

Numerische Mathematik

Oliver Ernst

Professur Numerische Mathematik

Sommersemester 2015



TECHNISCHE UNIVERSITÄT
CHEMNITZ

① Einführung und Begriffe

- 1.1 Mathematische Modellbildung und numerische Simulation am Beispiel eines Wasserkreislaufs
- 1.2 Linearisierung und Iterationsverfahren am Beispiel des Newton-Verfahrens
- 1.3 Diskretisierung und Stabilität am Beispiel der Wärmeleitungsgleichung

② Gleitpunktarithmetik und Fehleranalyse

- 2.1 Gleitpunktzahlen
- 2.2 Rundung
- 2.3 Der IEEE-754 Standard
- 2.4 Korrekt gerundete Gleitpunktarithmetik
- 2.5 Numerische Stabilität und Fehleranalyse
- 2.6 Ein Beispiel

③ Direkte Verfahren zur Lösung linearer Gleichungssysteme

- 3.1 Vorbemerkungen
- 3.2 Störungstheorie
- 3.3 Das Lösen von Dreieckssystemen
- 3.4 Gauß-Elimination
- 3.5 Pivotisierung

- 3.6 Equilibrierung und Nachiteration
- 3.7 Stabilität bei der Gauß-Elimination

4 Direkte Verfahren für spezielle Systeme

- 4.1 Die Cholesky-Zerlegung

- 1 Einführung und Begriffe
- 2 Gleitpunktarithmetik und Fehleranalyse
- 3 Direkte Verfahren zur Lösung linearer Gleichungssysteme
- 4 Direkte Verfahren für spezielle Systeme

- ① Einführung und Begriffe
- ② Gleitpunktarithmetik und Fehleranalyse
- ③ Direkte Verfahren zur Lösung linearer Gleichungssysteme
 - 3.1 Vorbemerkungen
 - 3.2 Störungstheorie
 - 3.3 Das Lösen von Dreieckssystemen
 - 3.4 Gauß-Elimination
 - 3.5 Pivotisierung
 - 3.6 Equilibrierung und Nachiteration
 - 3.7 Stabilität bei der Gauß-Elimination
- ④ Direkte Verfahren für spezielle Systeme

Einige Stimmen zum Ansinnen, lineare Gleichungssysteme mit Gauß-Elimination auf Computern zu lösen:

„A 78-rowed Matrix would need to be carried to no less than 46 places to ensure even an approximate accuracy in the first decimal place.“

Harold Hotelling, 1943³

„Very little is known about the methods so far described, [but] what little information there is tends to indicate that these methods are unstable and that rounding errors accumulate so seriously that the methods are impractical for large values of n .“

Bargmann, Montgomery and von Neumann, 1946⁴

³Some New Methods in Matrix Calculation. Ann. Math. Statist. 14(1) 1943, 1–34

⁴Solution of linear systems of high order. in John von Neumann Collected Works, Vol. 5, A. H. Taub, ed., Macmillan, New York, 1946, pp. 421–477.

Bezeichnungen:

$$\begin{array}{ccccccccccc} a_{1,1} & x_1 & + & a_{1,2} & x_2 & + & \cdots & + & a_{1,n} & x_n & = & b_1 \\ a_{2,1} & x_1 & + & a_{2,2} & x_2 & + & \cdots & + & a_{2,n} & x_n & = & b_2 \\ \vdots & & & & & & & & & & & \vdots \\ a_{m,1} & x_1 & + & a_{m,2} & x_2 & + & \cdots & + & a_{m,n} & x_n & = & b_m \end{array}$$

ist ein

System von m linearen Gleichungen in den n Unbekannten x_1, x_2, \dots, x_n .

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

mit

$$A = [a_{i,j}]_{1 \leq i \leq m, 1 \leq j \leq n} \in \mathbb{C}^{m \times n},$$

$$\mathbf{x} = [x_1, \dots, x_n]^\top \in \mathbb{C}^n,$$

$$\mathbf{b} = [b_1, \dots, b_m]^\top \in \mathbb{C}^m.$$

A heißt **Koeffizientenmatrix** und \mathbf{b} **rechte Seite** des linearen Gleichungssystems.

Geometrische Deutung: Jede der m Gleichungen repräsentiert eine Hyperebene im \mathbb{C}^n :

$$A(i, :)\mathbf{x} = b_i \text{ mit } A(i, :) = [a_{i,1}, a_{i,2}, \dots, a_{i,n}] \quad i = 1, 2, \dots, m.$$

Gesucht ist der **Durchschnitt dieser Hyperebenen**.

Analytische Deutung: Mit $A(:, j) = [a_{1,j}, a_{2,j}, \dots, a_{m,j}]^\top$ lässt sich $A\mathbf{x} = \mathbf{b}$ auch als

$$x_1 A(:, 1) + x_2 A(:, 2) + \dots + x_n A(:, n) = \mathbf{b}$$

schreiben. Gesucht sind also Koeffizienten x_j , mit deren Hilfe man die rechte Seite \mathbf{b} als **Linearkombination** der Spalten von A darstellen kann.

Vorbemerkungen

Lineare Gleichungssysteme: Existenz und Eindeutigkeit der Lösung

Definiert man das **Bild** von A (engl. range) durch

$$\mathcal{R}(A) := \left\{ \mathbf{y} \in \mathbb{C}^m : \exists \mathbf{x} = [x_1, \dots, x_n]^\top \in \mathbb{C}^n \text{ mit } \mathbf{y} = A\mathbf{x} = \sum_{j=1}^n x_j A(:, j) \right\},$$

so besitzt $A\mathbf{x} = \mathbf{b}$ genau dann (mindestens) eine Lösung, wenn $\mathbf{b} \in \mathcal{R}(A)$ gilt.

Der **Kern** oder **Nullraum** von A ist durch

$$\mathcal{N}(A) := \{ \mathbf{z} \in \mathbb{C}^n : A\mathbf{z} = \mathbf{0} \}$$

definiert. Offensichtlich besitzt $A\mathbf{x} = \mathbf{b}$ höchstens eine Lösung, wenn $\mathcal{N}(A) = \{\mathbf{0}\}$ gilt.

Vorbemerkungen

Lineare Gleichungssysteme: Existenz und Eindeutigkeit der Lösung

Für quadratische Matrizen $A \in \mathbb{C}^{n \times n}$ (in diesem Kapitel werden wir uns ausschließlich mit solchen Matrizen befassen) heißt das:

$Ax = b$ ist genau dann eindeutig lösbar, wenn A invertierbar ist.

Satz 3.1

Für eine quadratische Matrix $A \in \mathbb{C}^{n \times n}$ sind die folgenden fünf Aussagen äquivalent:

- (a) A ist invertierbar.
- (b) $\mathcal{N}(A) = \{\mathbf{0}\}$.
- (c) $\mathcal{R}(A) = \mathbb{C}^n$.
- (d) Alle Eigenwerte von A sind ungleich 0.
- (e) $\det(A) \neq 0$.

- ① Einführung und Begriffe
- ② Gleitpunktarithmetik und Fehleranalyse
- ③ Direkte Verfahren zur Lösung linearer Gleichungssysteme
 - 3.1 Vorbemerkungen
 - 3.2 Störungstheorie
 - 3.3 Das Lösen von Dreieckssystemen
 - 3.4 Gauß-Elimination
 - 3.5 Pivotisierung
 - 3.6 Equilibrierung und Nachiteration
 - 3.7 Stabilität bei der Gauß-Elimination
- ④ Direkte Verfahren für spezielle Systeme

Wir betrachten zunächst Störungen der Einheitsmatrix:

Lemma 3.2

Für $A \in \mathbb{C}^{n \times n}$ gelte $\|A\| < 1$ für eine Matrixnorm $\|\cdot\|$ auf $\mathbb{C}^{n \times n}$. Dann gilt

- (a) $I - A$ ist invertierbar.
- (b) Die Inverse besitzt die Darstellung

$$(I - A)^{-1} = \sum_{j=0}^{\infty} A^j \quad (\text{Neumannsche Reihe}).$$

- (c) Für die Inverse gilt die Abschätzung $\|(I - A)^{-1}\| \leq \frac{1}{1 - \|A\|}$.

Der folgende Satz gibt ein Maß für den Abstand einer invertierbaren Matrix zur nächstgelegenen singulären Matrix an.

Lemma 3.3

Zu gegebenen Vektoren $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$ mit $\|\mathbf{x}\|_p = \|\mathbf{y}\|_p = 1$, $1 \leq p \leq \infty$, existiert stets eine Matrix $B \in \mathbb{C}^{n \times n}$ mit

$$\|B\|_p = 1 \quad \text{und} \quad B\mathbf{x} = \mathbf{y}.$$

Satz 3.4

Es sei $A \in \mathbb{C}^{n \times n}$ invertierbar. Dann gilt

$$\begin{aligned} \min \left\{ \frac{\|\Delta A\|_p}{\|A\|_p} : \Delta A \in \mathbb{C}^{n \times n} \text{ so, dass } A + \Delta A \text{ singular} \right\} \\ = \frac{1}{\|A\|_p \|A^{-1}\|_p} \quad (1 \leq p \leq \infty). \end{aligned}$$

Es sei $\|\cdot\|$ eine Matrixnorm in $\mathbb{C}^{n \times n}$. Ist $A \in \mathbb{C}^{n \times n}$ invertierbar, so heißt

$$\text{cond}(A) = \text{cond}_{\|\cdot\|}(A) := \|A\| \|A^{-1}\| \quad (\geq 1)$$

die **Konditionszahl** von A . Für singuläre Matrizen A setzen wir $\text{cond}(A) = \infty$. (Kurzschreibweise: $\text{cond}_p(A)$, falls $\|\cdot\| = \|\cdot\|_p$.)

Satz 3.4 besagt, dass der (relative) Abstand (gemessen in $\|\cdot\|_p$) einer regulären Matrix A zur Menge der singulären Matrizen gerade $1/\text{cond}_p(A)$ beträgt.

Man kann $\text{cond}(A)$ als (normalisierte) Fréchet-Ableitung der Abbildung $A \mapsto A^{-1}$ interpretieren:

$$\text{cond}(A) = \lim_{h \rightarrow 0} \sup_{\|\Delta A\| \leq h \|A\|} \frac{\|(A + \Delta A)^{-1} - A^{-1}\|}{h} \frac{1}{\|A^{-1}\|}.$$

Satz 3.5

Mit $\|\cdot\|$ bezeichnen wir eine Vektornorm in \mathbb{C}^n sowie eine passende Matrixnorm in $\mathbb{C}^{n \times n}$. Es seien $A, \Delta A \in \mathbb{C}^{n \times n}$ sowie $\mathbf{b}, \Delta \mathbf{b} \in \mathbb{C}^n$. A sei invertierbar und für die Störungsmatrix ΔA gelte $\|\Delta A\| < 1/\|A^{-1}\|$. Schließlich seien $\mathbf{x} = A^{-1}\mathbf{b}$ die Lösung des ungestörten LGS $A\mathbf{y} = \mathbf{b}$ und $\mathbf{x} + \Delta \mathbf{x} = (A + \Delta A)^{-1}(\mathbf{b} + \Delta \mathbf{b})$ die Lösung des gestörten LGS $(A + \Delta A)\mathbf{y} = \mathbf{b} + \Delta \mathbf{b}$ (beachten Sie, dass $A + \Delta A$ invertierbar ist).

Dann folgt

$$\frac{\|\Delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\text{cond}(A)}{1 - \text{cond}(A) \frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta \mathbf{b}\|}{\|\mathbf{b}\|} \right). \quad (3.1)$$

Interpretation: Relative Fehler in den Daten A und \mathbf{b} verstärken sich mit dem Faktor $\text{cond}(A)$ ins Ergebnis $\mathbf{x} = A^{-1}\mathbf{b}$. Auf einem Rechner mit Rundungseinheit u muss man also mindestens mit einem Fehler der Größenordnung $\text{cond}(A) \cdot u$ in \mathbf{x} rechnen – ein LGS $A\mathbf{x} = \mathbf{b}$ ist durch *kein* Verfahren zuverlässig lösbar, wenn $\text{cond}(A) \geq 1/u$ gilt.

Weitere Eigenschaften der Konditionszahl:

- (1) Sie ist normabhängig, z.B. $\text{cond}_p(I_n) = \|I_n\|_p \|I_n\|_p = 1$ für $p \geq 1$, aber $\text{cond}_F(I_n) = \|I_n\|_F \|I_n\|_F = n$. Aber sind $\|\cdot\|_\alpha$ und $\|\cdot\|_\beta$ Matrixnormen in $\mathbb{C}^{n \times n}$, so gibt es stets Konstanten $c, C > 0$ mit $c \text{cond}_\alpha(A) \leq \text{cond}_\beta(A) \leq C \text{cond}_\alpha(A)$ für alle A .

Z.B.
$$\frac{1}{n} \text{cond}_2(A) \leq \text{cond}_1(A) \leq n \text{cond}_2(A),$$

oder
$$\frac{1}{n} \text{cond}_\infty(A) \leq \text{cond}_2(A) \leq n \text{cond}_\infty(A).$$

- (2) Immer gilt $\text{cond}(A) \geq 1$.
(3) Ist A symmetrisch und positiv definit, so ist

$$\text{cond}_2(A) = \lambda_{\max}(A) / \lambda_{\min}(A)$$

- (4) Ist $U \in \mathbb{C}^{n \times n}$ unitär, so ist $\text{cond}_2(U) = 1$ und

$$\text{cond}_2(AU) = \text{cond}_2(UA) = \text{cond}_2(A) \quad \forall A \in \mathbb{C}^{n \times n}.$$

Ist $\tilde{\mathbf{x}}$ eine **Näherungslösung** des linearen Gleichungssystems $A\mathbf{x} = \mathbf{b}$, so ist der **relative Rückwärtsfehler** (gemessen in einer Vektor- bzw. Matrixnorm $\|\cdot\|$ auf \mathbb{C}^n bzw. $\mathbb{C}^{n \times n}$, „normwise backward error“) definiert durch

$$\eta(\tilde{\mathbf{x}}) = \min \left\{ \varepsilon : (A + \Delta A)\tilde{\mathbf{x}} = \mathbf{b} + \Delta \mathbf{b}, \frac{\|\Delta A\|}{\|A\|} \leq \varepsilon, \frac{\|\Delta \mathbf{b}\|}{\|\mathbf{b}\|} \leq \varepsilon \right\}.$$

Satz 3.6 (Rigal und Gaches, 1967)

Es bezeichne $\|\cdot\|$ eine Vektornorm auf \mathbb{C}^n sowie die dadurch induzierte Matrixnorm auf $\mathbb{C}^{n \times n}$. Der Norm-Rückwärtsfehler einer Näherungslösung $\tilde{\mathbf{x}} \in \mathbb{C}^n$ zum linearen Gleichungssystem $A\mathbf{x} = \mathbf{b}$ mit $A \in \mathbb{C}^{n \times n}$ invertierbar, $\mathbf{b} \in \mathbb{C}^n$, ist gegeben durch

$$\eta(\tilde{\mathbf{x}}) = \frac{\|\mathbf{r}\|}{\|A\| \|\tilde{\mathbf{x}}\| + \|\mathbf{b}\|}, \quad (3.2)$$

wobei der Vektor $\mathbf{r} := \mathbf{b} - A\tilde{\mathbf{x}}$ das **Residuum** von $\tilde{\mathbf{x}}$ bezüglich des linearen Gleichungssystems bezeichnet.

Normabschätzungen sind nicht immer hinreichend aussagekräftig. In solchen Fällen ist oft eine komponentenweise Analyse notwendig. Der **Komponenten-Rückwärtsfehler** („componentwise backward error“) ist definiert durch

$$\omega_{E,\mathbf{f}}(\tilde{\mathbf{x}}) := \min \{ \varepsilon : (A + \Delta A)\tilde{\mathbf{x}} = \mathbf{b} + \Delta \mathbf{b}, |\Delta A| \leq \varepsilon E, |\Delta \mathbf{b}| \leq \varepsilon \mathbf{f}, \} .$$

(Die Ungleichungen sind komponentenweise zu lesen.) Hierbei besitzen die Matrix E und der Vektor \mathbf{f} nichtnegative Einträge und stellen Fehlertoleranzen für jede einzelne Komponente dar. Spezielle Wahlen:

- $E = |A|$, $\mathbf{f} = |\mathbf{b}|$: liefert den relativen Komponenten-Rückwärtsfehler.
- $E = |A|e e^\top$, $e = [1, \dots, 1]^\top$, $\mathbf{f} = |\mathbf{b}|$: Zeilen-Rückwärtsfehler
- $E = e e^\top |A|$, $\mathbf{f} = \|\mathbf{b}\|_\infty e$: Spalten-Rückwärtsfehler
- $E = \|A\| e e^\top$, $\mathbf{f} = \|\mathbf{b}\| e$: liefert bis auf Konstante Norm-Rückwärtsfehler.

Satz 3.7 (Oettli und Prager, 1964)

Der Komponenten-Rückwärtsfehler besitzt die Darstellung

$$\omega_{E,f}(\tilde{\mathbf{x}}) = \max_{i=1}^n \frac{|r_i|}{(E|\tilde{\mathbf{x}}| + \mathbf{f})_i},$$

wobei $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$ und $\xi/0$ gleich Null zu setzen ist, falls, $\xi = 0$ und andernfalls $\xi/0 = \infty$.

- ① Einführung und Begriffe
- ② Gleitpunktarithmetik und Fehleranalyse
- ③ Direkte Verfahren zur Lösung linearer Gleichungssysteme
 - 3.1 Vorbemerkungen
 - 3.2 Störungstheorie
 - 3.3 Das Lösen von Dreieckssystemen
 - 3.4 Gauß-Elimination
 - 3.5 Pivotisierung
 - 3.6 Equilibrierung und Nachiteration
 - 3.7 Stabilität bei der Gauß-Elimination
- ④ Direkte Verfahren für spezielle Systeme

Eine Matrix der Form

$$L = \begin{bmatrix} \ell_{1,1} & & & & \\ \ell_{2,1} & \ell_{2,2} & & & \\ \vdots & & \ddots & & \\ \ell_{n,1} & \ell_{n,2} & \cdots & \ell_{n,n} & \end{bmatrix} \in \mathbb{C}^{n \times n} \quad (\ell_{i,j} = 0 \text{ für } i < j)$$

heißt **untere Dreiecksmatrix**.

- L ist genau dann invertierbar, wenn $\det(L) = \prod_{j=1}^n \ell_{j,j} \neq 0$.
- Die invertierbaren unteren Dreiecksmatrizen bilden bez. der Matrizenmultiplikation eine Gruppe.
- L heißt **normierte** untere Δ -Matrix, wenn $\ell_{i,i} = 1$ ($i = 1, 2, \dots, n$) ist. Die normierten unteren Δ -Matrizen bilden bez. der Matrizenmultiplikation ebenfalls eine Gruppe.
- Analoge Aussagen gelten für **obere** Δ -Matrizen $R = [r_{i,j}] \in \mathbb{C}^{n \times n}$ ($r_{i,j} = 0$ für $i > j$).

Das Lösen von Dreieckssystemen

Substitutions-Algorithmus, Komplexität

Ist $\det(L) \neq 0$, so besitzt das untere Δ -System $Lx = c$ genau eine Lösung, die man durch **Vorwärts-Substitution**,

$$x_j = \frac{1}{\ell_{j,j}}(c_j - \ell_{j,1}x_1 - \cdots - \ell_{j,j-1}x_{j-1}) \quad (j = 1, 2, \dots, n),$$

mit n Divisionen, $n(n-1)/2$ Multiplikationen und $n(n-1)/2$ Additionen, also mit insgesamt n^2 Gleitpunktoperationen berechnen kann.

Obere Δ -Systeme $Rx = d$ werden – falls $\det(R) \neq 0$ – analog durch **Rückwärts-Substitution**,

$$x_j = \frac{1}{r_{j,j}}(d_j - r_{j,j+1}x_{j+1} - \cdots - r_{j,n}x_n) \quad (j = n, n-1, \dots, 1),$$

in $O(n^2)$ Gleitpunktoperationen gelöst.

Wir wollen als nächstes zeigen, dass Vorwärts- und Rückwärtssubstitution rückwärtsstabile Algorithmen zur Lösung von Dreieckssystemen sind. Hierzu sind zwei Lemmata hilfreich:

Lemma 3.8

Sind $|\delta_j| \leq u$ und $\rho_j = \pm 1$ für $j = 1, \dots, n$ und gilt $nu < 1$, so ist

$$\prod_{j=1}^n (1 + \delta_j)^{\rho_j} = 1 + \vartheta_n, \quad \text{mit} \quad |\vartheta_n| \leq \frac{nu}{1 - nu}.$$

Lemma 3.9

Der Ausdruck $y := \left(c - \sum_{i=1}^{k-1} a_i b_i\right) / b_k$ werde gemäß folgendem Algorithmus in Gleitpunktarithmetik mit Rundungseinheit u ausgewertet:

$$s := c; \quad \text{for } i = 1 : k - 1, s := s - a_i b_i; \text{ end} \quad s := s / b_k;$$

Dann besitzt der berechnete Wert $\tilde{y} := \text{fl}(y)$ die Darstellung

$$b_k \tilde{y} (1 + \vartheta_k) = c - \sum_{i=1}^{k-1} a_i b_i (1 + \vartheta_i), \quad |\vartheta_i| \leq \frac{iu}{1 - iu}.$$

Satz 3.10

Die untere Δ -Matrix $L = [\ell_{i,j}] \in \mathbb{C}^{n \times n}$ sei invertierbar. Die Lösung von $Lx = c$ werde in Gleitpunktarithmetik mit Rundungseinheit u durch Vorwärtssubstitution bestimmt. Dann erfüllt die berechnete Lösung \tilde{x} das LGS

$$(L + \Delta L)\tilde{x} = c$$

mit einer unteren Dreiecksmatrix $\Delta L = [\lambda_{i,j}] \in \mathbb{C}^{n \times n}$, wobei

$$|\lambda_{i,j}| \leq nu |\ell_{i,j}| + O(u^2) \quad (1 \leq i, j \leq n)$$

gilt.

- ① Einführung und Begriffe
- ② Gleitpunktarithmetik und Fehleranalyse
- ③ Direkte Verfahren zur Lösung linearer Gleichungssysteme
 - 3.1 Vorbemerkungen
 - 3.2 Störungstheorie
 - 3.3 Das Lösen von Dreieckssystemen
 - 3.4 Gauß-Elimination
 - 3.5 Pivotisierung
 - 3.6 Equilibrierung und Nachiteration
 - 3.7 Stabilität bei der Gauß-Elimination
- ④ Direkte Verfahren für spezielle Systeme

Gauß-Elimination

Geschichtliches

- 220 v. Chr. – 9 n. Chr. *Chiu Chang Suan Shu* (Neun Bücher arithmetischer Technik). China. Buch 8 enthält eine Anleitung, LGS bis zur Dimension 5 mittels Elimination zu lösen.
1750. Gabriel Cramer: *Cramersche Regel*. Theoretisch einwandfrei aber praktisch unbrauchbar. (Lösung eines 10×10 Problems erfordert ca. 300 Mio Multiplikationen.)
- 1809,1823. Carl Friedrich Gauß: *Theoria motus corporum ...* und *Theoria combinationis observationum ...*. Beschreibt Eliminationsverfahren für symmetrische Matrizen aus der Ausgleichsrechnung.
1890. Wilhelm Jordan: *Handbuch der Vermessungskunde*. Erste schriftliche Erwähnung des Gauß-Jordan Algorithmus.
1948. Alan Turing: Darstellung von GE als Folge von Multiplikationen mit unteren Dreiecksmatrizen.
1961. James Wilkinson: Rundungsfehleranalyse von GE.
- Stand der Kunst.** Zuverlässige Lösung von sehr großen Systemen möglich ($N \approx 10^5$ für vollbesetzte, $N \approx 10^7$ für dünn besetzte Matrizen). Software von hoher Qualität verfügbar (e.g. LAPACK).

Idee:

- Transformiere

$$Ax = b, \quad A \in \mathbb{C}^{n \times n} \text{ mit } \det(A) \neq 0,$$

auf ein oberes Dreieckssystem $Rx = d$, ohne die Lösung zu verändern.

- Löse dann $Rx = d$ durch Rückwärts-Substitution.
- Die Lösung eines Gleichungssystems verändert sich nicht, wenn man von der j -ten Gleichung das λ -fache der i -ten Gleichung subtrahiert ($i < j$).
- Formal: Statt $Ax = b$ betrachte $(LA)x = Lb$ mit

$$L = I - m_j u_i^\top \in \mathbb{R}^{n \times n}.$$

Dabei ist $m_j = \lambda u_j \in \mathbb{C}^n$; u_i und u_j bezeichnen den i -ten bzw. j -ten Einheitsvektor im \mathbb{R}^n .

Beispiel.

$$\begin{bmatrix} 2 & -1 & -3 & 3 \\ 4 & 0 & -3 & 1 \\ 6 & 1 & -1 & 6 \\ -2 & -5 & 4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ -8 \\ -16 \\ -12 \end{bmatrix}.$$

1. Schritt: Eliminiere x_1 aus der zweiten, dritten und vierten Gleichung. Um dies zu erreichen, subtrahiert man das $\ell_{i,1} = \frac{a_{i,1}}{a_{1,1}}$ -fache der ersten Gleichung von der i -ten ($i = 2, 3, 4$). Hier: $\ell_{2,1} = 2$, $\ell_{3,1} = 3$, $\ell_{4,1} = -1$.

Mit

$$L_1 = I - \begin{bmatrix} 0 \\ \ell_{2,1} \\ \ell_{3,1} \\ \ell_{4,1} \end{bmatrix} [1 \ 0 \ 0 \ 0] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -3 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

ergibt sich

Gauß-Elimination

Der Algorithmus

$$L_1[A|\mathbf{b}] = \left[\begin{array}{cccc|c} 2 & -1 & -3 & 3 & 1 \\ 0 & 2 & 3 & -5 & -10 \\ 0 & 4 & 8 & -3 & -19 \\ 0 & -6 & 1 & 4 & -11 \end{array} \right].$$

2. Schritt: Eliminiere x_2 aus der dritten und vierten Gleichung, d.h. subtrahiere das $\ell_{i,2} = \frac{a_{i,2}}{a_{2,2}}$ -fache der zweiten Gleichung von der i -ten ($i = 3, 4$). Hier: $\ell_{3,2} = 2, \ell_{4,2} = -3$. Mit

$$L_2 = I - \begin{bmatrix} 0 \\ 0 \\ \ell_{3,2} \\ \ell_{4,2} \end{bmatrix} [0 \ 1 \ 0 \ 0] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -2 & 1 & 0 \\ 0 & 3 & 0 & 1 \end{bmatrix}$$

ergibt sich

$$L_2 L_1 [A | \mathbf{b}] = \left[\begin{array}{cccc|c} 2 & -1 & -3 & 3 & 1 \\ 0 & 2 & 3 & -5 & -10 \\ 0 & 0 & 2 & 7 & 1 \\ 0 & 0 & 10 & -11 & -41 \end{array} \right].$$

3. Schritt: Eliminiere x_3 aus der vierten Gleichung, d.h. subtrahiere das $\ell_{i,3} = \frac{a_{i,3}}{a_{3,3}}$ -fache der dritten Gleichung von der i -ten ($i = 4$). Hier $\ell_{4,3} = 5$. Mit

$$L_3 = I - \begin{bmatrix} 0 \\ 0 \\ 0 \\ \ell_{4,3} \end{bmatrix} [0 \ 0 \ 1 \ 0] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -5 & 1 \end{bmatrix}$$

ergibt sich

Gauß-Elimination

Der Algorithmus

$$\left[\begin{array}{cccc|c} 2 & -1 & -3 & 3 & 1 \\ 0 & 2 & 3 & -5 & -10 \\ 0 & 0 & 2 & 7 & 1 \\ 0 & 0 & 0 & -46 & -46 \end{array} \right].$$

Die Eliminationsphase ist damit abgeschlossen und wir können x_4, x_3, x_2, x_1 durch Rückwärts-Substitution bestimmen:

$$x_4 = 1, \quad x_3 = \frac{1}{2}[1 - 7x_4] = -3, \quad x_2 = \frac{1}{2}[-10 - 3x_3 + 5x_4] = 2, \\ x_1 = \frac{1}{2}[1 + x_2 + 3x_3 - 3x_4] = -\frac{9}{2}.$$

Das Gaußsche Eliminationsverfahren liefert eine **LR-Zerlegung** von A , d.h. eine Faktorisierung $A = L \cdot R$ mit einer normierten unteren Δ -Matrix L und einer oberen Δ -Matrix R .

Gauß-Elimination

Der Algorithmus

Im allgemeinen: $L_{n-1} \cdots L_2 L_1 A = R$, d.h.

$$A = (L_{n-1} \cdots L_2 L_1)^{-1} R = (L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1}) R =: L \cdot R.$$

In unserem Beispiel:

$$\begin{aligned} L &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -3 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -2 & 1 & 0 \\ 0 & 3 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -5 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 5 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ -1 & -3 & 5 & 1 \end{bmatrix} \quad \text{also} \end{aligned}$$

Gauß-Elimination

Der Algorithmus

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ -1 & -3 & 5 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & -3 & 3 \\ 0 & 2 & 3 & -5 \\ 0 & 0 & 2 & 7 \\ 0 & 0 & 0 & -46 \end{bmatrix}.$$

Lemma 3.11

Mit \mathbf{u}_i bezeichnen wir den i -ten Einheitsvektor im \mathbb{R}^n . Sei

$$\mathbf{m}_i = [0, \dots, 0, \ell_{i+1,i}, \dots, \ell_{n,i}]^\top \in \mathbb{C}^n, \quad (1 \leq i \leq n-1).$$

Für die Matrizen

$$L_i = I - \mathbf{m}_i \mathbf{u}_i^\top \in \mathbb{C}^{n \times n}$$

gelten:

(a) $L_i^{-1} = I + \mathbf{m}_i \mathbf{u}_i^\top.$

(b) $L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1} = I + \mathbf{m}_1 \mathbf{u}_1^\top + \mathbf{m}_2 \mathbf{u}_2^\top + \cdots + \mathbf{m}_{n-1} \mathbf{u}_{n-1}^\top.$

Gauß-Elimination

Der Algorithmus

Bei der numerischen Rechnung wird die Matrix A durch die „Zwischenmatrizen“ überschrieben (analog für die rechte Seite). Außerdem müssen die Nullen unterhalb der Hauptdiagonalen nicht gespeichert werden, man verwendet diese Felder für die Zahlen $\ell_{i,j}$.

Pseudocode:

```
for  $j = 1 : n - 1$  do  
  if  $a_{j,j} = 0$  then  
    stop  
  else  
    for  $i = j + 1 : n$  do  
       $a_{i,j} := a_{i,j} / a_{j,j}$   
       $b_i := b_i - a_{i,j} b_j$   
      for  $k = j + 1 : n$  do  
         $a_{i,k} := a_{i,k} - a_{i,j} a_{k,j}$   
      end for  
    end for  
  end if  
end for
```

Aufwand: $\frac{2}{3}n^3 + O(n^2)$ flops.

In unserem Beispiel:

$$\left[\begin{array}{cccc|c} 2 & -1 & -3 & 3 & 1 \\ 4 & 0 & -3 & 1 & 8 \\ 6 & 1 & -1 & 6 & -16 \\ 2 & 5 & 4 & 1 & -12 \end{array} \right] \xrightarrow{1.} \left[\begin{array}{cccc|c} 2 & -1 & -3 & 3 & 1 \\ 2 & 2 & 3 & -5 & -10 \\ 3 & 4 & 8 & -3 & -19 \\ -1 & -6 & 1 & 4 & -11 \end{array} \right]$$

$$\xrightarrow{2.} \left[\begin{array}{cccc|c} 2 & -1 & -3 & 3 & 1 \\ 2 & 2 & 3 & -5 & -10 \\ 3 & 2 & 2 & 7 & 1 \\ -1 & -3 & 10 & -11 & 41 \end{array} \right] \xrightarrow{3.} \left[\begin{array}{cccc|c} 2 & -1 & -3 & 3 & 1 \\ 2 & 2 & 3 & -5 & -10 \\ 3 & 2 & 2 & 7 & 1 \\ -1 & -3 & 5 & -46 & -46 \end{array} \right]$$

Gauß-Elimination

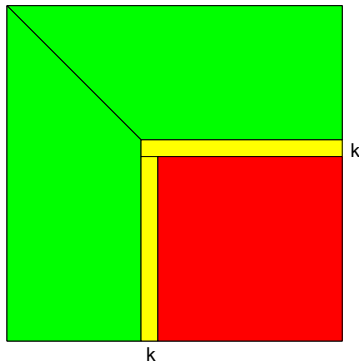
Algorithmische Varianten von Gauß-Elimination

kij-Variante: (Lehrbuchvariante)

```
for  $k = 1 : n - 1$  do
  for  $i = k + 1 : n$  do
     $\ell_{i,k} = a_{i,k}/a_{k,k}$ 
    for  $j = k + 1 : n$  do
       $a_{i,j} = a_{i,j} - \ell_{i,k}a_{k,j}$ 
    end for
  end for
end for
```

kji-Variante: (spaltenorientiert)

```
for  $k = 1 : n - 1$  do
  for  $s = k + 1 : n$  do
     $\ell_{s,k} = a_{s,k}/a_{k,k}$ 
  end for
  for  $j = k + 1 : n$  do
    for  $i = k + 1 : n$  do
       $a_{i,j} = a_{i,j} - \ell_{i,k}a_{k,j}$ 
    end for
  end for
end for
```



Zugriffsmuster *kij* / *kji* Varianten.

Gauß-Elimination

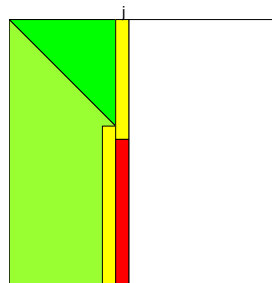
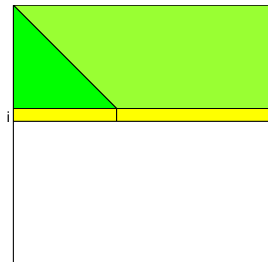
Algorithmische Varianten von Gauß-Elimination

ikj-Variante: (zeilenorientiert)

```
for  $i = 2 : n$  do
  for  $k = 1 : i - 1$  do
     $\ell_{i,k} = a_{i,k}/a_{k,k}$ 
    for  $j = k + 1 : n$  do
       $a_{i,j} = a_{i,j} - \ell_{i,k}a_{k,j}$ 
    end for
  end for
end for
```

jki-Variante: (spaltenorientiert)

```
for  $j = 2 : n$  do
  for  $s = j : n$  do
     $\ell_{s,j-1} = a_{s,j-1}/a_{j-1,j-1}$ 
  end for
  for  $k = 1 : j - 1$  do
    for  $i = k + 1 : n$  do
       $a_{i,j} = a_{i,j} - \ell_{i,k}a_{k,j}$ 
    end for
  end for
end for
```

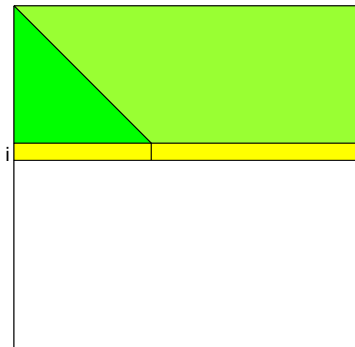


Gauß-Elimination

Algorithmische Varianten von Gauß-Elimination

ijk-Variante: (zeilenorientiert)

```
for  $i = 2 : n$  do  
  for  $j = 2 : i$  do  
     $\ell_{i,j-1} = a_{i,j-1}/a_{j-1,j-1}$   
    for  $k = 1 : j - 1$  do  
       $a_{i,j} = a_{i,j} - \ell_{i,k}a_{k,j}$   
    end for  
  end for  
  for  $j = i + 1 : n$  do  
    for  $k = 1 : i - 1$  do  
       $a_{i,j} = a_{i,j} - \ell_{i,k}a_{k,j}$   
    end for  
  end for  
end for
```

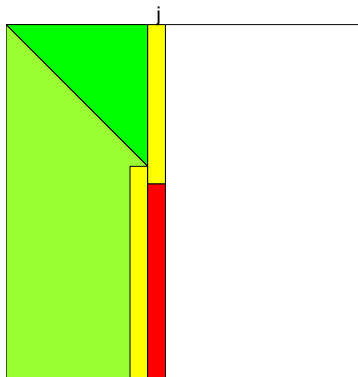


Gauß-Elimination

Algorithmische Varianten von Gauß-Elimination

jik-Variante: (spaltenorientiert)

```
for  $j = 2 : n$  do  
  for  $s = j : n$  do  
     $\ell_{s,j-1} = a_{s,j-1} / a_{j-1,j-1}$   
  end for  
  for  $i = 2 : j$  do  
    for  $k = 1 : i - 1$  do  
       $a_{i,j} = a_{i,j} - \ell_{i,k} a_{k,j}$   
    end for  
  end for  
  for  $i = j + 1 : n$  do  
    for  $k = 1 : j - 1$  do  
       $a_{i,j} = a_{i,j} - \ell_{i,k} a_{k,j}$   
    end for  
  end for  
end for
```

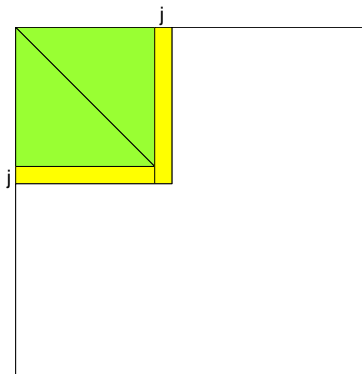


Gauß-Elimination

Algorithmische Varianten von Gauß-Elimination

Bordering-Variante: (jki)

```
for  $j = 2 : n$  do
  for  $k = 1 : j - 2$  do
    for  $i = k + 1 : j - 1$  do
       $a_{i,j} = a_{i,j} - \ell_{i,k} a_{k,j}$ 
    end for
  end for
  for  $k = 1 : j - 1$  do
     $\ell_{j,k} = a_{j,k} / a_{k,k}$ 
    for  $i = 1 : k + 1 : j$  do
       $a_{j,i} = a_{j,i} - \ell_{j,k} a_{k,i}$ 
    end for
  end for
end for
```



Gauß-Elimination

Algorithmische Varianten von Gauß-Elimination

Verfahren von Crout/Doolittle:

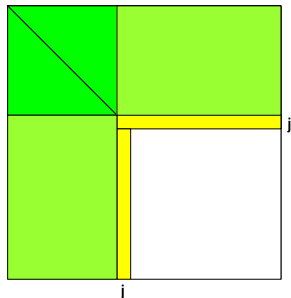
for $j = 1 : n$ **do**

$$a_{j:n,j} = a_{j:n,j} - \ell_{j:n,1:j-1} a_{1:j-1,j}$$

$$a_{j+1:n,j} = a_{j+1:n,j} / a_{j,j}$$

$$a_{j,j+1:n} = a_{j,j+1:n} - \ell_{j,1:j-1} a_{1:j-1,j+1:n}$$

end for



Hier wird das Aufdatieren der Restmatrix (Schur-Komplement) so spät wie möglich ausgeführt.

Um auf Rechnern mit ausgeprägter Speicherhierarchie effiziente Implementierungen von Algorithmen der numerischen linearen Algebra zu erzielen benutzen moderne Programmpakete als Bausteine die **Basic Linear Algebra Subroutines (BLAS)**. Diese werden von Rechnerherstellern in für ihre Systeme optimierter Form zur Verfügung gestellt.

Man unterscheidet drei „Levels“ unterschiedlicher Effizienz; hier einige typische Vertreter

Operation (Level)	Definition	n_A	n_S	$q = n_A/n_S$
saxpy (1)	$\mathbf{y} := \alpha \mathbf{x} + \mathbf{y}$	$2n$	$3n + 1$	$2/3$
Matrix-Vektor Produkt (2)	$\mathbf{y} := \mathbf{A}\mathbf{x} + \mathbf{y}$	$2n^2$	$n^2 + 3n$	2
Matrix-Matrix Produkt (3)	$\mathbf{Y} := \mathbf{Y} + \mathbf{A}\mathbf{X}$	$2n^3$	$4n^2$	$n/2$

in der Tabelle: $\alpha \in \mathbb{R}$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\mathbf{A}, \mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times n}$

n_A : Anzahl arithmetischer Operationen

n_S : Anzahl Speicherzugriffe

Da auf aktuellen (und derzeit absehbaren) Rechnerarchitekturen Speicherzugriffe deutlich länger dauern als Arithmetik, verspricht ein großer Quotient q eine schnellere Ausführung:

Bezeichnen t_A die Dauer einer (typischen) Gleitpunktoperation,
 t_S die Dauer einer (typischen) Speicherzugriffs, sowie
 t_G die Ausführungsdauer des Algorithmus,

so gilt

$$t_G = t_A \cdot n_A + t_S \cdot n_S = t_A n_A \left(1 + \frac{t_S n_S}{t_A n_A} \right) = t_A n_A \left(1 + \frac{1}{q} \frac{t_S}{t_A} \right)$$

Es ist somit effizient, Algorithmen der numerischen linearen Algebra möglichst aus BLAS2 bzw. BLAS3 Routinen aufzubauen.

Wir zeigen eine Möglichkeit, dies bei Gauß-Elimination (GE) durch **Blockeinteilung** zu erreichen und betrachten hierzu eine Abwandlung von GE für $m \times n$ -Matrizen, $m \geq n$.

Algorithmus 3.1:

for $k = 1 : \min\{m - 1, n\}$ **do**

$$\ell_{k+1:m,k} = a_{k+1:m,k} / a_{k,k}$$

if $k < n$ **then**

$$a_{k+1:m,k+1:n} = a_{k+1:m,k+1:n} - \ell_{k+1:m,k} a_{k,k+1:n}$$

end if

end for

Wir partitionieren die Matrizen nach $k - 1$ Schritten von GE:

$$\begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,1} & A_{3,2} & A_{3,3} \end{bmatrix} = \begin{bmatrix} L_{1,1} & O & O \\ L_{2,1} & I & O \\ L_{3,1} & O & I \end{bmatrix} \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ O & \tilde{A}_{2,2} & \tilde{A}_{2,3} \\ O & \tilde{A}_{3,2} & \tilde{A}_{3,3} \end{bmatrix}$$

mit quadratischen Matrizen $A_{1,1}$, $A_{2,2}$, $A_{3,3}$ der Dimension $k - 1$, n_B bzw. $n - (k - 1) - n_B$ (n_B eine Blockgröße zwischen 1 und n).

Folgende Variante der GE ist reicher an BLAS3-Operationen (gezeigt ist ein Blockschritt):

1. Verwende Algorithmus 3.1 zur Faktorisierung von

$$\begin{bmatrix} \tilde{A}_{2,2} \\ \tilde{A}_{3,2} \end{bmatrix} = \begin{bmatrix} L_{2,2} \\ L_{3,2} \end{bmatrix} R_{2,2} .$$

2. Bestimme $R_{2,3} := L_{2,2}^{-1} \tilde{A}_{2,3}$ als Lösung eines unteren Dreieckssystems mit n_B rechten Seiten (BLAS3).
3. Setze $\tilde{\tilde{A}}_{3,3} := \tilde{A}_{3,3} - L_{3,2} R_{2,3}$ (BLAS3).

- ① Einführung und Begriffe
- ② Gleitpunktarithmetik und Fehleranalyse
- ③ Direkte Verfahren zur Lösung linearer Gleichungssysteme
 - 3.1 Vorbemerkungen
 - 3.2 Störungstheorie
 - 3.3 Das Lösen von Dreieckssystemen
 - 3.4 Gauß-Elimination
 - 3.5 Pivotisierung
 - 3.6 Equilibrierung und Nachiteration
 - 3.7 Stabilität bei der Gauß-Elimination
- ④ Direkte Verfahren für spezielle Systeme

Das Eliminationsverfahren „bricht zusammen“ (bereits im ersten Schritt), wenn es auf die Matrix

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

angewandt wird (obwohl $\det(A) \neq 0$). A besitzt keine LR-Zerlegung.

Satz 3.12

Es sei $A \in \mathbb{C}^{n \times n}$ regulär. Dann sind die folgenden Aussagen äquivalent:

- (a) A besitzt eine LR-Zerlegung mit einer normierten unteren Dreiecksmatrix L und einer regulären oberen Dreiecksmatrix R .
- (b) Alle führenden Hauptuntermatrizen von A sind regulär.

Bemerkung. Besitzt die reguläre Matrix $A \in \mathbb{C}^{n \times n}$ eine LR-Zerlegung, so ist diese eindeutig bestimmt.

Satz 3.13 (Satz von Gerschgorin)

Für $A = [a_{i,j}] \in \mathbb{C}^{n \times n}$ seien die **Gerschgorin-Kreisscheiben** definiert durch

$$D_i := \left\{ \zeta \in \mathbb{C} : |\zeta - a_{i,i}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{i,j}| \right\}, \quad i = 1, 2, \dots, n.$$

Bezeichnet $\Lambda(A) := \{\lambda \in \mathbb{C} : \det(A - \lambda I) = 0\}$ das **Spektrum** von A , so gilt

$$\Lambda(A) \subseteq \bigcup_{i=1}^n D_i.$$

Korollar 3.14

Ist $A = [a_{i,j}] \in \mathbb{C}^{n \times n}$ **streng diagonaldominant**, d.h. gilt

$$|a_{i,i}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{i,j}| \quad \text{für alle } i = 1, 2, \dots, n,$$

dann ist A regulär und besitzt eine LR-Zerlegung.

Ist π eine Permutation der Indexmenge $\{1, 2, \dots, n\}$, dann ist die zugehörige **Permutationsmatrix** $P = P_\pi = [p_{i,j}] \in \mathbb{R}^{n \times n}$ durch

$$p_{i,j} = \begin{cases} 1, & \text{falls } j = \pi(i), \\ 0, & \text{falls } j \neq \pi(i), \end{cases}$$

definiert.

- Es gilt $P_\pi^{-1} = P_{\pi^{-1}} = P_\pi^\top$.
- Die Menge aller Permutationsmatrizen bildet eine Gruppe bezüglich der Matrizenmultiplikation.
- Die Determinante einer Permutationsmatrix hat entweder den Wert 1 oder den Wert -1 .
- Ist $A = [a_{i,j}] \in \mathbb{C}^{n \times n}$ mit den Zeilen $\mathbf{z}_1^\top, \dots, \mathbf{z}_n^\top$ und Spalten $\mathbf{s}_1, \dots, \mathbf{s}_n$, so besitzt $P_\pi A$ die Zeilen $\mathbf{z}_{\pi(1)}^\top, \dots, \mathbf{z}_{\pi(n)}^\top$ und AP_π^\top die Spalten $\mathbf{s}_{\pi(1)}, \dots, \mathbf{s}_{\pi(n)}$. Insbesondere ist

$$P_\pi A P_\pi^\top = [a_{\pi(i), \pi(j)}].$$

Satz 3.15

Es sei $A \in \mathbb{C}^{n \times n}$ invertierbar. Dann gibt es eine Permutationsmatrix $P \in \mathbb{R}^{n \times n}$, so dass PA eine LR-Zerlegung besitzt:

$$PA = LR.$$

Dabei kann P so gewählt werden (Spaltenpivotsuche), dass die Einträge $\ell_{i,j}$ der Matrix L alle der Ungleichung $|\ell_{i,j}| \leq 1$ genügen.

Die Lösung eines LGS $Ax = b$ für beliebige reguläre Matrizen $A \in \mathbb{C}^{n \times n}$ mittels Gauß-Elimination besteht somit aus folgenden Schritten:

- (a) Bestimme P, L, R so, dass $PA = LR$.
- (b) Löse $Ly = Pb$ durch Vorwärtssubstitution.
- (c) Löse $Rx = y$ durch Rückwärtssubstitution.

(Insbesondere dann vorteilhaft, wenn $Ax = b_i$ mit einer Matrix A und mehreren rechten Seiten b_i zu lösen ist.)

Pseudocode: (Gauß-Elimination mit Spaltenpivotsuche)

```
for  $k = 1 : n - 1$  do  
  Bestimme  $p$  so, dass  $|a_{p,k}| = \max_{k \leq i \leq n} |a_{i,k}|$   
  Vertausche  $a_{k,:}$  mit  $a_{p,:}$   
  Vertausche  $b_k$  mit  $b_p$   
  if  $a_{k,k} = 0$  then  
    stop  
  else  
    for  $i = k + 1 : n$  do  
       $a_{i,k} := a_{i,k}/a_{k,k}$   
       $b_i := b_i - a_{i,k}b_k$   
      for  $j = k + 1 : n$  do  
         $a_{i,j} := a_{i,j} - a_{i,k}a_{k,j}$   
      end for  
    end for  
  end if  
end for
```

Spaltenpivotsuche muss (i.Allg.) immer durchgeführt werden, um **kleine Pivotelemente** zu vermeiden, die zu **ungenauen Ergebnissen** führen können:

Beispiel:

$$\begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \text{Lösung } \mathbf{x} = \frac{1}{1 - 10^{-20}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \approx \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

Gauß-Elimination (mit Rundungseinheit $u = 2^{-53} \approx 10^{-16}$):

$$10^{-20}x_1 + x_2 = 1$$

$$(1 - 10^{20})x_2 = -10^{20}, \text{ gerundet } -10^{20}\tilde{x}_2 = -10^{20}, \text{ d.h. } \tilde{x}_2 = 1, \tilde{x}_1 = 0.$$

(Gauß-Elimination mit Spaltenpivotsuche würde $\tilde{x}_1 = -1$ und $\tilde{x}_2 = 1$ liefern.)

Gauß-Elimination

Pivotsierung

Gauß-Elimination ohne Pivotsuche liefert also bei gerundeter Rechnung die Faktoren

$$\tilde{L} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \quad \text{und} \quad \tilde{R} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}$$

anstelle der exakten Faktoren

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \quad \text{und} \quad R = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}.$$

Beim Gleichungslösen wurde also verwendet

$$\tilde{L}\tilde{R} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix} \quad \text{statt} \quad LR = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix} = A.$$

- ① Einführung und Begriffe
- ② Gleitpunktarithmetik und Fehleranalyse
- ③ Direkte Verfahren zur Lösung linearer Gleichungssysteme
 - 3.1 Vorbemerkungen
 - 3.2 Störungstheorie
 - 3.3 Das Lösen von Dreieckssystemen
 - 3.4 Gauß-Elimination
 - 3.5 Pivotisierung
 - 3.6 **Equilibrierung und Nachiteration**
 - 3.7 Stabilität bei der Gauß-Elimination
- ④ Direkte Verfahren für spezielle Systeme

Die Spaltenpivotsuche kann leicht ad absurdum geführt werden, wenn man die verschiedenen Gleichungen unterschiedlich wichtet:

Beispiel (aus Abschnitt 3.5, nur erste Gleichung wurde mit 10^{20} multipliziert)

$$\begin{bmatrix} 1 & 10^{20} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 10^{20} \\ 0 \end{bmatrix} \text{ führt bei Rechnung (mit Spaltenpivotsuche),} \\ u \approx 10^{-16}, \text{ wie oben auf } \tilde{x}_2 = 1, \tilde{x}_1 = 0.$$

A heißt **zeilenequilibriert**, wenn in jeder Zeile die Betragssumme der Einträge gleich ist,

$$\sum_{j=1}^n |a_{i,j}| = 1, \quad i = 1, 2, \dots, n.$$

Man kann das leicht dadurch erreichen, indem man die i -te Zeile vorab mit dem Faktor $1/\sum_{j=1}^n |a_{i,j}|$ multipliziert.

Alternative: Bestimme Pivotelement (in Schritt j) so, dass

$$\frac{|a_{p,j}^{(j)}|}{\sum_{k=1}^n |a_{p,k}|} \geq \frac{|a_{i,j}^{(j)}|}{\sum_{k=1}^n |a_{i,k}|} \quad \text{für alle } i = j, j + 1, \dots, n.$$

Die Genauigkeit einer berechneten Lösung \tilde{x} eines linearen Gleichungssystems $Ax = b$ läßt sich durch **Nachiteration**, auch **Defektkorrektur** genannt, verbessern, indem man die Residuaengleichung

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

nach dem Fehler $h := x - \tilde{x}$ auflöst:

- (1) Berechne Residuum $r := b - A\tilde{x}$ (doppelt genau).
- (2) Löse $Ah = r$ (verwende LR-Zerlegung von A).
- (3) Setze $x := \tilde{x} + h$.

- ① Einführung und Begriffe
- ② Gleitpunktarithmetik und Fehleranalyse
- ③ Direkte Verfahren zur Lösung linearer Gleichungssysteme
 - 3.1 Vorbemerkungen
 - 3.2 Störungstheorie
 - 3.3 Das Lösen von Dreieckssystemen
 - 3.4 Gauß-Elimination
 - 3.5 Pivotisierung
 - 3.6 Equilibrierung und Nachiteration
 - 3.7 Stabilität bei der Gauß-Elimination
- ④ Direkte Verfahren für spezielle Systeme

Nicht verwechseln darf man:

- **Die Stabilität eines mathematischen Problems** (hier: eines linearen Gleichungssystems mit invertierbarer Koeffizientenmatrix), die beschreibt, wie sich die Lösung **bei exakter Rechnung** verändert, wenn die Daten gestört werden.
- **Die Stabilität eines numerischen Verfahrens** (hier: der Gauß-Elimination mit Spaltenpivotsuche), die beschreibt, wie sich die **in Gleitpunktarithmetik berechnete** Lösung von der exakten Lösung unterscheidet.

Beispiel: (Instabilität des Problems)

$$A = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}, \quad \tilde{\mathbf{b}} = \begin{bmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{bmatrix},$$

$$\mathbf{x} := A^{-1}\mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{x}} := A^{-1}\tilde{\mathbf{b}} = \begin{bmatrix} 9.2 \\ -12.6 \\ 4.5 \\ -1.1 \end{bmatrix}$$

Erklärung: $\text{cond}_{\infty}(A) = 4.488 \cdot 10^3$, also

$$13.6 = \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_{\infty}}{\|\mathbf{x}\|_{\infty}} \lesssim \text{cond}_{\infty}(A) \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|_{\infty}}{\|\mathbf{b}\|_{\infty}} = (4.488 \cdot 10^3) \cdot \frac{0.1}{33} = 13.6 .$$

Stabilität bei der Gauß-Elimination

Beispiel Hilbert-Matrix

Ein weiteres Beispiel: Die **Hilbert-Matrix** der Dimension n

$$H_n = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{bmatrix}$$

ist symmetrisch und positiv definit, insbesondere also invertierbar.

Ihre Konditionszahl $\text{cond}_2(H_n)$ wächst wie $e^{7n/2} \approx 33^n$:

n	4	6	8	10	12	14
$\text{cond}_2(H_n)$	$1.6 \cdot 10^4$	$1.5 \cdot 10^7$	$1.5 \cdot 10^{10}$	$1.6 \cdot 10^{13}$	$1.7 \cdot 10^{16}$	$1.9 \cdot 10^{19}$
$\text{cond}_\infty(H_n)$	$2.8 \cdot 10^4$	$2.9 \cdot 10^7$	$3.4 \cdot 10^{10}$	$3.5 \cdot 10^{13}$	$4.1 \cdot 10^{16}$	$4.5 \cdot 10^{19}$

Stabilität bei der Gauß-Elimination

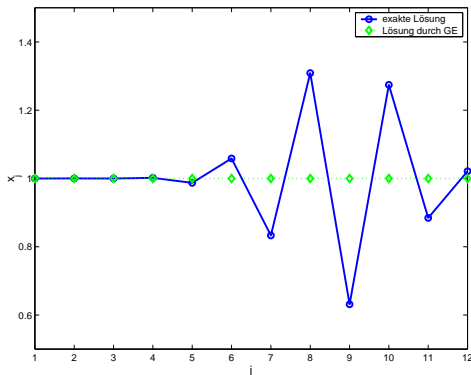
Beispiel Hilbert-Matrix

Wir lösen

$$H_{12}\mathbf{x} = \mathbf{b}$$

mit Rundungseinheit $u = 2^{-53} \approx 1.1 \cdot 10^{-16}$.

(Die rechte Seite \mathbf{b} wurde so gewählt, dass $A^{-1}\mathbf{b} = [1, \dots, 1]^T$ gilt.)



Stabilität bei der Gauß-Elimination

Stabilitätssatz für GE mit Pivotsuche

Bezeichnungen. Für $A = [a_{i,j}], B = [b_{i,j}] \in \mathbb{R}^{n \times n}$ bedeute $A \leq B$, dass $a_{i,j} \leq b_{i,j}$ für alle $1 \leq i, j \leq n$ erfüllt ist. Außerdem sei $|A| := [|a_{i,j}|] \in \mathbb{R}^{n \times n}$.

Satz 3.16 (Wilkinson, 1963)

Es seien $A \in \mathbb{C}^{n \times n}$ regulär und $\mathbf{b} \in \mathbb{C}^n$. In Gleitpunktarithmetik (mit Rundungseinheit u) werde (mit beliebiger Pivotstrategie) eine LR-Zerlegung $\tilde{L}\tilde{R} \sim PA$ von A sowie eine Lösung $\tilde{\mathbf{x}}$ des Systems $A\mathbf{x} = \mathbf{b}$ berechnet. Dann gelten:

- (a) $\tilde{L}\tilde{R} = PA + E$ mit $|E| \leq nu |\tilde{L}| |\tilde{R}| + O(u^2)$.
- (b) $(A + \Delta A)\tilde{\mathbf{x}} = \mathbf{b}$ mit $|\Delta A| \leq 3nu |\tilde{L}| |\tilde{R}| + O(u^2)$.

Definition 3.17

Eine Norm $\|\cdot\|$ auf \mathbb{C}^n bzw. $\mathbb{C}^{n \times n}$ heißt **absolut**, wenn

$$\|\mathbf{x}\| = |\mathbf{x}| \quad \forall \mathbf{x} \in \mathbb{C}^n \quad \text{bzw.} \quad \|A\| = |A| \quad \forall A \in \mathbb{C}^{n \times n}$$

gilt. Für $\mathbf{x} = [x_j]_{1 \leq j \leq n}$ und $A = [a_{i,j}]_{1 \leq i,j \leq n}$ sind dabei

$$|\mathbf{x}| := [|x_j|]_{1 \leq j \leq n} \quad \text{sowie} \quad |A| := [|a_{i,j}|]_{1 \leq i,j \leq n}.$$

$\|\cdot\|_1$, $\|\cdot\|_\infty$, $\|\cdot\|_F$ sind absolute Matrixnormen, während $\|\cdot\|_2$ nicht absolut ist.

Bemerkung. Eine Norm auf \mathbb{C}^n bzw. $\mathbb{C}^{n \times n}$ ist genau dann absolut, wenn

$$|\mathbf{x}| \leq |\mathbf{y}| \quad (\mathbf{x}, \mathbf{y} \in \mathbb{C}^n) \quad \text{stets} \quad \|\mathbf{x}\| \leq \|\mathbf{y}\| \quad \text{bzw.}$$

$$|A| \leq |B| \quad (A, B \in \mathbb{C}^{n \times n}) \quad \text{stets} \quad \|A\| \leq \|B\| \quad \text{impliziert.}$$

Dabei ist $\mathbf{x} \leq \mathbf{y} := x_j \leq y_j \quad \forall j = 1, 2, \dots, n$ bzw. $A \leq B := a_{i,j} \leq b_{i,j} \quad \forall i, j = 1, 2, \dots, n$.

Bemerkung 3.18

(a) Für absolute Normen $\|\cdot\|$ gelten somit

$$\|E\| \leq nu\|\tilde{L}\|\|\tilde{R}\| + O(u^2) \quad \text{sowie} \quad \|\Delta A\| \leq 3nu\|\tilde{L}\|\|\tilde{R}\| + O(u^2).$$

(b) Die Konstante $3\gamma_n + \gamma_n^2$ im Beweis von Satz 3.16 lässt sich verbessern zu $2\gamma_n$.

Korollar 3.19

Bei Spaltenpivotsuche gilt mit den Bezeichnungen aus Satz 3.16

$$\|\Delta A\|_{\infty} \leq 3 \rho n^3 \|A\|_{\infty} u + O(u^2),$$

wobei

$$\rho := \frac{\max_{1 \leq i, j \leq n} |\tilde{r}_{i,j}|}{\max_{1 \leq i, j \leq n} |a_{i,j}|}$$

der sog. **Wachstumsfaktor** bei Gauß-Elimination mit Spaltenpivotsuche ist. Es gilt

$$\rho \leq 2^{n-1}$$

und diese obere Schranke wird auch angenommen.

Bemerkung 3.20

Mit Hilfe von Bemerkung 3.18a lässt sich zeigen, dass sogar

$$\|\Delta A\|_{\infty} \leq 2n^2 \gamma_n \rho \|A\|_{\infty}.$$

Allerdings beobachtet man in der Praxis keineswegs, dass ρ wie 2^{n-1} wächst, sondern eher ein Anwachsen wie $n^{2/3}$.

Stabilität bei der Gauß-Elimination

Stabilitätssatz für GE mit Pivotsuche

Rückwärtsanalyse (interpretiere Rundfehler als Datenfehler)

$$(A + \Delta A)\tilde{\mathbf{x}} = \mathbf{b} \quad \text{mit} \quad \|\Delta A\|_{\infty} \lesssim 3 \rho n^3 \|A\|_{\infty} u$$

und **Konditionsanalyse** (wie wirken sich Datenfehler in *exakter Arithmetik* auf das Ergebnis aus)

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \lesssim \text{cond}(A) \left[\frac{\|A - \tilde{A}\|}{\|A\|} + \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{b}\|} \right] \quad \text{mit} \quad \text{cond}(A) := \|A\| \|A^{-1}\|$$

liefern insgesamt

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_{\infty}}{\|\mathbf{x}\|_{\infty}} \lesssim 3 n^3 \rho \text{cond}_{\infty}(A) u.$$

Gauß-Elimination mit Spaltenpivotsuche ist stabil, wenn Wachstumsfaktor ρ und Konditionszahl $\text{cond}(A)$ klein sind.

Stabilität bei der Gauß-Elimination

Stabilitätssatz für GE mit Pivotsuche

Wir haben gesehen, dass

$$\rho \leq 2^{n-1} \text{ für alle } A \in \mathbb{R}^{n \times n}$$

gilt und dass diese Schranke angenommen wird. In der 'Praxis' wächst ρ aber wesentlich langsamer mit n , so dass Gauß-Elimination mit Spaltenpivotsuche **praktisch rückwärtsstabil** ist.

Ob dieser Algorithmus auch vorwärtsstabil ist, hängt von A bzw. von $\text{cond}(A)$ ab. Wie wir am Beispiel der Hilbert-Matrix gesehen haben, gibt es invertierbare Matrizen A , die extrem schlecht konditioniert sind, aber auf den ersten Blick ganz unverdächtig aussehen.

Stabilität bei der Gauß-Elimination

Vollständige Pivotsuche

Vollständige Pivotsuche (complete pivoting): Hier zeigte Wilkinson 1961 für den Wachstumsfaktor $\rho = \rho_{CP}(n)$

$$\rho^{CP}(n) \leq \sqrt{n \cdot 2 \cdot 3^{1/2} \dots n^{1/(n-1)}} \sim cn^{1/2} n^{\frac{1}{4} \log n},$$

und dass diese Schranke nicht scharf ist. Diese Schranke wächst wesentlich langsamer als 2^{n-1} , kann aber auch beträchtlich werden: sie beträgt z.B. 3570 für $n = 100$. Bekannte Teilergebnisse:

- $\rho^{CP}(2) = 2$
- $\rho^{CP}(3) = 9/4$
- $\rho^{CP}(4) = 4$
- $\rho^{CP}(5) < 5.005$

Die lang anhaltende Vermutung $\rho^{CP}(n) \leq n$ wurde 1991 von Gould durch ein (13×13) Gegenbeispiel widerlegt.

Stabilität bei der Gauß-Elimination

Stimmen zu Theorie und Praxis

Die Lücke zwischen dem theoretisch schlimmsten Fall und dem in der Praxis beobachteten Wachstumsverhalten von ρ (beide Pivotstrategien) ist nach wie vor verblüffend:

"It is rare for the growth factor to exceed 10 on 'real' problems, regardless of the size of n ."

*P. Gill, W. Murray & M. Wright,
Numerical Linear Algebra and Optimization (1991)*

"In practice g_{PP} is always n or less."

J. Demmel, Applied Numerical Linear Algebra (1997)

"The growth suggested by the bound rarely occurs in practice. The reasons are not well understood."

G. W. Stewart, Matrix Algorithms I (1998)

"Anyone that unlucky has already been hit by a truck."

wird J. Wilkinson zugeschrieben