellen, dass die Dreieckszerlegung in diesem Fall stark vereinfacht werden kann und dass dieser vereinfachte Algorithmus für große Matrizen nur den halben Aufwand an Operationen benötigt. Wir erinnern hier nochmals an die folgende Definition:

Definition 3.6.1 *Eine symmetrische Matrix $A \in \mathbb{R}^{n\times n}$ heißt **positiv definit**, wenn $x^T Ax > 0$ für alle $x \in \mathbb{R}^n$ mit $x \neq 0$ ist.*

Bemerkung 3.6.2 *Die Bedingung in Definition 3.6.1 macht auch für nicht-symmetrische Matrizen Sinn. Deshalb fordert man oft zusätzlich zur positiven Definitheit die Symmetrie und nennt solche Matrizen **symmetrisch positiv definit**, abgekürzt **s.p.d.***

Positiv definite Matrizen haben folgende Eigenschaften.

Satz 3.6.3 (Eigenschaften positiv definiter Matrizen) *Für jede s.p.d. Matrix $A \in \mathbb{R}^{n\times n}$ gilt:*

i) A ist invertierbar,

ii) $a_{ii} > 0 \qquad$ *für* $\quad i = 1, \ldots, n\,,$

iii) $\max\limits_{i,j=1,\ldots,n} |a_{ij}| = \max\limits_{i=1,\ldots,n} a_{ii}\,,$

iv) Bei der Gauß-Elimination ohne Pivotsuche ist jede Restmatrix wiederum positiv definit.

Beweis. Wäre A nicht invertierbar, gäbe es einen Eigenwert $\lambda = 0$, da $\det A = 0$ nach Annahme. Dies steht aber im Widerspruch zur Voraussetzung, dass A positiv definit ist, da es ansonsten einen Eigenvektor $x \neq 0$ zu $\lambda = 0$ mit $Ax = \lambda x$ gäbe und somit dann auch $x^T Ax = 0$ gälte. Somit ist i) bewiesen. Setzt man für x einen kanonischen Einheitsvektor e_i ein, so folgt gerade $a_{ii} = e_i^T A e_i > 0$ und daher gilt die Aussage ii). Behauptung iii) sei als Hausübung überlassen. Um die verbleibende Behauptung iv) zu beweisen, schreiben wir $A = A^{(1)}$ in der Form

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

wobei $z = (a_{12}, \ldots, a_{1n})^T$ sei. Nach einem Eliminationsschritt ergibt sich

$$A^{(2)} \;= L_1 A^{(1)} \;= \left(\begin{array}{c|c} a_{11} & z^T \\ \hline 0 & \\ \vdots & B^{(2)} \\ 0 & \end{array}\right) \quad \text{mit} \quad L_1 = \begin{pmatrix} 1 & & & \\ -l_{21} & 1 & & \\ \vdots & & \ddots & \\ -l_{n1} & & & 1 \end{pmatrix}.$$

Multipliziert man nun $A^{(2)}$ von rechts mit L_1^T, so wird auch z^T in der ersten Zeile eliminiert und die Teilmatrix $B^{(2)}$ bleibt unverändert, d.h.

$$L_1 A^{(1)} L_1^T = \left(\begin{array}{c|ccc} a_{11} & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & B^{(2)} & \\ 0 & & & \end{array}\right).$$

Damit ist bewiesen, dass die Restmatrix $B^{(2)}$ symmetrisch ist. Können wir nun noch zeigen, dass $B^{(2)}$ auch wiederum positiv definit ist, so können wir unsere Argumentation sukzessive fortsetzen. Es sei $y \in \mathbb{R}^{n-1}$ mit $y \neq 0$. Da L_1 regulär ist, gilt $x^T := \big(0 \mid y^T\big) L_1 \neq 0$. Nach Voraussetzung ist A positiv definit, also gilt

$$0 < x^T Ax = \big(0 \mid y^T\big) L_1 A L_1^T \begin{pmatrix} 0 \\ \hline y \end{pmatrix} = \big(0 \mid y^T\big) \left(\begin{array}{c|c} a_{11} & 0 \\ \hline 0 & B^{(2)} \end{array}\right) \begin{pmatrix} 0 \\ \hline y \end{pmatrix} = y^T B^{(2)} y\,.$$

Somit haben wir gezeigt, dass auch $B^{(2)}$ wieder positiv definit ist. □

Aufgabe 3.6.4 Man beweise Aussage iii) in Satz 3.6.3.

Mit Hilfe des letzten Satzes über die LR-Zerlegung können wir jetzt die **rationale** Cholesky-Zerlegung für symmetrische, positiv definite Matrizen herleiten.

Satz 3.6.5 (rationale Cholesky-Zerlegung) *Für jede symmetrisch positiv definite Matrix A existiert eine eindeutig bestimmte Zerlegung der Form*

$$A = \bar{L} D \bar{L}^T,$$

wobei $\bar{L}$ eine unipotente untere Dreiecksmatrix und D eine positive Diagonalmatrix ist.

Beweis. Wir setzen die Konstruktion im Beweis von Satz 3.6.3(iv) für $k = 2, \ldots, n-1$ fort und erhalten so unmittelbar $\bar{L}$ als Produkt der $L_1^{-1}, \ldots, L_{n-1}^{-1}$ und D als Diagonalmatrix der Pivotelemente. □

Bemerkung 3.6.6 *Man beachte die Eigenschaften der L_k bei der tatsächlichen Berechnung von L, d.h. das Produkt $L_1^{-1} \cdot \ldots \cdot L_{n-1}^{-1}$ ist nicht wirklich durch Multiplikation auszurechnen (vgl.* (3.18)*).*

Definition und Bemerkung 3.6.7 (Cholesky-Zerlegung) *Da alle Einträge der Diagonalmatrix D positiv sind, existiert $D^{1/2} = \mathrm{diag}(\pm\sqrt{d_i})$ und daher die Cholesky-Zerlegung*

$$A = L\,L^T,$$

wobei L die untere Dreiecksmatrix $L := \bar{L}D^{1/2}$ mit $\bar{L}$ und D aus der rationalen Cholesky-Zerlegung ist.

Bemerkung 3.6.8 *Die Cholesky-Zerlegung ist nicht eindeutig, da $D^{1/2}$ nur bis auf Vorzeichen der Elemente in der Diagonalen eindeutig bestimmt ist und somit auch $\overline{L} = LD^{1/2}$ nicht eindeutig ist.*

Zur Herleitung des Algorithmus zur Berechnung einer Cholesky-Zerlegung betrachten wir folgende Gleichung

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & & & \vdots \\ \vdots & & & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ & l_{22} & & l_{n2} \\ & & \ddots & \vdots \\ & & & l_{nn} \end{pmatrix} =$$

$$\begin{pmatrix} l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} & \cdots & l_{11}l_{n1} \\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & l_{21}l_{31} + l_{22}l_{32} & \cdots & l_{21}l_{n1} + l_{22}l_{n2} \\ l_{11}l_{31} & l_{21}l_{31} + l_{22}l_{32} & l_{31}^2 + l_{32}^2 + l_{33}^2 & \cdots & l_{31}l_{n1} + l_{32}l_{n2} + l_{33}l_{n3} \\ \vdots & \vdots & \vdots & & \vdots \\ l_{11}l_{n1} & l_{21}l_{n1} + l_{22}l_{n2} & l_{31}l_{n1} + l_{32}l_{n2} + l_{33}l_{n3} & \cdots & \sum_{k=1}^{n} l_{nk}^2 \end{pmatrix}$$

d.h.

$$\text{für } i = k \qquad \text{gilt } a_{kk} = \sum_{j=1}^{k} l_{kj}^2 \quad \text{und}$$

$$\text{für } i \neq k \qquad \text{gilt } a_{ik} = \sum_{j=1}^{k} l_{ij} \cdot l_{kj} \qquad (k+1 \leq i \leq n)$$

und werten diese in einer geschickten Reihenfolge aus.

Cholesky-Verfahren zur Berechnung von L mit $A = LL^T$
Spaltenweise berechnet man für $k = 1, 2, \ldots, n$

$$\begin{aligned} l_{kk} &= \Big(a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2\Big)^{1/2} \\ l_{ik} &= \frac{1}{l_{kk}}\Big(a_{ik} - \sum_{j=1}^{k-1} l_{ij} \cdot l_{kj}\Big) \qquad (k+1 \leq i \leq n). \end{aligned}$$

Bemerkungen 3.6.9 *i) Aus der Cholesky-Zerlegung $A = L \cdot L^T$ ergibt sich für $1 \le k \le n$ die Abschätzung*

$$\sum_{j=1}^{k} l_{kj}^2 \le \max_{1\le j\le n} |a_{jj}|.$$

Folglich sind alle Elemente der Matrix L betragsweise durch $\max\limits_{1\le j\le n} \sqrt{|a_{jj}|}$ beschränkt. Die Elemente können damit nicht allzu stark anwachsen, was sich günstig auf die Stabilität des Verfahrens auswirkt.

ii) Da A symmetrisch ist, wird nur Information oberhalb und einschließlich der Hauptdiagonalen benötigt. Wenn man die Diagonalelemente l_{kk} separat in einem Vektor der Länge n speichert, und die Elemente $l_{jk},\ k < j$ unterhalb der Diagonale, so kann man die Information der Matrix A bewahren.

iii) Bei der algorithmischen Durchführung der Cholesky-Zerlegung liefert das Verfahren auch die Information, ob die Matrix positiv definit ist. Man mache sich dies als Übungaufgabe klar!

Satz 3.6.10 (Rechenaufwand Cholesky-Zerlegung) *Sei $A \in \mathbb{R}^{n\times n}$ positiv definit. Dann gilt*

$$\text{FLOP}(\textit{Cholesky-Zerl. von } A) = \frac{2n^3+3n^2-5n}{6} + n\,\text{Wurzelberechnung} = \frac{1}{3}n^3 + \mathcal{O}(n^2)\,.$$

Beweis. Untersuchen wir nun die Komplexität der Cholesky-Zerlegung zuerst für einen k-ten Zerlegungsschritt und betimmen dann den Gesamtaufwand.

i) Für den k-ten Zerlegungsschritt, d.h. die Berechnung der Elemente l_{ik} für festen Spaltenindex, sind jeweils $(k-1)$ Multiplikationen, $(k-1)$ Subtraktionen und eine Wurzelberechnung zur Berechnung des Diagonalelements nötig. Für jedes Nebendiagonalelement werden $(k-1)$ Multiplikationen, $(k-1)$ Subtraktionen und eine Division benötigt, d.h. bei $(n-k)$ Elementen unterhalb des k-ten Diagonalelements sind dies in der Summe

$$\begin{aligned} & \big(2(k-1)+(n-k)(2k-1)\big)\ \text{FLOP und 1 Wurzelberechnung} \\ = \ & \big(-2k^2+(3+2n)k-(n+2)\big)\ \text{FLOP und 1 Wurzelberechnung.} \end{aligned}$$

ii) Insgesamt ergibt sich dann für die Summe über alle Zerlegungschritte

$$\begin{aligned} & n\,\text{Wurzelberechnungen} + \sum_{k=1}^{n}\big(-2k^2+(3+2n)k-(n+2)\big) \\ = \ & n\,\text{Wurzelberechnungen} + \Big(-2\,\frac{(2n+1)(n+1)n}{6} + (3+2n)\,\frac{(n+1)n}{2} - (n+2)n\Big) \\ = \ & n\,\text{Wurzelberechnungen} + \frac{-(4n^3+6n^2+2n)+(6n^3+15n^2+9n)-(6n^2+12n)}{6} \\ = \ & n\,\text{Wurzelberechnungen} + \frac{2n^3+3n^2-5n}{6}\,. \end{aligned}$$

Da eine einzelne Wurzelberechnung an Aufwand ein konstantes Vielfaches einer Gleitkommaoperation benötigt, kann man den Term n Wurzelberechnungen $+(3n^2-5n)/6$ zu $\mathcal{O}(n^2)$ Operationen zusammenfassen. □

Bemerkung 3.6.11 *Für große n ist der Aufwand des Cholesky-Verfahrens im Vergleich zum Gauß-Algorithmus ungefähr halb so groß.*

3.7 Bandgleichungen

Eine weitere Klasse spezieller Matrizen, welche beim Lösen linearer Gleichungssysteme eine besondere Rolle spielen, sind Bandmatrizen. Man spricht von einer Bandmatrix A, falls alle von Null verschiedenen Elemente a_{ik} in der Diagonale und in einigen dazu benachbarten Nebendiagonalen stehen. Für die Anwendungen sind insbesondere die symmetrischen, positiv definiten Bandmatrizen wichtig.

Definition 3.7.1 *Unter der* **Bandbreite** m *einer symmetrischen Matrix* $A \in \mathbb{R}^{n\times n}$ *versteht man die kleinste natürliche Zahl* $m < n$, *so dass gilt*

$$a_{ik} = 0 \qquad \text{für alle } i \text{ und } k \text{ mit } |i-k| > m.$$

Bemerkung 3.7.2 *Die Bandbreite* m *gibt somit die Anzahl der Nebendiagonalen unterhalb bzw. oberhalb der Diagonalen an, welche die im Allgemeinen von Null verschiedenen Matrixelemente enthalten.*

Bemerkung 3.7.3 *In verallgemeinerter Form spricht man auch von unterer und oberer Bandbreite, welche sich auf suggestive Weise definieren.*

Satz 3.7.4 *Die untere Dreiecksmatrix* L *der Cholesky-Zerlegung* $A = LL^T$ *einer symmetrischen, positiv definiten Bandmatrix mit der Bandbreite* m *besitzt dieselbe Bandstruktur, denn es gilt*

$$l_{ik} = 0 \qquad \text{für alle } i,k \quad \text{mit } i-k > m\,.$$

Beweis. Ergibt sich direkt aus einer Hausübung zur Hülle einer Matrix. □

Aufgabe 3.7.5 Man beweise Satz 3.7.4.

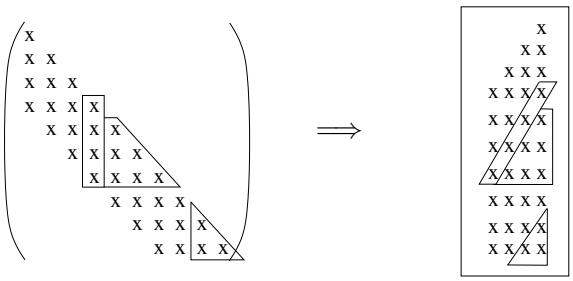


Abb. 3.3: Reduktion und Speicherung einer symmetrischen, positiv definiten Bandmatrix.

Analog zur Herleitung des Cholesky-Verfahrens auf Seite 49 erhalten wir die Rechenvorschrift des Cholesky-Verfahrens für Bandmatrizen, wobei nun die Koeffizienten zeilenweise bestimmt werden. Für $n = 5$ und $m = 2$ sieht die Situation wie folgt aus:

$$\begin{pmatrix} l_{11} & 0 & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 & 0 \\ 0 & l_{42} & l_{43} & l_{44} & 0 \\ 0 & 0 & l_{53} & l_{54} & l_{55} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} & 0 & 0 \\ 0 & l_{22} & l_{32} & l_{42} & 0 \\ 0 & 0 & l_{33} & l_{43} & l_{53} \\ 0 & 0 & 0 & l_{44} & l_{54} \\ 0 & 0 & 0 & 0 & l_{55} \end{pmatrix}$$

$$= \begin{pmatrix} l_{11}^2 & & \text{sym.} & & \\ l_{21}l_{11} & l_{21}^2 + l_{22}^2 & & & \\ l_{31}l_{11} & l_{31}l_{21} + l_{32}l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^2 & & \\ & l_{42}l_{22} & l_{42}l_{32} + l_{43}l_{33} & l_{42}^2 + l_{43}^2 + l_{44}^2 & \\ & & l_{53}l_{33} & l_{53}l_{43} + l_{54}l_{44} & l_{53}^2 + l_{54}^2 + l_{55}^2 \end{pmatrix}.$$

Cholesky-Verfahren zur Berechnung von L mit $A = LL^T$, wobei $A \in \mathbb{R}^{n\times n}$
mit Bandbreite $0 \le m < n$.
Zeilenweise berechnet man für $k = 1, 2, \ldots, n$

$$\begin{aligned} l_{ki} &= \frac{1}{l_{ii}}\Big(a_{ki} - \sum_{j=\max\{1,k-m\}}^{i-1} l_{kj}\cdot l_{ij}\Big) \qquad (\max\{1,k-m\} \le i \le k-1) \\ l_{kk} &= \Big(a_{kk} - \sum_{j=\max\{1,k-m\}}^{k-1} l_{kj}^2\Big)^{1/2} \end{aligned}$$

Satz 3.7.6 *Es sei $A \in \mathbb{R}^{n\times n}$ eine positiv definite Bandmatrix mit Bandbreite $1 \le m \le n$. Dann beträgt der Aufwand zur Berechnung der Cholesky-Zerlegung von A*

$$\text{FLOP}(\textit{Cholesky-Zerl. von } A) = n(m^2+2m) - \frac{4m^3+9m^2+5m}{6} + n \text{ Wurzelberechnungen}\,.$$

Beweis. Wendet man die Cholesky-Zerlegung auf eine positiv definite Bandmatrix A an, so ist die resultierende untere Dreiecksmatrix L mit $A = LL^T$ wieder eine Bandmatrix mit der gleichen Bandbreite, wie A sie hat.

i) Gehen wir davon aus, dass die Matrix L zeilenweise berechnet wird und betrachten zuerst den Aufwand für die ersten m Zeilen. Hier kann man das Ergebnis aus Satz 3.6.10 verwenden. Es sind hierfür somit

$$m \text{ Wurzelberechnungen} + \frac{2m^3+3m^2-5m}{6}$$

Operationen notwendig.

ii) In den verbleibenden $n-m$ Zeilen, in denen jeweils $m+1$ Koeffizienten zu bestimmen sind, ergibt sich Folgendes:
Um den j-ten von Null verschiedenen Nebendiagonaleintrag von L in der k-ten Zeile $m+1 \le k \le n$ zu berechnen, sind jeweils $j-1$ Multiplikationen, $j-1$ Subtraktionen und eine Division notwendig. Zur Berechnung des Diagonalelements werden m Multiplikationen, m Subtraktionen und eine Wurzelberechnung benötigt. Bei m Nebendiagonalelementen sind dies in der Summe

$$\begin{aligned} & 1 \text{ Wurzelberechnung} + 2m + \sum_{j=1}^{m}(2j-1) = 1\sqrt{\cdot} + m + 2\sum_{j=1}^{m} j \\ = \;& 1\sqrt{\cdot} + m + m(m+1) = 1\sqrt{\cdot} + m(m+2)\,. \end{aligned}$$

iii) Der gesamte Aufwand beträgt also

$$\begin{aligned} & n \text{ Wurzelberechnungen} + \frac{2m^3+3m^2-5m}{6} + (n-m)m(m+2) \\ = \;& n \text{ Wurzelberechnungen} + n(m^2+2m) - \frac{4m^3+9m^2+5m}{6}\,. \end{aligned}$$

□

Bemerkung 3.7.7 *Ist für eine Klasse von positiv definiten Bandmatrizen aus $\mathbb{R}^{n\times n}$ für beliebiges n die Bandbreite beschränkt, so wächst der Aufwand zur Berechnung der Cholesky-Zerlegung asymptotisch nur* ***linear*** *in n.*

Aufgabe 3.7.8 Man leite mit Hilfe des Cholesky-Verfahrens für symmetrische Bandmatrizen eine Rekursionsformel für s.p.d. Tridiagonalmatrizen her und gebe den Aufwand an.

3.8 Vandermond-Matrizen

Bei vielen Approximations- und Interpolationsproblemen treten sogenannte Vandermond-Matrizen auf, d.h. Matrizen $V \in \mathbb{R}^{(n+1)\times(n+1)}$ von der Form

$$V = \quad V(x_0, \ldots, x_n) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_n \\ \vdots & \vdots & & \vdots \\ x_0^n & x_1^n & \cdots & x_n^n \end{pmatrix}.$$

Wir bezeichnen mit $\mathbb{P}_n$ die Menge der Polynome vom Grad n. Sei z.B. zu $(n+1)$-Daten (x_i, f_i) ein Polynom n-ten Grades $p \in \mathbb{P}_n$ gesucht mit der Eigenschaft $p(x_i) = f_i$ $(i = 0, \ldots, n)$. Wählt man den Ansatz

$$p(x) = \sum_{j=0}^{n} a_j x^j \,,$$

so führt dies zu einem LGS

$$V^T a = f$$

mit $a = (a_0, \ldots, a_n)^T$ und $f = (f_0, \ldots, f_n)^T$. Man mache sich dies an einem einfachen Beispiel klar!

Diesen Zusammenhang zwischen dem Interpolationsproblem und dem System $V^T a = f$ macht man sich auch beim Lösen zu Nutze. Die Bestimmung des Koeffizientenvektors a bzw. das Lösen des Gleichungssystems gliedert sich dabei in zwei Teile.

- Zuerst bestimmt man das Interpolationspolynom p in der Newton–Darstellung, d.h.

$$\begin{aligned} p(x) &= \sum_{k=0}^{n} c_k \left(\prod_{j=0}^{k-1} (x - x_j) \right) \\ &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \ldots + c_n(x - x_0) \cdots (x - x_{n-1}) \end{aligned}$$

  Die Konstanten c_k werden als dividierte Differenzen bezeichnet, sie lösen das lineare Gleichungsystem

$$\begin{pmatrix} 1 & & & & \\ 1 & (x_1 - x_0) & & & \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) & & \\ \vdots & & & \ddots & \\ 1 & (x_n - x_0) & (x_n - x_0)(x_n - x_1) & \ldots & \prod_{j=0}^{n-1} (x_n - x_j) \end{pmatrix} \begin{pmatrix} c_0 \\ \\ \vdots \\ \\ c_n \end{pmatrix} = \begin{pmatrix} f_0 \\ \\ \vdots \\ \\ f_n \end{pmatrix}$$

  und können wie folgt bestimmt werden.

```
(c_0, ··· , c_n)^T = (f_0, ··· , f_n)^T
for  k = 0 : n − 1
    for  i = n : −1 : k + 1
        c_i = (c_i − c_{i−1})/(x_i − x_{i−k−1})
    end
end
```
(3.23)

- Im zweiten Schritt gewinnt man nun die a_i's aus den c_i's durch eine Basistransformation von $(1, (x-x_0), (x-x_0)(x-x_1), \ldots, \prod_{j=0}^{n-1}(x-x_j))$ nach $(1, x, x^2, \ldots, x^n)$. Die Transformationsmatrix U ergibt sich aus entsprechendem Koeffizientenvergleich

$$\begin{pmatrix} 1 & -x_0 & x_0x_1 & \cdots & \prod_{j=0}^{n-1}(-x_j) \\ & 1 & -(x_0+x_1) & & \\ & & 1 & & \vdots \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} c = a\,. \tag{3.24}$$

  Anstatt die Matrix U zu bestimmen, kann a wie folgt bestimmt werden.

$$\begin{array}{l} (a_0, \cdots, a_n)^T = (c_0, \cdots, c_n)^T \\ \texttt{for } k = n-1 : -1 : 0 \\ \quad \texttt{for } i = k : n-1 \\ \quad\quad a_i = a_i - x_k\, a_{i+1} \\ \quad \texttt{end} \\ \texttt{end} \end{array} \tag{3.25}$$

Die Kombination beider Schritte liefert den folgenden Algorithmus.

Algorithmus 3.8.1: Polynominterpolation: $V^T a = f$

Gegeben seien $x(0:n) \in \mathbb{R}^{n+1}$ mit paarweise verschiedenen Einträgen und $f = f(0:n) \in \mathbb{R}^{n+1}$. Der folgende Algorithmus überschreibt f mit der Lösung $a = a(0:n)$ zu dem Vandermondschen System

$$V(x_0, \ldots, x_n)^T a = f$$

```
for k = 0 : n-1
    for i = n : -1 : k+1
        f(i) = (f(i) - f(i-1))/(x(i) - x(i-k-1))
    end
end
for k = n-1 : -1 : 0
    for i = k : n-1
        f(i) = f(i) - x(k) f(i+1)
    end
end
```

Satz 3.8.1 *Das Verfahren 3.8.1 benötigt zum Lösen des Vandermondschen System*

$$V(x_0, \ldots, x_n)^T a = f$$

nur $5(n^2+n)/2$ *Operationen*.

Beweis. Die Anweisung in der ersten inneren FOR-Schleife von Algorithmus 3.8.1 wird für ein $k \in \{0, \ldots, n-1\}$ $(n-k)$-mal ausgeführt, insgesamt also $\sum_{k=0}^{n-1}(n-k) = \sum_{k=1}^{n} k = n(n+1)/2$

-mal. Hierbei werden jedesmal 3 Gleitkommaoperationen benötigt. Für die zweite Doppelschleife ergeben sich analog zu der obigen Betrachtung $n(n+1)/2$ Aufrufe der inneren Anweisung, für die jeweils 2 Gleitkommaoperationen benötigt werden. □

Bemerkung 3.8.2 *Man beachte, dass für spezielle vollbesetzte Matrizen Verfahren existieren, die weniger als $\mathcal{O}(n^3)$ Operationen zur Berechnung der Lösung benötigen. Auch die Matrix braucht nicht vollständig gespeichert werden.*

Das System $Vz = b$

Der Vollständigkeit wegen sei hier auch ein Verfahren für $Vz = b$ aufgeführt. Es sei $L_k(\alpha) \in \mathbb{R}^{(n+1)\times(n+1)}$ eine untere bidiagonale Matrix definiert durch

$$L_k(\alpha) = \left(\begin{array}{c|ccccccc} I_k & & & 0 & & & \\ \hline & 1 & & \dots & & 0 & \\ & -\alpha & 1 & & & & \\ & & \ddots & \ddots & & & \\ 0 & \vdots & & \ddots & \ddots & & \vdots \\ & & & & \ddots & 1 & \\ & 0 & & \dots & & -\alpha & 1 \end{array}\right)$$

und die Diagonalmatrix D_k sei definiert durch

$$D_k = \text{diag}(\underbrace{1,\dots,1}_{(k+1)\text{-mal}}, x_{k+1} - x_0, \dots, x_n - x_{n-k-1}) .$$

Mit diesen Definitionen kann man leicht aus (3.23) nachvollziehen, falls $f = f(0:n)$ und $c = c(0:n)$ der Vektor der dividierten Differenzen ist, dass $c = Lf$ gilt, wobei L die Matrix sei, die wie folgt definiert ist:

$$L = D_{n-1}^{-1} L_{n-1}(1) \cdots D_0^{-1} L_0(1) .$$

Ähnlich erhält man aus (3.25), dass

$$a = Uc \tag{3.26}$$

gilt, wobei die durch (3.26) definierte Matrix U sich auch definieren lässt durch

$$U = L_0(x_0)^T \cdots L_{n-1}(x_{n-1})^T .$$

Mit diesen Definitionen ergibt sich sofort die Aussage $a = ULf$ mit $V^{-T} = UL$. Die Lösung von $Vz = b$ ergibt sich somit durch

$$\begin{aligned} z &= V^{-1}b = L^T(U^T b) \\ &= \left(L_0(1)^T D_0^{-1} \cdots L_{n-1}(1)^T D_{n-1}^{-1}\right)\ \left(L_{n-1}(x_{n-1}) \cdots L_0(x_0)\right)\ b . \end{aligned}$$

Diese Erkenntnis führt zu dem folgenden Algorithmus.

Algorithmus 3.8.2: Vandermondsches System: $Vz = b$

Gegeben seien $x(0:n) \in \mathbb{R}^{n+1}$ mit paarweise verschiedenen Einträgen und $b = b(0:n) \in \mathbb{R}^{n+1}$. Der folgende Algorithmus überschreibt b mit der Lösung $z = z(0:n)$ zu dem Vandermondschen System

$$V(x_0, \ldots, x_n)z = b\,.$$

```
for k = 0 : n − 1
  for i = n : −1 : k + 1
      b(i) = b(i) − x(k) b(i − 1)
   end
end
for k = n − 1 : −1 : 0
   for i = k + 1 : n
      b(i) = b(i)/(x(i) − x(i − k − 1))
   end
   for i = k + 1 : n
      b(i) = b(i) − b(i + 1)
   end
end
```

Satz 3.8.3 *Das Verfahren 3.8.2 benötigt zum Lösen des Vandermondschen System*

$$V(x_0, \ldots, x_n)a = b$$

nur $5(n^2 + n)/2$ *Operationen*.

Beweis. Analog zu den Betrachtungen im Beweis von Satz 3.8.1 erkennt man, dass die Anweisungen in den inneren FOR-Schleifen von Algorithmus 3.8.2 jeweils $n(n+1)/2$ -mal aufgerufen werden. Jedesmal werden 2 bzw. 1 Gleitkommaoperation benötigt. □

3.9 Fehlerabschätzung mittels Kondition

Wir wollen nun zwei wichtige Fragestellungen untersuchen, welche bei dem numerischen Lösen von linearen Gleichungssystemen (LGS) auftreten. Zum einen betrachten wir eine Näherungslösung $\widetilde{x}$ zu der Lösung x von $Ax = b$. Welche Rückschlüsse kann man von der Größe des **Residuums** $r := b - A\widetilde{x}$ auf den Fehler $e = x - \widetilde{x}$ ziehen? Des Weiteren rechnet man im Allgemeinen mit endlicher Genauigkeit auf einem Rechner. Welche Einflüsse haben dabei Störungen der Ausgangsdaten auf die Lösung x?

Zu einer Matrix $A \in \mathbb{R}^{n\times n}$ und einem Vektor $x \in \mathbb{R}^n$ sei $\|A\|$ eine beliebige Matrixnorm[4] und $\|x\|$ eine dazu verträgliche Vektornorm (d.h. $\|Ax\| \le \|A\|\,\|x\|$). Nach Definition des Fehlers e und Residuums r gilt

$$Ae = Ax - A\widetilde{x} = b - A\widetilde{x} = r.$$

Daraus folgt

$$\|e\| = \|A^{-1}r\| \le \|A^{-1}\|\,\|r\|.$$

Mit

$$\|b\| = \|Ax\| \le \|A\|\,\|x\|$$

[4] Vgl. hierzu Anhang B.

folgt für den relativen Fehler die Abschätzung

$$\frac{\|e\|}{\|x\|} \le \frac{\|A^{-1}\|\,\|r\|}{\|b\|\cdot\frac{1}{\|A\|}} = \|A^{-1}\|\,\|A\|\cdot\frac{\|r\|}{\|b\|}\,. \tag{3.27}$$

Dies motiviert den Begriff der Konditionszahl.

Definition 3.9.1 (Konditionszahl) *Man bezeichnet*

$$\kappa(A) = \|A\|\,\|A^{-1}\| \tag{3.28}$$

als die **Konditionszahl** *der Matrix A bezüglich der verwendeten Matrixnorm.*

Beispiel 3.9.2 Man betrachte das LGS $Ax = b$ mit

$$A = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix} \qquad b := (b_j), \qquad b_j := \sum_{k=1}^{5} \frac{1}{k+j-1} \qquad (1 \le j \le 5)\,.$$

Für die Matrixnormen und die Konditionszahlen erhält man in den Fällen $p = 1, 2, \infty$

$$\begin{array}{lcllcllcl} \|A\|_1 &=& 2.28\overline{3}\,, & \|A^{-1}\|_1 &=& 413280\,, & \kappa_1(A) &=& 943656\,, \\ \|A\|_2 &\approx& 1.5670507\,, & \|A^{-1}\|_2 &\approx& 304142.84\,, & \kappa_2(A) &\approx& 476607.25\,, \\ \|A\|_\infty &=& 2.28\overline{3}\,, & \|A^{-1}\|_\infty &=& 413280\,, & \kappa_\infty(A) &=& \kappa_1(A)\,. \end{array}$$

Sei die exakte Lösung $x = (1, \ldots, 1)^T$. Ist nun $\widetilde{x} = (1-\varepsilon)x$, dann gilt

$$\frac{\|e\|}{\|x\|} = \frac{\|\varepsilon x\|}{\|x\|} = |\varepsilon|$$

und

$$r = b - A\widetilde{x} = Ax - A\widetilde{x} = A(x - (1-\varepsilon)x) = \varepsilon Ax = \varepsilon b,$$

d.h.

$$\frac{\|r\|}{\|b\|} = |\varepsilon|\,.$$

Obwohl die Konditionzahl sehr groß ist, verhält sich bei dieser Störung $\frac{\|e\|}{\|x\|}$ wie $\frac{\|r\|}{\|b\|}$. Obige Abschätzung ist offensichtlich eine „worst case“ Abschätzug. Betrachten wir nun

$$\widetilde{x} = (0.993826,\ 1.116692,\ 0.493836,\ 1.767191,\ 0.623754)^T. \tag{3.29}$$

A habe die Eigenwerte $0 \le \lambda_1 < \cdots < \lambda_5$ mit den zugehörigen Eigenvektoren $\varphi_1, \ldots, \varphi_5$, $\|\varphi_j\|_2 = 1$. Man beachte, dass $\widetilde{x}$ in (3.29) so gewählt ist, dass $\widetilde{x} - x \approx \varphi_1$ gilt und

$$x \approx -0.004\,\varphi_1 - 0.042\,\varphi_2 + 0.244\,\varphi_3 + 0.972\,\varphi_4 + 1.998\,\varphi_5\,,$$

d.h. x von φ_5 dominiert wird. Dann erhalten wir

$$\frac{\|e\|_1\|b\|_1}{\|x\|_1\|r\|_1} \approx 0.41\ \kappa_1(A), \quad \frac{\|e\|_2\|b\|_2}{\|x\|_2\|r\|_2} \approx 0.89\ \kappa_2(A), \quad \frac{\|e\|_\infty\|b\|_\infty}{\|x\|_\infty\|r\|_\infty} \approx 0.74\ \kappa_\infty(A)$$

und schätzen dann in der richtigen Größenordung ab!

Aufgabe 3.9.3 Es sei $A \in \mathbb{R}^{n\times n}$ eine positiv definite Matrix. Die Eigenwerte von Matrix A seien $0 < \lambda_1 \leq \ldots \leq \lambda_n$ mit den zugehörigen Eigenvektoren $\varphi_1, \ldots, \varphi_n$ ($\|\varphi_j\|_2 = 1, j = 1, \ldots, n$).

i) Es sei $x \in \mathbb{R}^n \setminus \{0\}$, $b := Ax$ und $\tilde{x} = cx$ ($0 \neq c \in \mathbb{R}$). Man zeige

$$\frac{\|x - \tilde{x}\|}{\|x\|} \frac{\|b\|}{\|b - A\tilde{x}\|} = 1\,.$$

ii) Man zeige

$$\kappa_2(A) = \lambda_n/\lambda_1\,.$$

iii) Es sei $x = c\,\varphi_n$, $c \neq 0$, $b := Ax$ und $\tilde{x} = x + \varphi_1$. Man zeige

$$\frac{\|x - \widetilde{x}\|}{\|x\|} \frac{\|b\|}{\|b - A\widetilde{x}\|} = \kappa_2(A)\,.$$

Untersuchen wir nun, welchen Einfluss kleine Störungen in den Ausgangsdaten A, b auf die Lösung x des linearen Gleichungssystems haben können, d.h. wir sind interessiert an der **Empfindlichkeit** der Lösung x auf Störungen in den Koeffizienten. Die genaue Frage lautet:
Wie groß kann die Änderung δx der Lösung x von $Ax = b$ sein, falls die Matrix um δA und b durch δb gestört sind? Dabei seien δA und δb kleine Störungen, so dass $A + \delta A$ immer noch regulär ist. Es sei $x + \delta x$ die Lösung zu

$$(A + \delta A)(x + \delta x) = (b + \delta b).$$

Teilweises Ausmultiplizieren liefert

$$Ax + \delta A\, x + (A + \delta A)\delta x = b + \delta b$$

und somit mit $Ax = b$

$$\begin{aligned}\delta x &= (A + \delta A)^{-1}(\delta b - \delta A\, x)\\ &= (I + A^{-1}\delta A)^{-1}A^{-1}(\delta b - \delta A\, x).\end{aligned}$$

Für eine submultiplikative Matrixnorm und eine dazu verträgliche Vektornorm ergibt sich somit

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

und wir erhalten die Abschätzung

$$\frac{\|\delta x\|}{\|x\|} \leq \|(I + A^{-1}\delta A)^{-1}\|\, \|A^{-1}\| \left(\frac{\|\delta b\|}{\|x\|} + \|\delta A\|\right). \tag{3.30}$$

Wir zeigen in Lemma 3.9.6, dass für $\|B\| < 1$ die Abschätzung

$$\|(I + B)^{-1}\| \leq \frac{1}{1 - \|B\|} \quad \text{existiert.}$$

Setzen wir nun $B = A^{-1}\delta A$ und $\|A^{-1}\|\, \|\delta A\| < 1$ (d.h. δA ist eine kleine Störung von A) voraus und erweitern auf der rechten Seite mit $\|A\|$, so erhalten wir unter Ausnutzung der Verträglichkeit der Normen und mit $Ax = b$

$$\begin{aligned}\frac{\|\delta x\|}{\|x\|} &\leq \frac{\|A^{-1}\|\, \|A\|}{1 - \|A^{-1}\delta A\|} \left(\frac{\|\delta b\|}{\|A\|\, \|x\|} + \frac{\|\delta A\|}{\|A\|}\right)\\ &\leq \frac{\|A^{-1}\|\, \|A\|}{1 - \|A^{-1}\|\, \|\delta A\|} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|}\right).\end{aligned}$$

Wir halten das Ergebnis nun in einem Satz fest.

Satz 3.9.4 *Es sei $A \in \mathbb{R}^{n\times n}$ regulär und $x \in \mathbb{R}^n$ die exakte Lösung von $Ax = b$. Die rechte Seite b sei um δb gestört und für die Störung von A gelte $\kappa(A)\,\|\delta A\|/\|A\| = \|A^{-1}\|\,\|\delta A\| < 1$ und $A + \delta A$ ist immer noch regulär. Dann gilt für die Lösung $\widetilde{x}$ des gestörten Systems mit Koeffizientenmatrix $A + \delta A$ und rechter Seite $b + \delta b$*

$$\frac{\|x - \widetilde{x}\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right) . \tag{3.31}$$

Bemerkung 3.9.5 *Die Konditionszahl $\kappa(A)$ der Koeffizientenmatrix A ist folglich die entscheidende Größe, welche die Empfindlichkeit der Lösung bzgl. der Störungen δA und δb beschreibt.*

Es bleibt folgendes Lemma zu beweisen:

Lemma 3.9.6 (Neumann-Reihe) *Sei $A \in \mathbb{R}^{n\times n}$, $\|\cdot\|$ eine submultiplikative Matrixnorm und $\|A\| < 1$. Dann ist $(I - A)$ regulär und*

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k \qquad \textit{(Neumann-Reihe)} \tag{3.32}$$

mit

$$\|(I - A)^{-1}\| \leq \frac{1}{1 - \|A\|} . \tag{3.33}$$

Beweis. Mit $\|A\| < 1$, der Submultiplikativität und der Summenformel für die geometrische Reihe erhält man

$$\sum_{k=0}^{m} \|A^k\| \leq \sum_{k=0}^{m} \|A\|^k \leq \sum_{k=0}^{\infty} \|A\|^k = \frac{1}{1 - \|A\|} < \infty \quad (m \in \mathbb{N}) . \tag{3.34}$$

Der Raum $\mathbb{R}^{n\times n}$ ist isomorph zu $\mathbb{R}^{n^2}$ (siehe Anhang B.2). In [Analysis II, Beispiel 8.4.6] wurde gezeigt, dass $\mathbb{R}^{n^2}$ vollständig ist bezüglich irgendeiner Norm auf $\mathbb{R}^{n^2}$. Somit ist $\mathbb{R}^{n\times n}$ vollständig bezüglich $\|\cdot\|$ und aus der absoluten Konvergenz folgt die Konvergenz von $\sum_{k=0}^{\infty} A^k$. Ebenso folgt aus $\|A^k\| \leq \|A\|^k \to 0$ für $k \to \infty$ die Konvergenz von $\lim\limits_{k\to\infty} A^k = 0$. Weiter gilt die Gleichung („Teleskopsumme “)

$$\left(\sum_{k=0}^{m} A^k \right) (I - A) = I - A^{m+1} \quad (m \in \mathbb{N}) . \tag{3.35}$$

Der Grenzübergang von (3.35) führt zur Gleichung

$$\left(\sum_{k=0}^{\infty} A^k \right) (I - A) = I. \tag{3.36}$$

Das bedeutet, $I - A$ ist regulär und $(I - A)^{-1} = \sum_{k=0}^{\infty} A^k$. Mit der Summenformel für die geometrische Reihe erhält man schließlich

$$\|(I - A)^{-1}\| \leq \lim_{N\to\infty} \sum_{k=0}^{N} \|A^k\| \leq \lim_{N\to\infty} \sum_{k=0}^{N} \|A\|^k = \frac{1}{1 - \|A\|} , \tag{3.37}$$

womit der Satz bewiesen ist. □

Zum Ende des Kapitels wollen wir nun den praktischen Nutzen der Abschätzung (3.31) in einer Faustregel festhalten.

Bei einer d-stelligen dezimalen Gleitkommarechnung können die relativen Fehler der Ausgangsgrößen für beliebige, kompatible Normen von der Größenordnung

$$\frac{\|\delta A\|}{\|A\|} \approx 5 \cdot 10^{-d} \qquad \frac{\|\delta b\|}{\|b\|} \approx 5 \cdot 10^{-d}$$

sein. Ist die Konditionszahl $\kappa(A) \approx 10^{\alpha}$ mit $5 \cdot 10^{\alpha-d} \ll 1$, so ergibt die Abschätzung (3.31)

$$\frac{\|\delta x\|}{\|x\|} \leq 10^{\alpha-d+1}.$$

Das heißt, dass $\|\delta x\|$ maximal in der Größenordnung der $(d-\alpha-1)$-ten Dezimalstelle von $\|x\|$ liegen kann und dies motiviert folgende Daumenregel:

Bemerkung 3.9.7 (Daumenregel zur Genauigkeit) *Wird $Ax = b$ mit d-stelliger dezimaler Gleitkommarechnung gelöst, und beträgt die Konditionszahl $\kappa(A) \approx 10^{\alpha}$, so sind, bezogen auf die betragsgrößte Komponente, nur $(d-\alpha-1)$ Dezimalstellen sicher.*

MATLAB-Beispiel:

Es ist bekannt, dass die Gauß-Elemination selbst mit Spaltenpivotstrategie zu überraschend ungenauen Ergebnissen beim Lösen von linearen Gleichungssystemen führen kann, obwohl die Matrix gut konditioniert ist.

Betrachten wir hierzu die von Wilkinson angegebene pathologische Matrix

$$A = \begin{pmatrix} 1 & & & 1 \\ -1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ -1 & \cdots & -1 & 1 \end{pmatrix}.$$

```
>> A=toeplitz([1,-ones(1,59)], ...
              [1,zeros(1,59)]);
>> A(:,60)=1;
>> cond(A)
ans =
   26.8035  % rel. gut konditioniert
>> randn('state', 3383)
>> x=randn(60,1);
>> b=A*x;
>> x1=A\b;
>> norm(x-x1)/norm(x)
ans =
    0.3402  % großer rel. Fehler
```

Bemerkung 3.9.8 *Das Beispiel lässt vermuten, dass das Gauß-Verfahren über die ganze Menge der invertierbaren Matrizen betrachtet nicht stabil ist. Für die in der Praxis auftretenden Matrizen, ist das Gauß-Verfahren mit Spaltenpivotierung jedoch „in der Regel" stabil. Für eine weitere Stabilitätsanalyse des Gauß-Verfahrens sei auf [Deuflhard/Hohmann], die grundlegenden Artikel von Wilkinson [Wilkinson65, Wilkinson69] sowie auf die Übersichtsartikel [Higham, Discroll/Maki] verwiesen.*