Vorlesung Numerische Mathematik 1

Kapitel 4: Numerische Lösung linearer Gleichungssysteme

19. Oktober 2017

Zürcher Hochschule
für Angewandte Wissenschaften

Gliederung des Kapitels

Numerische Lösung linearer Gleichungssysteme

- Lineare Gleichungssysteme treten in vielen Anwendungen der Numerik, Physik, Technik, Betriebswirtschaftslehre etc. auf, wie zum Beispiel
  - Newton-Verfahren für *nichtlineare* Gleichungssysteme, wo bei jedem Schritt lineare Gleichungssysteme auftreten;
  - die Methode der kleinsten Quadrate von Gauss in der Ausgleichsrechnung;
  - die numerische Lösung von Randwertproblemen bei gewöhnlichen und partiellen Differentialgleichungen mit Hilfe von Differenzenverfahren;
  - bei der Interpolation mittels Splines;
  - die Behandlung von Eigenwertproblemen in der mathematischen Physik;
  - in der Elektrotechnik die Berechnung von Netzwerken (Ströme zu vorgegebenen Spannungen und Widerständen); in der Betriebswirtschaftslehre bei der linearen Programmierung uvm.

Lernziele

- Sie können lineare Gleichungssysteme selbst aufstellen.
- Sie können den Gauss-Algorithmus mit und ohne Pivotisierung sowie die $LR-$Zerlegung auf konkrete Problemstellungen anwenden.
- Sie kennen die Cholesky-Zerlegung.
- Sie können die Fehler für gestörte lineare Gleichungssysteme berechnen.
- Sie können das Jacobi- sowie das Gauss-Seidel-Verfahren anwenden und in MATLAB implementieren.
- Sie beherrschen die zugehörigen Fehlerabschätzungen.

Historische Entwicklung

Historische Entwicklung

- Auch lineare Gleichungssystem beschäftigten Mathematiker schon vor Tausenden von Jahren.
- Eine Aufgabe, die rund 4000 Jahre alt ist und aus Mesopotamien stammt, lautet: "Ein Viertel der Breite zur Länge addiert ergibt 7 Handbreiten, Länge und Breite addiert macht 10 Handbreiten".

Historische Entwicklung

- Natürlich beschäftigten sich auch die Ägypter mit ähnlichen Problemen, wie z.B. der folgenden Aufgabe aus dem 'Papyrus Moskau' ca. 2000 v.Chr.:
  - "Berechne die Länge und Breite eines Rechteckes der Fläche 12, wenn die Breite 3/4 der Länge ist".
- In der heutigen Schreibweise würden wir das erste Beispiel als System zweier Gleichungen mit zwei Unbekannten formulieren (mit x als Breite und y als Länge):

$$\begin{aligned} \frac{1}{4}x + y &= 7 \\ x + y &= 10 \end{aligned}$$

bzw. im Matrizenkalkül

$$\begin{pmatrix} \frac{1}{4} & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 7 \\ 10 \end{pmatrix}$$

Historische Entwicklung

- Die Babylonier oder die Ägypter kannten kein Matrizenkalkül.
- Die Chinesen kamen dem zwischen 200 bis 100 v.Chr. schon bedeutend näher, wie im chinesischen Mathematikbuch Jiu Zhang Suanshu (dt. 'Neun Kapitel der Rechenkunst' od. 'Neun Bücher arithmetischer Technik') aus dieser Zeit festgehalten ist, welches die chinesische Mathematik und diejenige der umliegenden Länder bis ins 17. Jhr. prägte.
- So wurde darin bereits das Verfahren beschrieben, welches wir heute als Gauss-Algorithmus kennen[1].

[1] siehe z.B. MacTutor unter http://www-history.mcs.st-and.ac.uk/HistTopics/Matrices_and_determinants.html

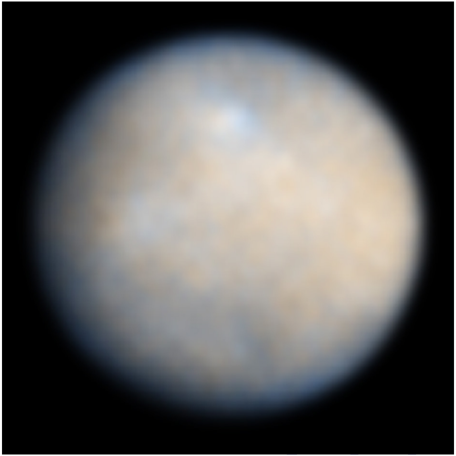
Historische Entwicklung

- Die erste systematische Untersuchung von linearen Gleichungssystemen wird Gottfried Wilhelm Leibniz (1646-1716) zugeschrieben.
- Er führte die Formeln für Determinanten für 2x2 und 3x3 Gleichungssysteme ein.
- Gabriel Cramer (1704-1752) entwickelte die nach ihm benannte allgemeine Lösungsformel für Systeme von n Gleichung mit n Unbekannten.
- Seine Regel benötigt allerdings einen enormen Rechenaufwand von rund $n(n+1)!$ Gleitkommaoperationen.
- Für $n = 10$ benötigt man bereits fast 400 Mio. Punktoperationen und für $n = 20$ bereits 10^{21}. Deshalb ist die Cramersche Regel in der Praxis völlig unbrauchbar (dies gilt bereits für $n = 3$).

Historische Entwicklung

- Der deutsche Mathematiker, Physiker, Astronom und Geodät Carl Friedrich Gauss (1777-1855) betrachtete lineare Gleichungssysteme im Zusammenhang mit astronomischen Problemen.
- So gelang es ihm, den Zwergplaneten Ceres im Asteroidengürtel zwischen Mars und Jupiter, der 1801 entdeckt und gleich darauf wieder verlorengegangen war, aufgrund seiner Berechnungen basierend auf der Methode der kleinsten Quadrate wieder zu finden.

Der Zwergplanet Ceres (NASA, ESA, J. Parker (Southwest Research Institute), P. Thomas (Cornell University), and L. McFadden (University of Maryland, College Park), http://de.wikipedia.org/wiki/(1)_Ceres):



Historische Entwicklung

Grössenvergleich von Ceres:

Historische Entwicklung

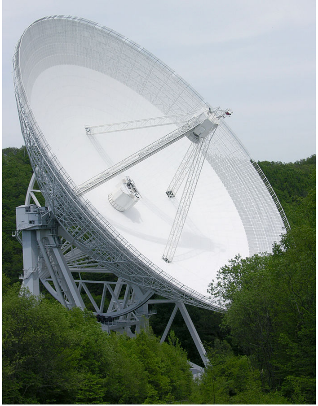
- 1811 entwickelte Gauss den nach ihm benannten Gauss-Algorithmus (vgl. Kap. 4.3), eines der heutigen Standardverfahren zur Lösung von linearen Gleichungssystemen.
- Der Gauss-Algorithmus benötigt für die Lösung eines $n \times n$ Gleichungssystem lediglich $\frac{2}{3}n^3 + \frac{5}{2}n^2 - \frac{13}{6}n$ Punktoperationen (vgl. Kap. 4.6.2), d.h. für $n = 20$ also nur rund 6000 im Gegensatz zu 10^{21} bei der Cramerschen Regel.

Historische Entwicklung

- Ausgehend von den Untersuchungen linearer Gleichungssysteme entwickelte sich daraus das Gebiet der linearen Algebra, unter anderem basierend auf den Werken von Wiliam Rowan Hamilton (1805-1865; Vektoren, Quaternionen), Hermann Grassmann (1809-1877; endlichdimensonale Vektorräume), Arthur Cayley (1821-1895; Matrizen als algebraische Objekte), Camille Jordan (1838-1922; Jordansche Normalform), Ferdinand Georg Frobenius (1849-1917; Gruppentheorie), Maxime Bôcher (1867-1918; *Introduction to higher algebra*), Herbert Westren Trunbull (1885-1961) und Alexander Aitken (1895-1967) mit *Introduction to the Theory of Canonical Matrices* sowie Leon Mirsky (1918-1983) mit *An introduction to linear algebra.*

Beispiele aus der Praxis
Parabolantenne

- Parabolantenne der Firma Krupp mit 100 m Durchmesser am oberen Rand



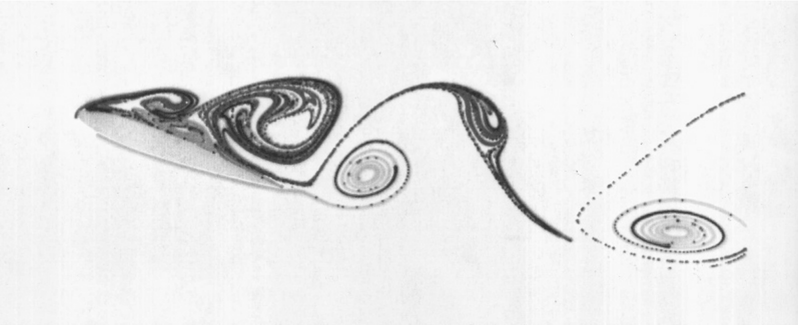
Beispiele aus der Praxis
Parabolantenne

Beispiele aus der Praxis

Parabolantenne

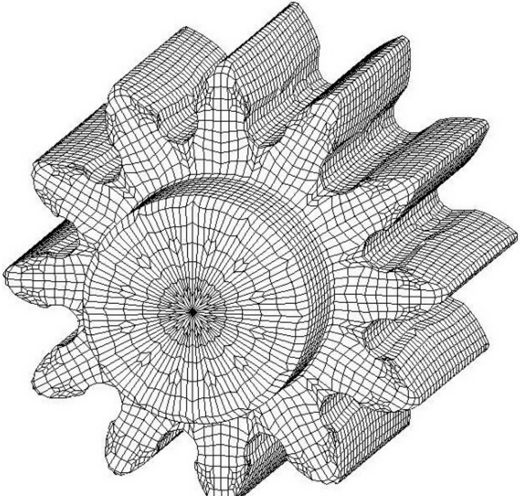
- Es handelt sich dabei um einen räumlichen Verbund aus Stäben und Balken, die geometrisch ein Rotationsparaboloid bilden.
- Die Berechnung muss so erfolgen, dass bei Verformung durch Neigung und Eigengewicht wegen der Richtgenauigkeit der Antenne immer wieder ein Rotationsparaboloid entsteht.
- Es sind jeweils ca. 5000 Gleichungen mit 5000 Unbekannten zu lösen.
- Nur der Empfänger muss dann jeweils in den neuen Brennpunkt nachgeführt werden. Für jede neue Einstellung beträgt die mittlere Abweichung vom idealen Paraboloid weniger als 0.6 mm (2012).

- Beispiel des Aerodynamischen Instituts der RWTH Aachen.



- Numerische Simulation einer ablösenden Strömung um Tragflügelprofile, gerechnet mit den Navier-Stokes-Gleichungen:
- Wenn 3-dimensional gerechnet wird und ein $(31 \times 31 \times 51)$-Gitter mit je 4 Gleichungen verwendet wird, so erhält man nichtlineare Systeme aus 196 044 Gleichungen mit 196 044 Unbekannten, die iterativ (etwa mit 5 Iterationen) gelöst werden.
  - Rechnet man bis zum Wirbelablösen 10 000 Zeitschritte, so ergeben sich $5 \times 10\,000 = 50\,000$ lineare Gleichungssysteme aus rund ca. 200 000 Gleichungen, die zu lösen sind.

- Ein Finite-Element-Beispiel aus dem Institut für Bildsame Formgebung der RWTH Aachen:



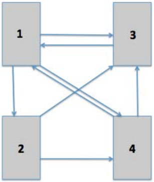
- Bei der Simulation des Fließpress-Verfahrens zur Herstellung eines Zahnrades mit zwölf Zähnen wird unter Ausnutzung der Symmetriebedingungen mit dem Modell eines halben Zahnes gerechnet.
- In diesem Beispiel wird dazu ein Netz mit 2911 Knoten erstellt. Man erhält unter Berücksichtigung aller Randbedingungen insgesamt 7560 nichtlineare Gleichungen, die iterativ gelöst werden. Dabei tritt eine Bandmatrix auf.

Beispiele aus der Praxis
PageRank

- Der PageRank-Algorithmus, welcher auf die Gründer von Google, Sergey Brin und Lawrence Page, zurückgeht[2], erlaubt die Klassifizierung einer Menge von verlinkten Dokumenten, z.B. der Seiten des Internets, nach ihrer "Wichtigkeit", bzw. dem ***rank***.
- Die zugrunde liegende Idee ist, dass eine Seite umso wichtiger ist, je mehr Links von anderen wichtigen Seiten auf sie zeigen.
- Zur Bestimmung der Wichtigkeit wird die **PageRank**- bzw. **Google-Matrix** benötigt.
- Diese Matrix repräsentiert einen gerichteten Graphen, wobei die Knoten des Graphen den Web-Seiten entsprechen und die Kanten den Links dazwischen.

[2] Sergei Brin, Lawrence Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Computer Networks and ISDN Systems, Band 30, 1998, S. 107-117

Beispiele aus der Praxis
PageRank

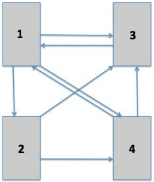


Beispiel: ein einfaches Web mit 4 Seiten ist in obiger Abb. dargestellt. Ein Pfeil von Seite i zur Seite j entspricht einem Link. Die Bedeutung der Web-Seiten wird durch den Vektor

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

angegeben, wobei $x_i \in \mathbb{R}$ die Wichtigkeit der Seite i angibt.

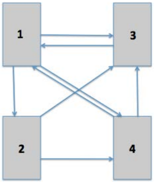
Beispiele aus der Praxis
PageRank



Für die Seite 1 ergibt sich z.B.

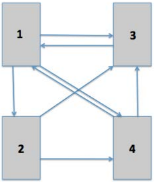
$$x_1 = 0 \cdot x_1 + 0 \cdot x_2 + 1 \cdot x_3 + \frac{1}{2} \cdot x_4,$$

da die Seite 1 keinen Link auf sich selber $(0 \cdot x_1)$ oder von Seite 2 $(0 \cdot x_2)$ hat, jedoch je einen Link von Seite 3 und Seite 4. Da Seite 3 insgesamt nur einen ausgehenden Link aufweist, erhält dieser für Seite 1 das volle Gewicht $(1 \cdot x_3)$. Da Seite 4 aber 2 ausgehende Links aufweist, erhält der Link auf Seite 1 nur das Gewicht $\frac{1}{2}$ (also $\frac{1}{2} \cdot x_4$).



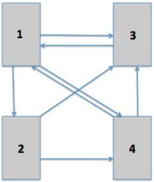
Für alle vier Seiten erhält man so das lineare Gleichungssystem:

$$
\begin{aligned}
x_1 &= 0x_1 + 0x_2 + 1x_3 + \frac{1}{2}x_4 \\
x_2 &= \frac{1}{3}x_1 + 0x_2 + 0x_3 + 0x_4 \\
x_3 &= \frac{1}{3}x_1 + \frac{1}{2}x_2 + 0x_3 + \frac{1}{2}x_4 \\
x_4 &= \frac{1}{3}x_1 + \frac{1}{2}x_2 + 0x_3 + 0x_4
\end{aligned}
$$



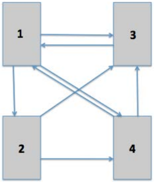
Oder in Matrixschreibweise:

$$\boldsymbol{x} = \boldsymbol{P} \cdot \boldsymbol{x}, \quad \text{mit } \boldsymbol{P} = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{pmatrix}$$



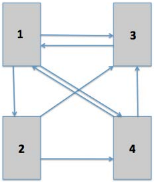
Oder in Matrixschreibweise:

$$\boldsymbol{x} = \boldsymbol{P} \cdot \boldsymbol{x}, \quad \text{mit } \boldsymbol{P} = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{pmatrix}$$



Also ist $\boldsymbol{x}$ ein Eigenvektor von $\boldsymbol{P}$ zum Eigenwert 1. Dies ist zudem eine Fixpunktgleichung und kann gemäss Kap. 3.4 iterativ gelöst werden. Mit dem Startvektor $\boldsymbol{x_0} = (1,1,1,1)^T$ erhalten wir mittels der Fixpunktiteration $\boldsymbol{x}_{i+1} = \boldsymbol{P}\boldsymbol{x}_i$ die Näherungslösung

$$\boldsymbol{x}_1 = \boldsymbol{P}\boldsymbol{x}_0 = \begin{pmatrix} 1.5000 \\ 0.3333 \\ 1.3333 \\ 0.8333 \end{pmatrix}, \boldsymbol{x}_2 = \boldsymbol{P}\boldsymbol{x}_1 = \begin{pmatrix} 1.75 \\ 0.5 \\ 1.0833 \\ 0.6667 \end{pmatrix}, \dots, \boldsymbol{x}_{16} = \boldsymbol{P}\boldsymbol{x}_{15} = \begin{pmatrix} 1.5484 \\ 0.5161 \\ 1.1613 \\ 0.7742 \end{pmatrix}$$



Also hat die Seite 1 die höchste Wichtigkeit, Seite 3 die zweithöchste, Seite 4 die dritthöchste, und Seite 2 die vierhöchste bzw. die niedrigste Wichtigkeit. Unter Verwendung der Einheitsmatrix $\boldsymbol{I}$ lässt sich $\boldsymbol{x}$ auch mit dem aus der linearen Algebra bereits bekannten und in Kap. 4.3 nochmals detailliert beschriebenen Gauss-Algorithmus als eine Lösung der homogenen Gleichung

$$(\boldsymbol{P} - \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$$

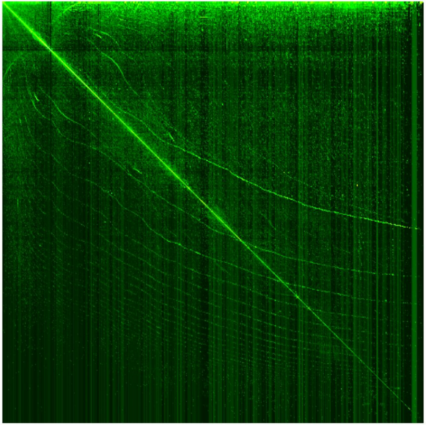
bestimmen.

- Um auch zufälliges Hüpfen zwischen den Seiten (ohne Benützung von Links) abbilden zu können, wird die Matrix $\boldsymbol{P}$ noch modifiziert mit einer Matrix $\boldsymbol{S}$, deren Elemente alle den Wert $\frac{1}{n}$ haben bei einem Web mit n Seiten.
- Die **Google-Matrix** $\boldsymbol{G}$ erhält man als Überlagerung der beiden Matrizen:

$$\boldsymbol{G} = \alpha \boldsymbol{P} + (1-\alpha)\boldsymbol{S}.$$

- Dabei ist $0 \leq \alpha \leq 1$ eine Faktor, der das zufällige Hüpfen modelliert (für $\alpha = 1$ findet kein zufälliges Hüpfen statt, für $\alpha = 0$ findet ausschliesslich zufälliges Hüpfen statt).
- Die Erfinder des PageRank-Algorithmus wählten $\alpha = 0.85$.

Google Matrix des Netzwerks der Cambridge Universtität aus dem Jahr 2006 mit $n = 212710$ (http://arxiv.org/abs/1106.6215, GFDL, Wikimedia Commons):



Problemstellung

Problemstellung

- Gesucht ist eine Lösung zu einem linearen Gleichungssystem mit n Gleichungen und n Unbekannten der Form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n &= b_n \end{aligned} \tag{1}$$

- Üblicherweise schreibt man solche Gleichungssysteme in Matrix-Form, nämlich als $\boldsymbol{Ax} = \boldsymbol{b}$ mit

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

- $\boldsymbol{A}$ und $\boldsymbol{b}$ sind gegeben, $\boldsymbol{x}$ ist gesucht.
- Bezüglich der Notation werden in diesem Skript Matrizen mit fettgedruckten Grussbuchstaben und Vektoren mit fettgedruckten Kleinbuchstaben hervorgehoben.

Problemstellung

- Gewisse Eigenschaften der Matrix $\boldsymbol{A} = (a_{ij})$ entscheiden darüber, was für ein Verfahren zur Lösung des Gleichungssystems sinnvoll eingesetzt werden kann.
- Da die Anzahl n der Gleichungen der Anzahl Unbekannten $x_1, \ldots, x_n$ entspricht, ist A eine quadratische Matrix der Dimension $n \times n$.
- Für quadratische Matrizen $\boldsymbol{A}$ wissen wir aus der linearen Algebra, dass genau dann eine eindeutige Lösung existiert, wenn die Determinante $det(\boldsymbol{A})$ nicht verschwindet (gleichbedeutend mit A ist invertierbar bzw. A ist regulär), d.h. wenn eine Matrix $\boldsymbol{A}^{-1}$ existiert, so dass $\boldsymbol{A} \cdot \boldsymbol{A}^{-1} = \boldsymbol{A}^{-1} \cdot \boldsymbol{A} = \boldsymbol{I}$, wobei $\boldsymbol{I}$ die $n \times n$ Einheitsmatrix ist (die Einträge auf der Diagonalen sind 1, alle anderen Einträge sind 0).

Problemstellung

- Bei der numerischen Lösung von Systemen der Art $\boldsymbol{Ax} = \boldsymbol{b}$ unterscheidet man zwischen
  - direkten Verfahren
    - Mit einem direkten Verfahren erhält man mit einer endlichen Zahl von Rechenschritten die exakte Lösung (wenn man Rundungsfehler vernachlässigt)
  - iterativen Verfahren
    - Hier wird eine Folge von Vektoren erzeugt, die gegen die Lösung von $\boldsymbol{Ax} = \boldsymbol{b}$ konvergiert.
- Wir beginnen mit den direkten Verfahren.

Problemstellung

- Hierfür benötigen wir die Definition der oberen bzw. unteren Dreiecksform.

Definition 4.1: Untere Dreiecksmatrix / Obere Dreiecksmatrix [6]

- Eine $n \times n$ Matrix $\boldsymbol{L} = (l_{ij})$ heisst **untere Dreiecksmatrix**, wenn $l_{ij} = 0$ für $j > i$ gilt; sie heisst **normierte untere Dreiecksmatrix**, wenn ausserdem $l_{ii} = 1$ für alle i gilt.
- Eine $n \times n$ Matrix $\boldsymbol{R} = (r_{ij})$ heisst **obere Dreiecksmatrix**, wenn $r_{ij} = 0$ für $i > j$ gilt; sie heisst **normierte obere Dreiecksmatrix**, wenn ausserdem $r_{ii} = 1$ für alle i gilt.

Problemstellung

Beispiel 4.1

- Untere normierte Dreiecksmatrix:

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn-1} & 1 \end{pmatrix}$$

- Obere Dreiecksmatrix:

$$\boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ 0 & r_{22} & r_{23} & \cdots & r_{2n} \\ 0 & 0 & r_{33} & \cdots & r_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & r_{nn} \end{pmatrix}$$

Der Gauss-Algorithmus

- Das Eliminationsverfahren nach Gauss (der "Gauss-Algorithmus") ist ein anschauliches Verfahren, das zudem gut implementiert werden kann.
- Es beruht auf der Tatsache, dass ein lineares Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ leicht lösbar ist, falls die Matrix $\boldsymbol{A}$ in oberer Dreiecksform vorliegt, d.h. alle Elemente unterhalb der Diagonalen verschwinden.

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix}$$

- In diesem Fall kann man für $\boldsymbol{Ax} = \boldsymbol{b}$ mittels der folgenden rekursiven Vorschrift, dem sogenannten Rückwärtseinsetzen, die Komponenten von x berechnen: $x_n = \frac{b_n}{a_{nn}}$, $x_{n-1} = \frac{b_{n-1} - a_{n-1\,n} x_n}{a_{n-1\,n-1}}$, $\ldots$, $x_1 = \frac{b_1 - a_{12}x_2 - \ldots - a_{1n}x_n}{a_{11}}$
- oder, kompakt geschrieben,

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}, \qquad i = n, n-1, \ldots, 1.$$

- Die Idee des Gauß'sche Eliminationsverfahren ist nun, ein beliebiges Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ umzuformen in ein äquivalentes Gleichungssystem $\widetilde{\boldsymbol{A}}\boldsymbol{x} = \widetilde{\boldsymbol{b}}$, so dass die Matrix $\widetilde{\boldsymbol{A}}$ in als obere Dreiecksmatrix vorliegt.
- Bei dieser Transformation sind folgende Umformungen zugelassen:
  - $z_j \equiv z_j - \lambda z_i$ mit $i < j$ $(\lambda \in \mathbb{R})$, wobei z_i die i-te Zeile des Gleichungssystems bezeichnet
  - $z_i \to z_j$: Vertauschen der i-ten und j-ten Zeile im System

- Es soll folgendes System $\boldsymbol{Ax} = \boldsymbol{b}$ gelöst werden, wobei

$$\boldsymbol{A} = \begin{pmatrix} 1 & 5 & 6 \\ 7 & 9 & 6 \\ 2 & 3 & 4 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 29 \\ 43 \\ 20 \end{pmatrix}$$

- Wir subtrahieren das 7-fache der erste Zeile von der zweiten Zeile ($z_2 \equiv z_2 - 7z_1$) und erhalten

$$\boldsymbol{A}_1 = \begin{pmatrix} 1 & 5 & 6 \\ 0 & -26 & -36 \\ 2 & 3 & 4 \end{pmatrix} \text{ und } \boldsymbol{b}_1 = \begin{pmatrix} 29 \\ -160 \\ 20 \end{pmatrix}.$$

- Anschliessend subtrahieren wir 2-mal die erste Zeile von der letzten ($z_3 \equiv z_3 - 2z_1$) und erhalten

$$\boldsymbol{A}_2 = \begin{pmatrix} 1 & 5 & 6 \\ 0 & -26 & -36 \\ 0 & -7 & -8 \end{pmatrix} \text{ und } \boldsymbol{b}_2 = \begin{pmatrix} 29 \\ -160 \\ -38 \end{pmatrix}.$$

- Im letzten Schritt subtrahieren wir 7/26-mal die zweite Zeile von der dritten ($z_3 \equiv z_3 - \frac{7}{26} z_2$):

$$\boldsymbol{A}_3 = \begin{pmatrix} 1 & 5 & 6 \\ 0 & -26 & -36 \\ 0 & 0 & \frac{22}{13} \end{pmatrix} \text{ und } \boldsymbol{b}_3 = \begin{pmatrix} 29 \\ -160 \\ \frac{66}{13} \end{pmatrix}.$$

- Somit erhalten wir über Rückwärtseinsetzen die gesuchten Komponenten von x:

$$x_3 = \frac{\frac{66}{13}}{\frac{22}{13}} = 3,\ x_2 = \frac{-160 - 3 \cdot (-36)}{-26} = 2$$

$$\text{und } x_1 = \frac{29 - 2 \cdot 5 - 3 \cdot 6}{1} = 1.$$

- Für die Programmierung des Gauss-Algorithmus sollte nun folgendermassen vorgegangen werden:
- Zuerst erzeugt man Nullen in der ersten Spalte unterhalb von a_{11} mit der Operation $z_j \equiv z_j - \frac{a_{j1}}{a_{11}} z_1$ mit $j = 2, \ldots, n$.
  - Dies geht nur, falls $a_{11} \neq 0$. Ist $a_{11} = 0$, so vertauschen wir die erste Zeile mit der i-ten Zeile, wobei $a_{i1} \neq 0$ sein muss. Falls alle Zeilen der Matrix in der ersten Zeile eine Null besitzen funktioniert die Vertauschung nicht. Dann ist allerdings auch die Matrix nicht regulär und die Lösungsmenge kann leer sein oder auch unendlich viele Elemente enthalten.
- Dieser Schritt wird nun wiederholt, in dem man mit der zweiten Spalte fortfährt und unterhalb der Diagonalen Nullen erzeugt.

Gauss-Algorithmus zur Transformation von $Ax = b$ auf ein oberes Dreiecksystem [1]

- für $i = 1, ..., n-1$:
  erzeuge Nullen unterhalb des Diagonalelementes in der i-ten Spalte
  - Falls nötig und möglich, sorge durch Zeilenvertauschung für $a_{ii} \neq 0$:
    falls $a_{ii} \neq 0$: tue nichts
    $$\text{falls } a_{ii} = 0 : \begin{cases} \text{falls } a_{ji} = 0 \text{ für alle } j = i+1, ..., n : \\ \quad A \text{ ist nicht regulär; stop;} \\ \text{wenn } a_{ji} \neq 0 \text{ für ein } j = i+1, ..., n : \\ \quad \text{sei } j \geq i+1 \text{ der kleinste Index mit } a_{ji} \neq 0 \\ \quad z_i \longleftrightarrow z_j \end{cases}$$
  - Eliminationsschritt:
    für $j = i+1, ..., n$ eliminiere das Element a_{ji} durch:
    $$z_j := z_j - \frac{a_{ji}}{a_{ii}} \cdot z_i$$

- Bringen Sie das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ auf die obere Dreiecksform und lösen sie nach x auf, wobei

$$\boldsymbol{A} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -3 & -2 \\ 5 & 1 & 4 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 0 \\ 5 \\ 3 \end{pmatrix}$$

- Es soll das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{c}$ mit der Matrix $\boldsymbol{A}$ aus der vorherigen Aufgabe gelöst werden für
$\boldsymbol{c} = (13, -32, 22)^T$
Lösung:

$$z_2 \equiv z_2 - \frac{1}{(-1)} z_1 \Rightarrow \boldsymbol{c}_1 = \begin{pmatrix} 13 \\ -19 \\ 22 \end{pmatrix}$$

$$z_3 \equiv z_3 - \frac{5}{(-1)} z_1 \Rightarrow \boldsymbol{c}_2 = \begin{pmatrix} 13 \\ -19 \\ 87 \end{pmatrix}$$

$$z_3 \equiv z_3 - \frac{6}{(-2)} z_2 \Rightarrow \boldsymbol{c}_3 = \begin{pmatrix} 13 \\ -19 \\ 30 \end{pmatrix}$$

Rückeinsetzen liefert die Lösung $\boldsymbol{x} = (-1, 7, 5)^T$

- Eine zusätzliche Anwendung des Gauss-Algorithmus ist die Determinantenbestimmung.
- Wenn wir mit $\widetilde{\boldsymbol{A}}$ die obere Dreiecksmatrix von A bezeichnen, dann gilt die Beziehung

$$det(\boldsymbol{A}) = (-1)^l \cdot det(\widetilde{\boldsymbol{A}}) = (-1)^l \prod_{i=1}^{n} \widetilde{a}_{ii}$$

wobei $\widetilde{a}_{ii}$ die Diagonalelemente von $\widetilde{\boldsymbol{A}}$ sind und l die Anzahl der im Laufe des Gauss-Algorithmus vorgenommen Zeilenvertauschungen.

- Bestimmen Sie die Determinante der Matrix $\boldsymbol{A}$ aus Aufgabe 3.1 mittels des Gauss-Algorithmus.

Der Gauss-Algorithmus
Aufgabe 4.3

- Während den Übungen: Implementieren Sie den Gauss-Algorithmus in MATLAB und bestimmen Sie damit die Lösungen für die untenstehenden Gleichungssysteme sowie die Determinanten der Matrizen $\boldsymbol{A}_1 - \boldsymbol{A}_4$. Wer die Aufgabe lieber von Hand löst, kann dies tun.

$$\boldsymbol{A}_1\boldsymbol{x} = \begin{pmatrix} 4 & -1 & -5 \\ -12 & 4 & 17 \\ 32 & -10 & -41 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} -5 \\ 19 \\ -39 \end{pmatrix} \text{ bzw. } = \begin{pmatrix} 6 \\ -12 \\ 48 \end{pmatrix}$$

$$\boldsymbol{A}_2\boldsymbol{x} = \begin{pmatrix} 2 & 7 & 3 \\ -4 & -10 & 0 \\ 12 & 34 & 9 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} 25 \\ -24 \\ 107 \end{pmatrix} \text{ bzw. } = \begin{pmatrix} 5 \\ -22 \\ 42 \end{pmatrix}$$

$$\boldsymbol{A}_3\boldsymbol{x} = \begin{pmatrix} -2 & 5 & 4 \\ -14 & 38 & 22 \\ 6 & -9 & -27 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} 1 \\ 40 \\ 75 \end{pmatrix} \text{ bzw. } = \begin{pmatrix} 16 \\ 82 \\ -120 \end{pmatrix}$$

$$
\boldsymbol{A}_4\boldsymbol{x} = \begin{pmatrix} -1 & 2 & 3 & 2 & 5 & 4 & 3 & -1 \\ 3 & 4 & 2 & 1 & 0 & 2 & 3 & 8 \\ 2 & 7 & 5 & -1 & 2 & 1 & 3 & 5 \\ 3 & 1 & 2 & 6 & -3 & 7 & 2 & -2 \\ 5 & 2 & 0 & 8 & 7 & 6 & 1 & 3 \\ -1 & 3 & 2 & 3 & 5 & 3 & 1 & 4 \\ 8 & 7 & 3 & 6 & 4 & 9 & 7 & 9 \\ -3 & 14 & -2 & 1 & 0 & -2 & 10 & 5 \end{pmatrix} \cdot x
$$

$$
= \begin{pmatrix} -11 \\ 103 \\ 53 \\ -20 \\ 95 \\ 78 \\ 131 \\ -26 \end{pmatrix}
$$

Fehlerfortpflanzung und Pivotisierung

Fehlerfortpflanzung und Pivotisierung

- Im vorherigen Abschnitt haben wir Zeilen nur vertauscht, falls ein Diagonalelement im Laufe der Berechnungen Null wurde.
- Man kann aber Zeilenvertauschungen aber auch dazu verwenden, um Fehler z.B. durch Gleitpunktoperationen, zu minimieren.
- In jedem Eliminationsschritt werden die Zeilen mit $\lambda = \frac{a_{ji}}{a_{ii}}$ mulitpliziert, d.h. der absolute Fehler vergrössert sich um den Faktor $|\lambda|$(siehe Kap. 2).
- Wünschenswert wäre es also, wenn $|\lambda| = |\frac{a_{ji}}{a_{ii}}| < 1$.
- Dies lässt sich einfach dadurch erreichen, dass man vor dem Eliminationsschritt überprüft, welches Element in der Spalte betragsmässig am grössten ist und die Zeile vertauscht, so dass dieses grösste Element zum Diagonalelement wird.
- Dieses Vorgehen wird Spaltenpivotisierung genannt.

Gauss-Algorithmus zur Transformation von $Ax = b$ mit Spaltenpivotisierung [1]:

- für $i = 1, \ldots, n-1$:
  erzeuge Nullen unterhalb des Diagonalelementes in der i-ten Spalte
  - Suche das betragsgrösste Element unterhalb der Diagonalen in der i-ten Spalte:
    Wähle k so, dass $|a_{ki}| = \max\{|a_{ji}| \,|\, j = i, \ldots, n\}$
    $\begin{cases} \text{falls } a_{ki} = 0: \; A \text{ ist nicht regulär; stop;} \\ \text{falls } a_{ki} \neq 0: \; z_k \longleftrightarrow z_i; \end{cases}$
  - Eliminationsschritt:
    für $j = i+1, \ldots, n$ eliminiere das Element a_{ji} durch:

$$z_j := z_j - \frac{a_{ji}}{a_{ii}} \cdot z_i$$

- Die Matrix

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

soll mittels Spaltenpivotisierung auf die (rechts-) obere Dreiecksform gebracht werden.

- Lösung (es hat einen Fehler, finden Sie ihn?):

$$\boldsymbol{A} := \begin{pmatrix} 1 & 2 & -1 \\ 4 & -2 & 6 \\ 3 & 1 & 0 \end{pmatrix} \xrightarrow{z_1 \longleftrightarrow z_2} \begin{pmatrix} 4 & -2 & 6 \\ 1 & 2 & -1 \\ 3 & 1 & 0 \end{pmatrix} \xrightarrow{z_2 := z_2 - 0.25\, z_1} \begin{pmatrix} 4 & -2 & 6 \\ 0 & 2.5 & -2.5 \\ 3 & 1 & 0 \end{pmatrix}$$

$$\xrightarrow{z_3 := z_3 - 0.75\, z_1} \begin{pmatrix} 4 & -2 & 6 \\ 0 & 2.5 & -2.5 \\ 0 & 2.5 & -4.5 \end{pmatrix} \xrightarrow{z_3 := z_3 - z_2} \begin{pmatrix} 4 & -2 & 6 \\ 0 & 2.5 & -2.5 \\ 0 & 0 & -2.5 \end{pmatrix}$$

Dreickszerlegung von Matrizen

Dreieckszerlegung von Matrizen

- In der obigen Version des Gauß–Verfahrens haben wir die Matrix $\boldsymbol{A}$ auf obere Dreiecksform gebracht und zugleich alle dafür notwendigen Operationen auch auf den Vektor $\boldsymbol{b}$ angewendet.
- Es gibt alternative Möglichkeiten, lineare Gleichungssysteme zu lösen, bei denen der Vektor $\boldsymbol{b}$ unverändert bleibt.

Die LR-Zerlegung

- Wir werden nun ein Verfahren kennen lernen, bei dem die Matrix $\boldsymbol{A}$ in ein Produkt von zwei Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ zerlegt wird, also $\boldsymbol{A} = \boldsymbol{LR}$, wobei $\boldsymbol{R}$ eine obere Dreiecksmatrix und $\boldsymbol{L}$ eine untere normierte Dreieckmatrix ist:

$$\boldsymbol{L} = \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn-1} & l_{nn} \end{pmatrix}$$

$$\boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ 0 & r_{22} & r_{23} & \cdots & r_{2n} \\ 0 & 0 & r_{33} & \cdots & r_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & r_{nn} \end{pmatrix}$$

Die LR-Zerlegung

- Die Zerlegung $\boldsymbol{A} = \boldsymbol{LR}$ wird als ***LR−Faktorisierung*** oder ***LR−Zerlegung*** bezeichnet. Das ursprüngliche Gleichungssystem

$$\boldsymbol{Ax} = \boldsymbol{b}$$

lautet dann

$$\boldsymbol{LRx} = \boldsymbol{b} \Longleftrightarrow \boldsymbol{Ly} = \boldsymbol{b} \text{ und } \boldsymbol{Rx} = \boldsymbol{y}$$

und lässt sich wie folgt in zwei Schritten lösen:

Die LR-Zerlegung

1. Zunächst löst man das Gleichungssystem $\boldsymbol{Ly} = \boldsymbol{b}$. Dies kann, ganz analog zum Rückwärtseinsetzen durch Vorwärtseinsetzen geschehen:

$$y_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} y_j}{l_{ii}},, \qquad i = 1, 2, \ldots, n.$$

2. Anschliessend löst man durch Rückwärtseinsetzen das Gleichungssystem $Rx = y$. Dann gilt

$$\boldsymbol{Ax} = \boldsymbol{LRx} = \boldsymbol{Ly} = \boldsymbol{b}$$

womit das System $\boldsymbol{Ax} = \boldsymbol{b}$ gelöst ist.

Die LR-Zerlegung

- Wir haben beim Gauss-Algorithmus bereits gesehen, dass sich eine beliebige Matrix durch Zeilenumformungen in eine obere Dreiecksfrom transformieren lässt.
- Im Folgenden gehen wir davon aus, dass Zeilenvertauschungen nicht notwendig sind.
- Das Gauß'sche Eliminationsverfahren läßt sich dann so erweitern, dass damit eine $\boldsymbol{LR}$–Zerlegung einer invertierbaren Matrix $\boldsymbol{A}$ möglich ist.

Die LR-Zerlegung

- Tatsächlich gilt:
  - $\boldsymbol{R}$ ist gerade die durch den Gauss-Algorithmus auf die obere Dreiecksform gebrachte Matrix $\widetilde{\boldsymbol{A}}$
  - Die Elemente l_{ji} von $\boldsymbol{L}$ entsprechen gerade den berechneten Faktoren λ aus den Eliminationsschritten $z_j := z_j - \lambda_{ji} z_i$, also $l_{ji} = \lambda_{ji}$

Die LR-Zerlegung

Beispiel 4.5

- Wir berechnen für die Matrix $\boldsymbol{A}$ aus Aufgabe 4.1 die normierte untere Dreiecksmatrix $\boldsymbol{L}$ und die obere Dreiecksmatrix $\boldsymbol{R}$, so dass $\boldsymbol{A} = \boldsymbol{LR}$.
  Lösung: Wir hatten

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

und

$$i=1, j=2 \Rightarrow z_2 \equiv z_2 - \frac{1}{\underbrace{(-1)}_{l_{21}}} z_1 \Rightarrow \boldsymbol{A}_1 = \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 5 & 1 & 4 \end{pmatrix}$$

$$i=1, j=3 \Rightarrow z_3 \equiv z_3 - \frac{5}{\underbrace{(-1)}_{l_{31}}} z_1 \Rightarrow \boldsymbol{A}_2 = \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 6 & 9 \end{pmatrix}$$

$$i=2, j=3 \Rightarrow z_3 \equiv z_3 - \frac{6}{\underbrace{(-2)}_{l_{32}}} z_2 \Rightarrow \boldsymbol{A}_3 = \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 0 & 6 \end{pmatrix} = \boldsymbol{R}$$

Das heisst, wir können

$$\boldsymbol{R} = \boldsymbol{A}_3 = \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 0 & 6 \end{pmatrix}$$

setzen und für die Elemente von L erhalten wir aus den 3 Eliminationsschritten die drei Elemente

$$\begin{aligned} l_{21} &= \frac{1}{(-1)} = -1 \\ l_{31} &= \frac{5}{(-1)} = -5 \\ l_{32} &= \frac{6}{(-2)} = -3 \end{aligned}$$

und damit

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -5 & -3 & 1 \end{pmatrix}$$

Die Probe ergibt wie gewünscht

$$\boldsymbol{LR} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -5 & -3 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 0 & 6 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -3 & -2 \\ 5 & 1 & 4 \end{pmatrix}$$

Satz 4.1: $\boldsymbol{LR}$-Zerlegung [1]

Zu jeder regulären $n \times n$ Matrix $\boldsymbol{A}$, für die der Gauss-Algorithmus ohne Zeilenvertauschung durchführbar ist, gibt es $n \times n$ Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ mit den folgenden Eigenschaften:

- $\boldsymbol{L}$ ist eine normierte untere Dreiecksmatrix (also mit $l_{ii} = 1$ für $i = 1, ..., n$)
- $\boldsymbol{R}$ ist eine obere Dreiecksmatrix mit $r_{ii} \neq 0$ für $i = 1, ..., n$
- $\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{R}$ ist die **$\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$**.

Aufwand: Die Berechnung der $\boldsymbol{LR}$-Zerlegung mit dem Gauss-Algorithmus benötigt $\frac{1}{3}(n^3 - n)$ Punktoperationen

Die LR-Zerlegung
Satz

- Bemerkungen:
  - Die direkte Lösung von $\boldsymbol{Ax} = \boldsymbol{b}$ durch die Berechnung der inversen $\boldsymbol{A}^{-1}$ ist nicht praktikabel, da dies die Lösung von n linearen Gleichungssystemen erfordern würde und damit erheblich aufwendiger wäre.
  - Ein mit der $\boldsymbol{LR}$-Zerlegung (in Englisch $\boldsymbol{LU}$-decomposition) verwandter Algorithmus wird auch teilweise angewendet als Benchmark für die Rechengeschwendigkeit.
  - Unter anderem ist die $\boldsymbol{LR}$-Zerlegung eine geschickte Variante, die Zwischenresultate des Gauss-Algorithmus zu speichern.

- Finden Sie für die Matrix A des linearen Gleichungssystems $\boldsymbol{Ax} = \boldsymbol{b}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & -1 \\ 4 & -2 & 6 \\ 3 & 1 & 0 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 9 \\ -4 \\ 9 \end{pmatrix}$$

die LR-Zerlegung. Benutzen Sie dafür die folgenden Operationen der Gauss-Elimination:

$$(\boldsymbol{A} \mid \boldsymbol{b}) = \begin{pmatrix} 1 & 2 & -1 & | & 9 \\ 4 & -2 & 6 & | & -4 \\ 3 & 1 & 0 & | & 9 \end{pmatrix} \xrightarrow{z_2 := z_2 - 4z_1} \begin{pmatrix} 1 & 2 & -1 & | & 9 \\ 0 & -10 & 10 & | & -40 \\ 3 & 1 & 0 & | & 9 \end{pmatrix}$$

$$\xrightarrow{z_3 := z_3 - 3z_1} \begin{pmatrix} 1 & 2 & -1 & | & 9 \\ 0 & -10 & 10 & | & -40 \\ 0 & -5 & 3 & | & -18 \end{pmatrix} \xrightarrow{z_3 := z_3 - 0.5z_2} \begin{pmatrix} 1 & 2 & -1 & | & 9 \\ 0 & -10 & 10 & | & -40 \\ 0 & 0 & -2 & | & 2 \end{pmatrix}$$

Berechnen Sie die Lösung $\boldsymbol{x}$ zuerst mittels Rückwärtseinsetzen direkt aus der obigen Dreiecksform und dem $\boldsymbol{b}$ Vektor. Lösen Sie anschliessend die beiden linearen Systeme $\boldsymbol{Ly} = \boldsymbol{b}$ und $\boldsymbol{Rx} = \boldsymbol{y}$ und vergleichen Sie.

- Während den Übungen:
  - Erweitern Sie ihr unter Aufgabe 4.3 erstelltes Programm zum Gauss-Algorithmus, so dass es gleichzeitig auch die $\boldsymbol{LR}$- Zerlegung von $\boldsymbol{A}$ berechnet.
  - Berechnen Sie damit die $\boldsymbol{LR}$-Zerlegung für die Matrixen aus Aufgabe 4.3.

Die LR-Zerlegung

Aufgabe 4.4: Lösung des ersten Teils

LR-Zerlegung mit Zeilenvertauschung

- Sind Zeilenvertauschungen nötig, so lässt sich in der Regel keine $\boldsymbol{LR}$–Zerlegung erhalten.
- Allerdings lässt sich die Vertauschung der i-ten und j-ten Zeile in $\boldsymbol{A}$ durch eine Multiplikation von links der Form

$$\boldsymbol{P}_k = \begin{pmatrix} 1 &&&&&&&&&& \\ & \ddots &&&&&&&&& \\ && 1 &&&&&&&& \\ &&& 0 &&&& 1 &&& \\ &&&& 1 &&&&&& \\ &&&&& \ddots &&&&& \\ &&&&&& 1 &&&& \\ &&& 1 &&&& 0 &&& \\ &&&&&&&& 1 && \\ &&&&&&&&& \ddots & \\ &&&&&&&&&& 1 \end{pmatrix}$$

erreichen, wobei die 0 gerade an der i-ten und j-ten Stelle auf der Diagonale steht (also $p_{ii} = p_{jj} = 0$).
- Die 1 erscheint dann in der i-ten Zeile und j-ten Spalte sowie der j-ten Zeile und i-ten Spalte ($p_{ij} = p_{ji} = 1$).

LR-Zerlegung mit Zeilenvertauschung

- Der ganzahlige Index $k = 1, 2, \ldots$ dient hier nur dazu, mehrere solcher Matrizen voneinander unterscheiden zu können, denn bei mehreren Zeilenvertauschungen können die dafür benötigten Matrizen $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_l$ zu einer einzigen Matrix $\boldsymbol{P} = \boldsymbol{P}_l \cdot \ldots \cdot \boldsymbol{P}_1$ aufmultipliziert werden (bei linksseitiger Multiplikation).
- Die Matrix $\boldsymbol{P}$ nennt sich die Permutationsmatrix, sie ist immer regulär und es gilt $\boldsymbol{P}^{-1} = \boldsymbol{P}$.

LR-Zerlegung mit Zeilenvertauschung

Beispiel 4.6.1

- Die Vertauschung der 2. und 4. Zeile bei der Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \text{ führt zu } \boldsymbol{A}^* = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \end{pmatrix},$$

welches sich auch durch die Muliplikation von links

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \end{pmatrix}$$

ausdrücken lässt, also $\boldsymbol{P}_1 \cdot \boldsymbol{A} = \boldsymbol{A}^*$.

LR-Zerlegung mit Zeilenvertauschung

Beispiel 4.6.1: Fortsetzung

- Die zusätzliche Vertauschung der 1. und 3. Zeile, also

$$\boldsymbol{A}^* = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \end{pmatrix} \text{ geht über in } \boldsymbol{A}^{**} = \begin{pmatrix} 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix},$$

  lässt sich darstellen durch die Multiplikation von links mit $\boldsymbol{P}_2 \cdot \boldsymbol{A}^* = \boldsymbol{A}^{**}$:

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \end{pmatrix} = \begin{pmatrix} 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}$$

- Die beiden Zeilenvertauschungen können zusammengefasst werden durch Multiplikation von $\boldsymbol{P} = \boldsymbol{P}_2 \cdot \boldsymbol{P}_1$, also $\boldsymbol{P} \cdot \boldsymbol{A} = \boldsymbol{A}^{**}$ wobei

$$\boldsymbol{P} = \boldsymbol{P}_2 \cdot \boldsymbol{P}_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

LR-Zerlegung mit Zeilenvertauschung

- Mit dieser Permuationsmatrix erhält man dann als $\boldsymbol{RL}$-Zerlegung
$$\boldsymbol{PA} = \boldsymbol{LR}$$
und das lineare Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ lässt sich schreiben als $\boldsymbol{PAx} = \boldsymbol{Pb}$ bzw. $\boldsymbol{LRx} = \boldsymbol{Pb}$ und in den zwei Schritten lösen:

$$\begin{aligned} \boldsymbol{Ly} &= \boldsymbol{Pb} \Rightarrow \boldsymbol{y} = \ldots \\ \boldsymbol{Rx} &= \boldsymbol{y} \Rightarrow \boldsymbol{x} = \ldots \end{aligned}$$

LR-Zerlegung mit Zeilenvertauschung

- Wird die Zerlegung mittels Gauss-Algorithmus mit Spaltenpivotisierung (vgl. Kap. 4.4) durchgeführt, muss man also bei jeder Zeilenvertauschung die dazugehörige Permuationsmatrix berechnen und erhält schliesslich $\boldsymbol{L}$, $\boldsymbol{R}$ und $\boldsymbol{P}$.
- Dieses Verfahren nennt man auch $\boldsymbol{LR}$-Zerlegung mit **Spalten- bzw. Kolonnenmaximumstrategie**.

LR-Zerlegung mit Zeilenvertauschung
Beispiel 4.6.2

- Gegeben ist das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 3 & 9 & 12 & 12 \\ -2 & -5 & 7 & 2 \\ 6 & 12 & 18 & 6 \\ 3 & 7 & 38 & 14 \end{pmatrix}, \ \boldsymbol{b} = \begin{pmatrix} 51 \\ 2 \\ 54 \\ 79 \end{pmatrix}$$

Berechen Sie die $\boldsymbol{LR}-$Zerlegung von $\boldsymbol{A}$ mit Spaltenmaximumstrategie und bestimmen Sie anhand von $\boldsymbol{L}$, $\boldsymbol{R}$ und $\boldsymbol{P}$ die Lösung $\boldsymbol{x}$.

LR-Zerlegung mit Zeilenvertauschung

Beispiel 4.6.2: Lösung

1. Zeilenvertauschung von 1. Zeile mit 3. Zeile in $\boldsymbol{A}$, so dass mit $a_{31} = 6$ das betragsmässig grösste Element auf der Diagonale liegt. $\boldsymbol{P}_1$ bildet diese Zeilenvertauschung ab. Da die Elemente in $\boldsymbol{L}$ unterhalb der Diagonalen noch unbestimmt sind, hat eine Zeilenvertauschung noch keinen Einfluss.

$$\boldsymbol{A}^* = \begin{pmatrix} 6 & 12 & 18 & 6 \\ -2 & -5 & 7 & 2 \\ 3 & 9 & 12 & 12 \\ 3 & 7 & 38 & 14 \end{pmatrix}, \quad \boldsymbol{P_1} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ ? & 1 & 0 & 0 \\ ? & ? & 1 & 0 \\ ? & ? & ? & 1 \end{pmatrix},$$

$$i = 1, j = 2 \Rightarrow z_2 \equiv z_2 - \frac{(-2)}{6} z_1 \Rightarrow \boldsymbol{A}_1^* = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & -1 & 13 & 4 \\ 3 & 9 & 12 & 12 \\ 3 & 7 & 38 & 14 \end{pmatrix}, \quad \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 \\ ? & ? & 1 & 0 \\ ? & ? & ? & 1 \end{pmatrix}$$

$$i = 1, j = 3 \Rightarrow z_3 \equiv z_3 - \frac{3}{6} z_1 \Rightarrow \boldsymbol{A}_2^* = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & -1 & 13 & 4 \\ 0 & 3 & 3 & 9 \\ 3 & 7 & 38 & 14 \end{pmatrix}, \quad \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 \\ \frac{1}{2} & ? & 1 & 0 \\ ? & ? & ? & 1 \end{pmatrix}$$

$$i = 1, j = 3 \Rightarrow z_4 \equiv z_4 - \frac{3}{6} z_1 \Rightarrow \boldsymbol{A}_3^* = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & -1 & 13 & 4 \\ 0 & 3 & 3 & 9 \\ 0 & 1 & 29 & 11 \end{pmatrix}, \quad \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 \\ \frac{1}{2} & ? & 1 & 0 \\ \frac{1}{2} & ? & ? & 1 \end{pmatrix}$$

LR-Zerlegung mit Zeilenvertauschung

Beispiel 4.6.2: Lösung

2. Zeilenvertauschung von 2. Zeile mit 3. Zeile in $\boldsymbol{A}_3^*$, auch für die Elemente in der ersten Spalte von $\boldsymbol{L}$.

$$\boldsymbol{A}^{**} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & -1 & 13 & 4 \\ 0 & 1 & 29 & 11 \end{pmatrix}, \; \boldsymbol{P_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \; \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{3} & ? & 1 & 0 \\ \frac{1}{2} & ? & ? & 1 \end{pmatrix}$$

$$i=2, j=3 \Rightarrow z_3 \equiv z_3 - \frac{(-1)}{3} z_2 \Rightarrow \boldsymbol{A}_1^{**} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 14 & 7 \\ 0 & 1 & 29 & 11 \end{pmatrix}, \; \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & 1 & 0 \\ \frac{1}{2} & ? & ? & 1 \end{pmatrix}$$

$$i=2, j=4 \Rightarrow z_4 \equiv z_4 - \frac{1}{3} z_2 \Rightarrow \boldsymbol{A}_2^{**} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 14 & 7 \\ 0 & 0 & 28 & 8 \end{pmatrix}, \; \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & 1 & 0 \\ \frac{1}{2} & \frac{1}{3} & ? & 1 \end{pmatrix}$$

3. Zeilenvertauschung von 4. Zeile mit 3. Zeile in $\boldsymbol{A}_2^{**}$, auch für die Elemente in der ersten und zweiten Spalte von $\boldsymbol{L}$:

$$\boldsymbol{A}^{***} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 28 & 8 \\ 0 & 0 & 14 & 7 \end{pmatrix}, \quad \boldsymbol{P_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & ? & 1 \end{pmatrix}$$

$$i = 3, j = 4 \Rightarrow z_4 \equiv z_4 - \frac{2}{14} z_3 \Rightarrow \boldsymbol{A}_3^{**} = \boldsymbol{R} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 28 & 8 \\ 0 & 0 & 0 & 3 \end{pmatrix}, \quad \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{2} & 1 \end{pmatrix}$$

4. Als Resultat erhalten wir damit:

$$\boldsymbol{R} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 28 & 8 \\ 0 & 0 & 0 & 3 \end{pmatrix}, \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{2} & 1 \end{pmatrix}, \boldsymbol{P} = \boldsymbol{P_3} \cdot \boldsymbol{P_2} \cdot \boldsymbol{P_1} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

und es gilt wie gewünscht

$$\boldsymbol{LR} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 3 & 9 & 12 & 12 \\ 3 & 7 & 38 & 14 \\ -2 & -5 & 7 & 2 \end{pmatrix} = \boldsymbol{PA}$$

LR-Zerlegung mit Zeilenvertauschung
Beispiel 4.6.2: Lösung

Für die zu lösenden Gleichungssysteme

$$\begin{aligned} \boldsymbol{Ly} &= \boldsymbol{Pb} \\ \boldsymbol{Rx} &= \boldsymbol{y} \end{aligned}$$

erhalten wir den Vektor $\boldsymbol{y}$ durch Vorwärtseinsetzen:

$$\boldsymbol{Ly} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \boldsymbol{Pb} = \begin{pmatrix} 54 \\ 51 \\ 79 \\ 2 \end{pmatrix} \Rightarrow \boldsymbol{y} = \begin{pmatrix} 54 \\ 24 \\ 44 \\ 6 \end{pmatrix}$$

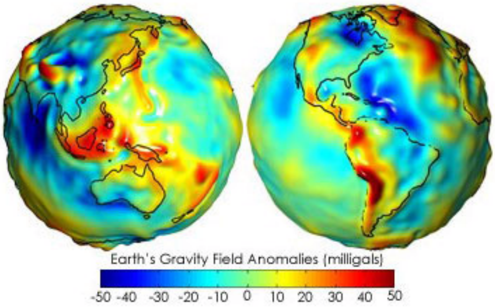
und die eigentlich gesuchte Lösung $\boldsymbol{x}$ durch Rückwärtseinsetzen:

$$\boldsymbol{R}\boldsymbol{x} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 28 & 8 \\ 0 & 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \boldsymbol{y} = \begin{pmatrix} 54 \\ 24 \\ 44 \\ 6 \end{pmatrix} \Rightarrow \boldsymbol{x} = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 2 \end{pmatrix}$$

Die Cholesky-Zerlegung

- Im folgenden lernen wir ein weiteres Verfahren zur Dreieckszerlegung von Matrizen kennen, einen Spezialfall der LR-Zerlegung.
- Dieses Verfahren ist nach seinem Entdecker André-Louis Cholesky (1875 -1918) benannt, einem französischen Mathematiker. Er entwickelte es für Anwendungen in der Geodäsie.
- Die Geodäsie ist die Wissenschaft von der Ausmessung und Abbildung der Erdoberfläche und umfasst die Bestimmung der geometrischen Figur der Erde (Geoid), ihres Schwerefeldes und der Orientierung im Weltraum.

Die Cholesky-Zerlegung



These "gravity anomaly" maps show where models of the Earth's gravity field based on GRACE data differ from a simplified mathematical model that assumes the Earth is perfectly smooth and featureless. From NASA's GRACE Mission (Grvity Recovery and Climate Experiment), see http://earthobservatory.nasa.gov/Features/GRACE/.

Die Cholesky-Zerlegung

- Die Cholesky-Zerlegung funktioniert nicht für allgemeine Matrizen sondern nur für *symmetrische, positiv definite* Matrizen.
- Falls anwendbar, ist es aber etwa um einen Faktor zwei effizienter als die allgemeine $\boldsymbol{LR}$-Zerlegung.

Definition 4.2: Symmetrische / positiv definite Matrizen [1]

- Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ heisst symmetrisch, falls $\boldsymbol{A}^T = \boldsymbol{A}$ gilt ($\boldsymbol{A}^T$ ist die transponierte Matrix).
- Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ heisst positiv definit, falls für alle $\boldsymbol{x} \in \mathbb{R}^n$, $x \neq 0$ gilt $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > 0$.

Die Cholesky-Zerlegung

- Natürlich ist diese Bedingung eine gewisse Einschränkung, allerdings werde wir sehen, dass sie für viele Anwendungen erfüllt ist, so führt z.B. das Ausgleichsproblem i.A. auf ein Gleichungssystem mit symmetrischer und positiv definiter Matrix $\boldsymbol{A}$.
- Für solche Matrizen kann nun gezeigt werden, dass immer eine $\boldsymbol{LR}$-Zerlegung existiert, wobei $\boldsymbol{R}$ hier zusätzlich die besonders einfache Form $\boldsymbol{R} = \boldsymbol{L}^T$ bzw. $\boldsymbol{R}^T = \boldsymbol{L}$ besitzt. Die Matrix $\boldsymbol{A}$ können wir dann schreiben als $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$ bzw. als $\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$ und es gilt der folgende Satz.

Die Cholesky-Zerlegung

Satz 4.2: Cholesky Zerlegung [1]
Für jede positiv definite $n \times n$ Matrix $\boldsymbol{A}$ gibt es genau eine rechts-obere Dreiecksmatrix $\boldsymbol{R}$ mit $r_{ii} > 0$ für $i = 1, \ldots, n$ und $\boldsymbol{A} = \boldsymbol{R}^T \boldsymbol{R}$. Diese Zerlegung heisst **Cholesky-Zerlegung** von $\boldsymbol{A}$.

- Die Berechnung der Cholesky-Zerlegung geschieht anhand des folgenden Algorithmus, der uns die Koeffizienten r_{ij} der oberen Dreiecksmatrix $\boldsymbol{R}$ berechnet und gleichzeitig überprüft, ob $\boldsymbol{A}$ positiv definit ist:

Cholesky-Algorithmus [1]:
Gegeben sei eine symmetrische $n \times n$ Matrix $\boldsymbol{A}$. Für $i = 1, \dots, n$ berechne:

- $S = a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2$ (für $i = 1$ ist also $S = a_{ii}$)
- falls $S \leq 0$, dann ist $\boldsymbol{A}$ nicht positiv definit $\rightarrow$stopp.
- falls $S > 0$:
  - $r_{ii} = \sqrt{S}$
  - für $j = i+1, \dots, n:\ r_{ij} = \frac{1}{r_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj} \right)$

- Es soll geprüft werden, ob die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 7 \\ 3 & 7 & 26 \end{pmatrix}$$

positiv definit ist und wenn ja, die entsprechende Cholesky-Zerlegung berechnet werden.

Die Cholesky-Zerlegung
Beispiel 4.7: Lösung

- $\boldsymbol{A}$ ist symmetrisch, also können wir den Cholesky-Algorithmus anwenden:
- $i = 1 \Rightarrow S = a_{11} - \sum_{k=1}^{0} r_{k1}^2 = a_{11} = 1 > 0 \Rightarrow r_{11} = \sqrt{1} = 1$
  - $j = 2 \Rightarrow r_{12} = \frac{1}{r_{11}}(a_{12} - \sum_{k=1}^{0} r_{k1} r_{k2}) = a_{12} = 2$
    - $j = 3 \Rightarrow r_{13} = \frac{1}{r_{11}}(a_{13} - \sum_{k=1}^{0} r_{k1} r_{k3}) = a_{13} = 3$
  - $i = 2 \Rightarrow S = a_{22} - \sum_{k=1}^{1} r_{k2}^2 = a_{22} - r_{12}^2 = 5 - 4 = 1 > 0 \Rightarrow$ $r_{22} = \sqrt{1} = 1$
    - $j = 3 \Rightarrow r_{23} = \frac{1}{r_{22}}(a_{23} - \sum_{k=1}^{1} r_{k2} r_{k3}) = \frac{1}{r_{22}}(a_{23} - r_{12} r_{13}) =$ $\frac{1}{1}(7 - 2 \cdot 3) = 1$

- $i = 3 \Rightarrow S = a_{33} - \sum_{k=1}^{2} r_{k3}^2 = a_{33} - r_{13}^2 - r_{23}^2 =$
  $26 - 3^2 - 1^2 = 16 > 0 \Rightarrow r_{33} = \sqrt{16} = 4$
- Da bei keinem der Diagonalelemente eine Wurzel aus einer negativen Zahl gezogen werden sollte, konnte der Algorithmus bis zum Ende durchgeführt werden, d.h die Matrix ist positiv definit und wir haben

$$\boldsymbol{R} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{pmatrix}$$

$$\boldsymbol{A} = \boldsymbol{R}^T \boldsymbol{R} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 1 & 4 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 7 \\ 3 & 7 & 26 \end{pmatrix}$$

- Bemerkung: für eine 3×3 Matrix $\boldsymbol{A} = (a_{ij})$ können die obigen Schritte zusammengefasst werden als

$$\boldsymbol{R} = \begin{pmatrix} \sqrt{a_{11}} & \frac{a_{12}}{r_{11}} & \frac{a_{12}}{r_{11}} \\ 0 & \sqrt{a_{22} - r_{12}^2} & \frac{1}{r_{22}}(a_{23} - r_{12} r_{13}) \\ 0 & 0 & \sqrt{a_{33} - r_{13}^2 - r_{23}^2} \end{pmatrix}$$

wobei $r_{11} = \sqrt{a_{11}}$, $r_{12} = \frac{a_{12}}{r_{11}}$ etc.

- Berechnen Sie die Cholesky-Zerlegung für die folgende Matrix:

$$\boldsymbol{A}_1 = \begin{pmatrix} 4 & -2 & 6 \\ -2 & 5 & -1 \\ 6 & -1 & 26 \end{pmatrix}$$

- Optional: Implementieren Sie den Cholesky-Algorithmus in MATLAB und testen Sie Ihn an den folgenden Matrizen:

$$\boldsymbol{A}_2 = \begin{pmatrix} 9 & 12 & 6 \\ 12 & 25 & 23 \\ 6 & 23 & 78 \end{pmatrix}, \boldsymbol{A}_3 = \begin{pmatrix} 4 & -8 & 6 \\ -8 & 17 & -8 \\ 6 & -8 & 34 \end{pmatrix},$$

$$\boldsymbol{A}_4 = \begin{pmatrix} 36 & -24 & 18 \\ -24 & 17 & -8 \\ 18 & -8 & 25 \end{pmatrix}, \boldsymbol{A}_5 = \begin{pmatrix} 64 & -40 & 16 \\ -40 & 29 & -4 \\ 16 & -4 & 62 \end{pmatrix},$$

$$\boldsymbol{A}_6 = \begin{pmatrix} 9 & -21 & 6 \\ -21 & 49 & -14 \\ 6 & -14 & 29 \end{pmatrix}$$

$$A_7 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 & 26 & 29 \\ 3 & 8 & 14 & 20 & 26 & 32 & 38 & 44 & 50 & 56 \\ 4 & 11 & 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 5 & 14 & 26 & 40 & 55 & 70 & 85 & 100 & 115 & 130 \\ 6 & 17 & 32 & 50 & 70 & 91 & 112 & 133 & 154 & 175 \\ 7 & 20 & 38 & 60 & 85 & 112 & 140 & 168 & 196 & 224 \\ 8 & 23 & 44 & 70 & 100 & 133 & 168 & 204 & 240 & 276 \\ 9 & 26 & 50 & 80 & 115 & 154 & 196 & 240 & 285 & 330 \\ 10 & 29 & 56 & 90 & 130 & 175 & 224 & 276 & 330 & 385 \end{pmatrix}$$

Fehlerrechnung und Aufwandabschätzung

Fehlerrechnung bei lin. Gleichungssystemen

- Wie bereits in Kapitel 2 ausgeführt, können Computer nicht alle reellen Zahlen darstellen, weswegen alle Zahlen intern gerundet werden.
- Aufgrund von diesen Rundungsfehlern aber auch wegen Eingabe– bzw. Messfehlern in den vorliegenden Daten oder Fehlern aus vorhergehenden numerischen Rechnungen, wird durch einen Algorithmus üblicherweise nicht die exakte Lösung $\boldsymbol{x}$ des linearen Gleichungssystems

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

berechnet, sondern eine Näherungslösung $\widetilde{\boldsymbol{x}}$. Um dies formal zu fassen, führt man ein "benachbartes" oder "gestörtes" Gleichungssystems

$$\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}} = \boldsymbol{b} + \triangle\boldsymbol{b}$$

ein, für das $\widetilde{\boldsymbol{x}}$ gerade die exakte Lösung ist.

- Dabei ist $\triangle \boldsymbol{b}$ das *Residuum* oder der *Defekt* der Näherungslösung $\widetilde{\boldsymbol{x}}$.
- Den Vektor $\triangle \boldsymbol{x} = \widetilde{\boldsymbol{x}} - \boldsymbol{x}$ nennen wir den Fehler der Näherungslösung $\widetilde{\boldsymbol{x}}$.
- Da Rundung und andere Fehlerquellen i.A. nur kleine Fehler bewirken, ist es gerechtfertigt anzunehmen, dass der noch zu definierende 'Betrag' $\| \triangle \boldsymbol{b} \|$ 'klein' ist.
- Das Ziel dieses Abschnittes ist es nun, aus der Grösse des Residuum $\| \triangle \boldsymbol{b} \|$ auf die Grösse des Fehlers $\| \widetilde{\boldsymbol{x}} \|$ zu schließen.
- Insbesondere wollen wir untersuchen, wie sensibel die Grösse $\| \widetilde{\boldsymbol{x}} \|$ von $\| \triangle \boldsymbol{b} \|$ abhängt, d.h. ob kleine Residuen $\| \triangle \boldsymbol{b} \|$ große Fehler $\| \widetilde{\boldsymbol{x}} \|$ hervorrufen können. Dafür brauchen wir das Konzept der Norm.

Definition 4.3: Vektornorm [1]

Eine Abbildung $\|.\| : \mathbb{R}^n \longrightarrow \mathbb{R}$ heisst Vektornorm, wenn die folgenden Bedingungen für alle $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ erfüllt sind:

- $\|\boldsymbol{x}\| \geq 0$ und $\|\boldsymbol{x}\| = 0 \Longleftrightarrow x = 0$
- $\|\lambda \boldsymbol{x}\| = |\lambda| \, \|\boldsymbol{x}\|$
- $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ “Dreiecksungleichung”

Definition 4.4: Vektornormen / Matrixnormen [1]

- Für Vektoren $\boldsymbol{x} = (x_1, x_2, ..., x_n)^{\mathrm{T}} \in \mathbb{R}^n$ gibt es die folgenden Vektornormen:

$$\text{1-Norm, Summennorm} \quad : \quad \| \boldsymbol{x} \|_1 = \sum_{i=1}^{n} | x_i |$$

$$\text{2-Norm, euklidische Norm} \quad : \quad \| \boldsymbol{x} \|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

$$\infty\text{-Norm, Maximumnorm} \quad : \quad \| \boldsymbol{x} \|_\infty = \max_{i=1,...,n} | x_i |$$

Definition 4.4: Vektornormen / Matrixnormen [1]

- Für eine $n \times n$ Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ sind mit den Vektornormen die folgenden Matrixnormen verbunden, welche die Eigenschaften der Definition 4.3 ebenfalls erfüllen:

$$
\begin{aligned}
\text{1-Norm, Spaltensummennorm} &: \quad \| \boldsymbol{A} \|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} | a_{ij} | \\
\text{2-Norm, Spektralnorm} &: \quad \| \boldsymbol{A} \|_2 = \sqrt{\rho(\boldsymbol{A}^T \boldsymbol{A})} \\
\infty\text{-Norm, Zeilensummennorm} &: \quad \| \boldsymbol{A} \|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} | a_{ij} |
\end{aligned}
$$

Fehlerrechnung bei lin. Gleichungssystemen

- Bemerkungen:
  - Die euklidische Norm entspricht dem herkömmlichen Verständnis der Länge eines Vektors, die beiden anderen Vektornormen sind aber im Zusammenhang mit Matrixoperationen einfacher berechenbar.
  - Insbesondere ist die 2-Norm für Matrizen hier nur zur Vollständigkeit aufgeführt, wir werden im Folgenden also nur die 1- und $\infty-$ Norm verwenden.

- Berechnen Sie die 1-, 2 -, und $\infty-$ Norm des Vektors $\begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}$ sowie die 1- und $\infty-$ Norm von $\begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & -2 \\ 7 & -3 & 5 \end{pmatrix}$.

$$\left\|\begin{pmatrix}-1\\2\\3\end{pmatrix}\right\|_1 = 1+2+3=6, \quad \left\|\begin{pmatrix}-1\\2\\3\end{pmatrix}\right\|_2 = \sqrt{1+2^2+3^2} = \sqrt{14},$$

$$\left\|\begin{pmatrix}-1\\2\\3\end{pmatrix}\right\|_\infty = \max\{1,\,2,\,3\} = 3$$

$$\left\|\begin{pmatrix}1&2&3\\3&4&-2\\7&-3&5\end{pmatrix}\right\|_1 = \max\{1+3+7,\,2+4+3,\,3+2+5\} = 11$$

$$\left\|\begin{pmatrix}1&2&3\\3&4&-2\\7&-3&5\end{pmatrix}\right\|_\infty = \max\{1+2+3,\,3+4+2,\,7+3+5\} = 15.$$

- Für die Fehlerabschätzung von $\widetilde{\boldsymbol{x}}$ und $\widetilde{\boldsymbol{b}}$ gilt der folgende Satz:

Satz 4.3: Abschätzung für fehlerbehaftete Vektoren

- Sei $\| \, . \, \|$eine Norm, $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ eine reguläre $n \times n$ Matrix und $\boldsymbol{x}$, $\widetilde{\boldsymbol{x}}$, $\boldsymbol{b}$, $\widetilde{\boldsymbol{b}} \in \mathbb{R}^n$ mit $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ und $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$. Dann gilt für den absoluten und den relativen Fehler in $\boldsymbol{x}$:
  - $\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \| \leq \| \boldsymbol{A}^{-1} \| \cdot \| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|$
  - $\frac{\|\boldsymbol{x}-\widetilde{\boldsymbol{x}}\|}{\|\boldsymbol{x}\|} \leq \| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \| \cdot \frac{\|\boldsymbol{b}-\widetilde{\boldsymbol{b}}\|}{\|\boldsymbol{b}\|}$ falls $\| \boldsymbol{b} \| \neq 0$
- Die Zahl cond$(\boldsymbol{A}) = \| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \|$ nennt man Konditionszahl der Matrix $\boldsymbol{A}$ bzgl. der verwendeten Norm.

- Für Matrizen, deren Kondition $\text{cond}(\boldsymbol{A})$ groß ist, können sich kleine Fehler im Vektor $\boldsymbol{b}$ (bzw. Rundungsfehler im Verfahren) zu großen Fehlern im Ergebnis $\boldsymbol{x}$ verstärken. Man spricht in diesem Fall von schlecht konditionierten Matrizen.

- Untersuchen Sie die Fehlerfortpflanzung im linearen Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 2 & 4 \\ 4 & 8.1 \end{pmatrix}, \; \boldsymbol{b} = \begin{pmatrix} 1 \\ 1.5 \end{pmatrix}$$

für den Fall, dass die rechte Seite von $\widetilde{\boldsymbol{b}}$ in jeder Komponente um maximal 0.1 von $\boldsymbol{b}$ abweicht.

- Wir betrachten das System $\boldsymbol{A}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{b}}$, wobei $\tilde{\boldsymbol{b}}$ maximal um 0.1 von jeder Komponente von $\boldsymbol{b}$ abweicht.
- Zuerst müssen wir eine der möglichen Norm wählen. Hierfür ist die $\infty-$ Norm besonders geeignet, da wir schreiben können $\| \tilde{\boldsymbol{b}} - \boldsymbol{b} \|_\infty \leq 0.1$.
- Zusätzlich haben wir $\| \boldsymbol{A} \|_\infty = 12.1$ und mit $\boldsymbol{A}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix} = \frac{1}{2 \cdot 8.1 - 4 \cdot 4} \begin{pmatrix} 8.1 & -4 \\ -4 & 2 \end{pmatrix}$ erhalten wir $\| \boldsymbol{A}^{-1} \|_\infty = \frac{12.1}{0.2} = 60.5$.

- Für die Konditionszahl $cond(A)$ erhalten wir in der $\infty-$ Norm $cond(\boldsymbol{A})_\infty = \| \boldsymbol{A} \|_\infty \| \boldsymbol{A}^{-1} \|_\infty = 12.1 \cdot 60.5 = 732.05.$ Mit dem obigen Satz gilt also

$$\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty \leq \| \boldsymbol{A}^{-1} \|_\infty \cdot \| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty \leq 60.5 \cdot 0.1 = 6.05$$

$$\frac{\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty}{\| \boldsymbol{x} \|_\infty} \leq cond(\boldsymbol{A})_\infty \frac{\| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty}{\| \boldsymbol{b} \|_\infty} \leq 732 \cdot \frac{0.1}{1.5} = 48.8$$

- n Wie ist dies nun zu interpretieren? Die Lösung $\widetilde{\boldsymbol{x}}$ des gestörten Systems $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$ wird also von der Lösung x des exakten Systems $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ in jeder Komponente um maximal 6.05 abweichen (absoluter Fehler), und der relative Fehler wird maximal 48.8 betragen. Testen wir das an einem konkreten Beispiel.

- Betrachten Sie obiges Beispiel und nehmen sie für die gestörte rechte Seite $\widetilde{\boldsymbol{b}} = \begin{pmatrix} 0.9 \\ 1.6 \end{pmatrix}$.
- Berechnen sie die Lösungen von $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ und $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$.
- Berechnen Sie anschliessend den absoluten Fehler $\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty$ und den relativen Fehler $\frac{\|\boldsymbol{x}-\widetilde{\boldsymbol{x}}\|_\infty}{\|\boldsymbol{x}\|_\infty}$.
- Vergleichen Sie mit $\| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty$ und $\frac{\|\boldsymbol{b}-\widetilde{\boldsymbol{b}}\|_\infty}{\|\boldsymbol{b}\|_\infty}$.

- Wir erhalten $\boldsymbol{x} = \begin{pmatrix} 10.5 \\ -5 \end{pmatrix}$ und $\widetilde{\boldsymbol{x}} = \begin{pmatrix} 4.45 \\ -2 \end{pmatrix}$.
- Mit $\| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty = 0.1$ und $\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty = 6.05$ sehen wir, dass der absolute Fehler um den maximal möglichen Faktor 60.5 verstärkt worden ist.
- Mit $\frac{\|\boldsymbol{b}-\widetilde{\boldsymbol{b}}\|_\infty}{\|\boldsymbol{b}\|_\infty} = \frac{0.1}{1.5} = 0.0667$ und $\frac{\|\boldsymbol{x}-\widetilde{\boldsymbol{x}}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \frac{6.05}{10.5} = 0.5762$ wurde der relative Fehler um einen Faktor 8.6 verstärkt, weniger als der maximal mögliche Faktor von 732.

Fehlerrechnung bei lin. Gleichungssystemen

- Wir waren bisher davon ausgegangen, dass die Matrix A selbst exakt ist.
- Wie verhält sich die Fehlerabschätzung nun unter der Annahme, dass auch noch A fehlerbehaftet ist, wir es also mit einem Gleichungssystem

$$\widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$$

zu tun haben? Dafür gilt die folgende Fehlerabschätzung:

Satz 4.4: Abschätzung für fehlerbehaftete Matrix [1]

Sei $\| \, . \, \|$ eine Norm, $A, \tilde{A} \in \mathbb{R}^{n \times n}$ reguläre $n \times n$ Matrizen und x, $\widetilde{x}$, b, $\widetilde{b} \in \mathbb{R}^n$ mit $Ax = b$ und $\widetilde{A}\widetilde{x} = \widetilde{b}$. Falls

$$\mathrm{cond}(A) \cdot \frac{\| A - \widetilde{A} \|}{\| A \|} < 1$$

dann gilt:

$$\frac{\| x - \widetilde{x} \|}{\| x \|} \leq \frac{\mathrm{cond}(A)}{1 - \mathrm{cond}(A) \cdot \frac{\|A - \widetilde{A}\|}{\|A\|}} \cdot \left(\frac{\| A - \widetilde{A} \|}{\| A \|} + \frac{\| b - \widetilde{b} \|}{\| b \|} \right)$$

Fehlerrechnung bei lin. Gleichungssystemen

- Für den Fall, dass $\boldsymbol{A}$ exakt gegeben ist, gilt $\frac{\|\boldsymbol{A}-\widetilde{\boldsymbol{A}}\|}{\|\boldsymbol{A}\|} = 0$ und der relative Fehler für $\boldsymbol{x}$ aus Satz 4.4 reduziert sich auf den relativen Fehler in Satz 4.3.

- Nehmen Sie noch einmal das Beispiel 4.9 und untersuchen Sie die Fehlerfortpflanzung unter der zusätzlichen Annahme, dass die Matrix $\boldsymbol{A}$ um maximal 0.003 elementweise gestört ist.

- Wir hatten bereits die folgenden Grössen berechnet

$$\| \boldsymbol{A} \|_\infty = 12.1,\ \mathrm{cond}(\boldsymbol{A}) = 732.05,\ \|\boldsymbol{b}\|_\infty = 1.5,\ \| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty \leq 0.1$$

- Wenn nun jedes Element von $\boldsymbol{A}$ um maximal 0.003 gestört wird, summiert sich diese Störung in der $\infty-$Norm auf und wir erhalten $\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\|_\infty \leq 0.006$ und damit

$$\mathrm{cond}(\boldsymbol{A}) \cdot \frac{\| \boldsymbol{A} - \widetilde{\boldsymbol{A}} \|_\infty}{\| \boldsymbol{A} \|_\infty} \leq 0.363 < 1.$$

- Wir können also die Abschätzung aus Satz 4.4 anwenden und erhalten

$$\frac{\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty}{\| \boldsymbol{x} \|_\infty} \leq \frac{732.05}{1 - 0.363} \left(\frac{0.006}{12.1} + \frac{0.1}{1.5} \right) \leq 77.2$$

Aufwandabschätzung

- Ein wichtiger Aspekt bei der Analyse numerischer Verfahren ist die Abschätzung, wieviel Aufwand diese Verfahren in der Regel benötigen, um zu dem gewünschten Ergebnis zu kommen.
- Dies hängt entscheidend von der Leistungsfähigkeit des verwendeten Computers ab. Deshalb wird nicht direkt die Zeit abgeschätzt, sondern vielmehr die Anzahl der Rechenoperationen, die ein Algorithmus benötigt.
- Da hierbei die Gleitkommaoperationen, also Addition, Multiplikation etc. von reellen Zahlen, die mit Abstand zeitintensivsten Operationen sind, beschränkt man sich in der Analyse üblicherweise auf diese.

Aufwandabschätzung

- Bisher haben wir nur direkte Verfahren angeschaut, welche nach einer endlichen Anzahl von Rechenschritten die 'exakte' Lösung liefern.
- Natürlich hängt hierbei die Anzahl Schritte von dere Dimension n der Matrix $\boldsymbol{A}$ ab.
- Es genügt also, die Anzahl der dafür benötigten Gleitkommaoperationen in Abhängigkeit von n zu bestimmen. Dafür benötigt man die Gleichungen

- Wie viele Gleitkommaoperationen benötigt das Rückwärtseinsetzen gemäss der Gleichung

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}, \qquad i = n, n-1, \ldots, 1$$

- Verwenden Sie dazu

$$\sum_{i=1}^{n} i = \frac{(n+1)n}{2} \text{ und } \sum_{i=1}^{n} i^2 = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n.$$

- Anzahl Multiplikationen und Divisionen: für $i = n$ haben wir eine Division, für $i = n-1$ haben wir eine Muliplikation und eine Division, etc. Für $i = 1$ schliesslich haben wir $n-1$ Multiplikationen und eine Division. Das ergibt

$$1+2+3+\ldots+n = \sum_{i=1}^{n} i = \frac{(n+1)n}{2}.$$

- Anzahl Additionen und Subtraktionen: für $i = n$ haben wir keine, für $i = n-1$ haben wir eine Subtraktion, etc., das ergibt dann

$$0+1+2+\ldots+n-1 = \sum_{i=1}^{n-1} i = \frac{(n-1+1)(n-1)}{2} = \frac{n(n-1)}{2}.$$

Für die Summe beider Operationstypen erhalten wir also

$$\frac{n^2}{2} + \frac{n}{2} + \frac{n^2}{2} - \frac{n}{2} = n^2.$$

Aufwandabschätzung

- Für die Gauss-Elimination erhält man nach einer ähnlichen Betrachtung die Anzahl Gleitkommaoperationen (ohne Pivotisierung) zu

$$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{13}{6}n.$$

Die Anzahl Operationen für die LR-Zerlegung ist identisch, wenn sie mit der Gauss-Elimination durchgeführt wurde. Für das Choleski-Verfahren erhält man (ohne Beweis)

$$\frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n.$$

Aufwandabschätzung

- Für die vollständige Lösung eines linearen Gleichungssystems müssen nun die Operationen für Rückwärtseinsetzen (Gauss) bzw. Rückwärts- *und* Vorwärtseinsetzen (LR-Zerlegung und Cholesky) noch addiert werden. Für die Gauss-Elimination erhalten wir dann

$$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{13}{6}n + n^2 = \frac{2}{3}n^3 + \frac{5}{2}n^2 - \frac{13}{6}n,$$

für die LR-Zerlegung

$$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{13}{6}n + 2n^2 = \frac{2}{3}n^3 + \frac{7}{2}n^2 - \frac{13}{6}n,$$

und für die Cholesky-Zerlegung

$$\frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n + 2n^2 = \frac{1}{3}n^3 + \frac{5}{2}n^2 + \frac{1}{6}n$$

Aufwandabschätzung

- Berücksichtigt man, dass für große n die "n^3–Terme" dominant werden, so ergibt sich, dass das Choleski–Verfahren etwa doppelt so schnell wie die Gauß–Elimination ist.
- Im Vergleich dazu müssen bei der Cramerschen Regel $n+1$ Determinanten und n Quotienten bestimmt werden, was für jede Determinante mit der Regel von Leibniz $(n-1)\cdot n!$ Multiplikationen und $n!-1$ Additionen beinhaltet. Das ergibt also

$$(n+1)((n-1)\cdot n! + n! - 1) + n = n(n+1)! - 1$$

Punktoperationen.

Aufwandabschätzung

- Um einen Eindruck von den tatsächlichen Rechenzeiten zu bekommen, nehmen wir an, dass wir einen handelsüblichen PC verwenden, der mit einer 3GHz Quad-Core CPU ausgestattet ist mit einer tatsächlichen Leistung von 30 GFLOPS (FLOPS = floating point operations per second), d.h. mit $30 \cdot 10^9$ Gleitkommaoperationen pro Sekunde (zum Vergleich: die Zuse Z3 schaffte mit einer Taktrate von 5.3 Hz rund 1 FLOPS) .
- Nehmen wir weiterhin an, dass wir Implementierungen der obigen Algorithmen haben, die diese Leistung optimal ausnutzen. Dann ergeben sich für $n \times n$ Gleichungssysteme die folgenden (ungefähren) Rechenzeiten

Aufwandabschätzung

n	Gauss	LR-Zerlegung	Cholesky	Cramer
10^1	30 ns	30 ns	15 ns	0.1 s
10^2	23 μs	23 μs	12 μs	10^{143}y
10^3	22 ms	22 ms	11 ms	–
10^4	22 s	22 s	11 s	–
10^5	6 h	6 h	3 h	–
10^6	257 d	257 d	129 yd	–
10^7	704 y	704 y	352 y	–

Aufwandabschätzung

- Wie man sieht, wächst die benötigte Zeit für den Gauss-Algorithmus, die $LR-$Zerlegung und dem Cholesky-Algorithmus um einen Faktor $10^3 = 1000$, wenn n um einen Faktor 10 erhöht wird.
- Für die Cramersche Regel benötigt man für $n = 10$ 'erst' 0.1 Sekunde, für $n = 20$ bereits rund 1000 Jahre, für $n = 25$ bereits 10 Mia. Jahre und für $n = 100$ läppische 10^{143} Jahre. Ein eindrückliches Beispiel, wie schnell die Fakultät wächst.
- Ab $n > 10^5$ kommt aber auch für die anderen Algorithmen die Wartezeit in einen Bereich, der kaum mehr akzeptabel ist. Im nächsten Abschnitt werden wir deshalb die iterativen Verfahren kennen lernen, die zwar nicht mehr die 'exakte' Lösung berechnen, dafür aber wesentlich schneller sind.

Aufwandabschätzung

- Zum Abschluss dieses Abschnitts wollen wir aber noch ein etwas gröberes Konzept zur Aufwandabschätzung betrachten,
- Man interessiert sich dabei nur für eine Abschätzung bei grossen Dimensionen, d.h. wie der Aufwand sich asysmptotisch für $n \to \infty$ verhält.

Definition 4.5: Ordnung [3]

- Ein Algorithmus hat die Ordnung $O(n^q)$, wenn $q > 0$ die minimale Zahl ist, für die es eine Konstante C > 0 gibt, so dass der Algorithmus für alle $n \in N$ weniger als Cn^q Operationen benötigt.

- Die Zahl q ist einfach abzulesen. Sie entspricht der höchsten auftretenden Potzen von n. Es folgt, dass Vor- und Rückwärtseinsetzten sind von der Ordnung $O(n^2)$, das Gauss-Vefahren, die $\boldsymbol{LR}$- und Cholesky-Zerlegung von der Ordnung $O(n^3)$.

Iterative Verfahren

Iterative Verfahren

- Wir haben bereits gesehen, dass die bisher betrachteten direkten Verfahren die Ordnung $O(n^3)$ besitzen.
- Für große Gleichungssysteme mit mehreren 100'000 Unbekannten, die in der Praxis durchaus auftreten, führt dies wie oben gesehen zu unakzeptabel hohen Rechenzeiten.
- Eine Klasse von Verfahren, die eine niedrigere Ordnung hat, sind die iterativen Verfahren.
- Allerdings zahlt man für den geringeren Aufwand einen Preis: Man kann bei diesen Verfahren nicht mehr erwarten, eine (bis auf Rundungsfehler) exakte Lösung zu erhalten, sondern muss von vornherein eine gewisse Ungenauigkeit im Ergebnis in Kauf nehmen.

Iterative Verfahren

- Das Grundprinzip iterativer Verfahren funktioniert dabei wie folgt: Ausgehend von einem Startvektor $\boldsymbol{x}^{(0)}$ berechnet man mittels einer Rechenvorschrift $F : \mathbb{R}^n \to \mathbb{R}^n$ iterativ eine Folge von Vektoren $\boldsymbol{x}^{(k+1)} = F(\boldsymbol{x}^{(k)})$, $k = 0, 1, 2, \dots$, die für $k \to \infty$ gegen die Lösung $\boldsymbol{x}$ des Gleichungssystems $\boldsymbol{Ax} = \boldsymbol{b}$ konvergieren.
- Wenn die gewünschte Genauigkeit erreicht ist, wird die Iteration abgebrochen und der letzte Wert $x^{(k)}$ als Näherung des Ergebnisses verwendet.

- Bemerkung zur Notation: ein hochgestellter Index in Klammern $\boldsymbol{x}^{(k)}$ bezeichnet einen Vektor aus $\mathbb{R}^n$ nach der $k-$ten Iteration. Die Elemente des Vektors $\boldsymbol{x}^{(k)}$ werden wie üblich mit einem tiefgestellten Index bezeichnet, z.B. ist also $\boldsymbol{x}_i^{(k)}$ das $i-$te Element des Vektors, bzw.

$$\boldsymbol{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_{n-1}^{(k)} \\ x_n^{(k)} \end{pmatrix}$$

Iterative Verfahren

- Wir wollen versuchen, dieses Problem als Fixpunktiteration zu behandeln.
- Wir hatten in Kapitel 3 gesehen, dass die allgemeine Fixpunktgleichung die Form $F(x) = x$ hat, d.h. wir wollen die ürsprüngliche Gleichung $\boldsymbol{Ax} = \boldsymbol{b}$ in eine ähnliche Form bringen. Dies gelingt uns, wenn wir die Matrix $\boldsymbol{A}$ Zerlegen können in eine Form

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{R}$$

zerlegen können, wobei $\boldsymbol{L}$ eine untere Dreiecksmatrix sein soll (mit $l_{ii} = 0$) , $\boldsymbol{D}$ eine Diagonalmatrix und $\boldsymbol{R}$ eine obere Dreiecksmatrix (mit $r_{ii} = 0$).

Die einfachste Form ist

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{R}$$

$$= \underbrace{\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn-1} & 0 \end{pmatrix}}_{=:\boldsymbol{L}} + \underbrace{\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix}}_{=:\boldsymbol{D}}$$

$$+ \underbrace{\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}}_{=:\boldsymbol{R}}$$

- Achtung: diese Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ hier sind nicht die gleichen wie die $\boldsymbol{LR}-$Zerlegung!

Das Jacobi-Verfahren (Gesamtschritt-Verfahren)

- Wir können mit obiger Zerlegung dann die folgende Fixpunktiteration, die auch als Jacobi- oder Gesamtschritt-Verfahren bekannt ist, durchführen:

Definition 4.6: Jacobi- bzw. Gesamtschrittverfahren

- Zu lösen sei $\boldsymbol{Ax} = \boldsymbol{b}$. Die Matrix $\boldsymbol{A} = (a_{ij})$ sei zerlegt in der Form $\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{R}$. Dann heisst die Fixpunktiteration

$$
\begin{aligned}
\boldsymbol{D}\boldsymbol{x}^{(k+1)} &= -(\boldsymbol{L}+\boldsymbol{R})\boldsymbol{x}^{(k)} + \boldsymbol{b} \text{ bzw.} \\
\boldsymbol{x}^{(k+1)} &= -\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{R})\boldsymbol{x}^{(k)} + \boldsymbol{D}^{-1}\boldsymbol{b}
\end{aligned}
$$

Gesamtschrittverfahren oder **Jacobi-Verfahren**.

Das Jacobi-Verfahren

- Hinweis:, auf der Diagonalen von $\boldsymbol{D}^{-1}$ stehen einfach die Kehrwerte der Diagonalen von $\boldsymbol{D}$:

$$\boldsymbol{D} = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix}$$

$$\Rightarrow \boldsymbol{D}^{-1} = \begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{22}} & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{a_{33}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \frac{1}{a_{nn}} \end{pmatrix}$$

- Die Herleitung des Jacobi-Verfahrens folgt direkt aus der Zerlegung von A:

$$\begin{aligned} \boldsymbol{Ax} &= \boldsymbol{b} \\ (\boldsymbol{L}+\boldsymbol{D}+\boldsymbol{R})\boldsymbol{x} &= \boldsymbol{b} \\ (\boldsymbol{L}+\boldsymbol{R})x+\boldsymbol{Dx} &= \boldsymbol{b} \\ \boldsymbol{Dx} &= -(\boldsymbol{L}+\boldsymbol{R})x+\boldsymbol{b} \\ \boldsymbol{x} &= -\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{R})\boldsymbol{x}+\boldsymbol{D}^{-1}\boldsymbol{b} \equiv F(\boldsymbol{x}) \end{aligned}$$

womit wir die Fixpunktgleichung bereits aufgestellt haben.

- Wenden Sie das Jacobi-Verfahren auf das folgende System an:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \text{ mit } \boldsymbol{A} = \begin{pmatrix} 4 & -1 & 1 \\ -2 & 5 & 1 \\ 1 & -2 & 5 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}$$

Das Jacobi-Verfahren
Beispiel 4.13: Lösung

Lösung: Mit den Bezeichnungen aus (3.7) haben wir:

$$\boldsymbol{L} = \begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & -2 & 0 \end{pmatrix}, \quad \boldsymbol{D} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}, \quad \boldsymbol{R} = \begin{pmatrix} 0 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Die Iteration lautet somit:

$$\begin{aligned} \boldsymbol{x}^{(n+1)} &= -\boldsymbol{D}^{-1}((\boldsymbol{L}+\boldsymbol{R})\,\boldsymbol{x}^{(n)} - \boldsymbol{b}) \\ &= -\begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{pmatrix} \left(\begin{pmatrix} 0 & -1 & 1 \\ -2 & 0 & 1 \\ 1 & -2 & 0 \end{pmatrix} \boldsymbol{x}^{(n)} - \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix} \right) \\ &= \begin{pmatrix} 0 & 0.25 & -0.25 \\ 0.4 & 0 & -0.2 \\ -0.2 & 0.4 & 0 \end{pmatrix} \boldsymbol{x}^{(n)} + \begin{pmatrix} 1.25 \\ 2.2 \\ 2.4 \end{pmatrix} \end{aligned}$$

Wir wählen als Startvektor den Nullvektor und erhalten:

i	0	1	2	3	4	5
$\boldsymbol{x}^{(i)}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1.25 \\ 2.2 \\ 2.4 \end{pmatrix}$	$\begin{pmatrix} 1.2 \\ 2.22 \\ 3.03 \end{pmatrix}$	$\begin{pmatrix} 1.0475 \\ 2.074 \\ 3.048 \end{pmatrix}$	$\begin{pmatrix} 1.0065 \\ 2.0094 \\ 3.0201 \end{pmatrix}$	$\begin{pmatrix} 0.997325 \\ 1.99858 \\ 3.00246 \end{pmatrix}$

Es sieht so aus, als konvergiere diese Folge gegen $(1,2,3)^{\mathrm{T}}$, was übrigens die Lösung des System $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ darstellt. ∎

- Üblicherweise wird die Iteration nicht mit Matrixmultiplikation durchgeführt sondern für jede Komponente des Vektors $\boldsymbol{x}$ separat, für obiges Beispiel wäre das also

$$\begin{aligned} x_1^{(k+1)} &= 0.25x_2^{(k)} - 0.25x_3^{(k)} + 1.25 \\ x_2^{(k+1)} &= 0.4x_1^{(k)} - 0.2x_3^{(k)} + 2.2 \\ x_3^{(k+1)} &= -0.2x_1^{(k)} + 0.4x_2^{(k)} + 2.4 \end{aligned}$$

und in der allgemeinen Form (zur einfacheren Implementation) können wir das schreiben als

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

Das Gauss-Seidel-Verfahren (Einzelschritt)

- Wenn man nun davon ausgeht, dass nach der k-ten Iteration der Vektor $\boldsymbol{x}^{(k+1)}$ komponentenweise näher an der Lösung liegt als der Vektor vom vorherigen Iterationsschritt $\boldsymbol{x}^{(k)}$, dann ist es im obigen Beispiel vermutlich genauer, die gerade berechnete Komponente $x_1^{(k+1)}$ aus der ersten Gleichung in die noch zu berechnende Komponente $x_2^{(k+1)}$ in die zweite Gleichung einzusetzen.
- Analog setzt man anschliessend die Komponenten $x_1^{(k+1)}$ und $x_2^{(k+1)}$ in die dritte Gleichung ein, um $x_3^{(k+1)}$ zu erhalten.

Das Gauss-Seidel-Verfahren

- Dies führt dann auf die Iteration

$$\begin{aligned} x_1^{(k+1)} &= 0.25 x_2^{(k)} - 0.25 x_3^{(k)} + 1.25 \\ x_2^{(k+1)} &= 0.4 x_1^{(k+1)} - 0.2 x_3^{(k)} + 2.2 \\ x_3^{(k+1)} &= -0.2 x_1^{(k+1)} + 0.4 x_2^{(k+1)} + 2.4 \end{aligned}$$

welche man in Matrix-Form schreiben kann als

$$x^{(k+1)} = \begin{pmatrix} 0 & 0.25 & -0.25 \\ 0 & 0 & -0.2 \\ 0 & 0 & 0 \end{pmatrix} x^{(k)} + \begin{pmatrix} 0 & 0 & 0 \\ 0.4 & 0 & 0 \\ -0.2 & 0.4 & 0 \end{pmatrix} x^{(k+1)} + \begin{pmatrix} 1.25 \\ 2.2 \\ 2.4 \end{pmatrix}.$$

Das Gauss-Seidel-Verfahren

- Mit unseren Matrizen $\boldsymbol{L}$, $\boldsymbol{D}$, und $\boldsymbol{R}$ wird das zu

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{D}^{-1}\left(\boldsymbol{b} - \boldsymbol{L}\boldsymbol{x}^{(k+1)} - \boldsymbol{R}\boldsymbol{x}^{(k)}\right)$$

oder in der allgemeinen Form

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

Das Gauss-Seidel-Verfahren

- Umformung, so dass alle Terme mit $\boldsymbol{x}^{(k+1)}$ auf der linken Seite erscheinen, führt zum sogenannten Gauss-Seidel-Verfahren oder auch Einzelschrittverfahren.

Definition 4.7: Gauss-Seidel bzw. Einzelschrittverfahren [1]

- Zu lösen sei $\boldsymbol{Ax} = \boldsymbol{b}$. Die Matrix $\boldsymbol{A} = (a_{ij})$ sei wieder zerlegt in der Form $\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{R}$.
  Dann heisst die Fixpunktiteration

$$\begin{aligned} (\boldsymbol{D}+\boldsymbol{L})\boldsymbol{x}^{(k+1)} &= -\boldsymbol{R}\boldsymbol{x}^{(k)} + \boldsymbol{b} \text{ bzw.} \\ \boldsymbol{x}^{(k+1)} &= -(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{R}\boldsymbol{x}^{(k)} + (\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{b} \end{aligned}$$

  Einzelschrittverfahren oder **Gauss-Seidel-Verfahren**.

- Wenden Sie das Gauss-Seidel-Verfahren auf das System aus Beispiel 4.13 an, also:

$$\boldsymbol{Ax} = \boldsymbol{b} \text{ mit } \boldsymbol{A} = \begin{pmatrix} 4 & -1 & 1 \\ -2 & 5 & 1 \\ 1 & -2 & 5 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}$$

Das Gauss-Seidel-Verfahren

Beispiel 4.14: Lösung

- Wir verwenden die allgemeine Gleichung für die Komponenten und erhalten

$$
\begin{aligned}
x_1^{(k+1)} &= \frac{1}{a_{11}}\left(b_1 - \sum_{j=1}^{0} a_{1j}x_j^{(k+1)} - \sum_{j=2}^{3} a_{1j}x_j^{(k)}\right)\\
&= \frac{1}{4}(5-(-1)x_2^{(k)} - 1x_3^{(k)})\\
&= 1.25 + 0.25x_2^{(k)} - 0.25x_3^{(k)}\\
x_2^{(k+1)} &= \frac{1}{a_{22}}\left(b_2 - \sum_{j=1}^{1} a_{2j}x_j^{(k+1)} - \sum_{j=3}^{3} a_{2j}x_j^{(k)}\right)\\
&= \frac{1}{5}(11-(-2)x_1^{(k+1)} - 1x_3^{(k)})\\
&= 2.2 + 0.4x_1^{(k+1)} - 0.2x_3^{(k)}
\end{aligned}
$$

$$
\begin{aligned}
x_3^{(k+1)} &= \frac{1}{a_{33}}\left(b_3 - \sum_{j=1}^{2} a_{3j}x_j^{(k+1)} - \sum_{j=4}^{3} a_{3j}x_j^{(k)}\right) \\
&= \frac{1}{5}(12 - 1x_1^{(k+1)} - (-2)x_2^{(k+1)}) \\
&= 2.4 - 0.2x_1^{(k+1)} + 0.4x_2^{(k+1)}
\end{aligned}
$$

- Wir wählen den Null-Vektor als Startvektor, also $\boldsymbol{x}^{(0)} = (0,0,0)^T$ und setzen oben ein. Damit erhalten wir $x_1^{(1)} = 1.25$, $x_2^{(1)} = 2.2 + 0.4 \cdot 1.25 = 2.7$, $x_3^{(1)} = 2.4 - 0.2 \cdot 1.25 + 0.4 \cdot 2.7 = 3.23$, also $\boldsymbol{x}^{(1)} = (1.25, 2.7, 3.23)^T$. Weiteres Einsetzen liefert:

i	0	1	2	3	4
$x^{(i)}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1.25\\2.7\\3.23\end{pmatrix}$	$\begin{pmatrix}1.1175\\2.001\\2.9769\end{pmatrix}$	$\begin{pmatrix}1.006025\\2.00703\\3.001607\end{pmatrix}$	$\begin{pmatrix}1.00135575\\2.0002209\\2.99981721\end{pmatrix}$

- Also können wir annehmen, dass diese Folge gegen $(1,2,3)^T$ konvergiert, und zwar schneller als mit dem Jacobi-Verfahren.

Das Gauss-Seidel-Verfahren
Beispiel 4.14: Lösung

- Natürlich hätten wir die Iterationsgleichungen auch etwas übersichtlicher aus dem Zusammenhang

$$(\boldsymbol{D}+\boldsymbol{L})\boldsymbol{x}^{(k+1)} = -\boldsymbol{R}\boldsymbol{x}^{(k)} + \boldsymbol{b}$$

herleiten können:

$$\begin{pmatrix} 4 & 0 & 0 \\ -2 & 5 & 0 \\ 1 & -2 & 5 \end{pmatrix} \boldsymbol{x}^{(k+1)} = -\begin{pmatrix} 0 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{x}^{(k)} + \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}.$$

- Komponentenweise folgt (ohne Berechnung der Inversen $(D+L)^{-1}$)

$$\begin{aligned} 4x_1^{(k+1)} &= -(-x_2^{(k)} + x_3^{(k)}) + 5 \\ -2x_1^{(k+1)} + 5x_2^{(k+1)} &= -x_3^{(k)} + 11 \\ x_1^{(k+1)} - 2x_2^{(k+1)} + 5x_3^{(k+1)} &= 12 \end{aligned}$$

Das Gauss-Seidel-Verfahren

Beispiel 4.14: Lösung

- Oder wie erwartet:

$$\begin{aligned}
x_1^{(k+1)} &= \frac{1}{4}x_2^{(k)} - \frac{1}{4}x_3^{(k)}) + \frac{5}{4} \\
x_2^{(k+1)} &= \frac{2}{5}x_1^{(k+1)} - \frac{1}{5}x_3^{(k)} + \frac{11}{5} \\
x_3^{(k+1)} &= -\frac{1}{5}x_1^{(k+1)} + \frac{2}{5}x_2^{(k+1)} + \frac{12}{5}
\end{aligned}$$

Konvergenz

- Wir haben bereits Kriterien bezüglich der Konvergenz von Fixpunktiterationen kennengelernt.
- Diese können direkt auf die vektoriellen Fixpunktgleichungen des Jacobi- und des Gauss-Seidel-Verfahrens angewandt werden, es muss dabei nur eine Norm statt des Betragszeichens verwendet werden.

Definition 4.8: anziehender / abstossender Fixpunkt [1]

- Gegeben sei eine Fixpunktiteration

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{B}\boldsymbol{x}^{(n)} + \boldsymbol{b} =: F(\boldsymbol{x}^{(n)})$$

wobei $\boldsymbol{B}$ eine $n \times n$ Matrix ist und $\boldsymbol{b} \in \mathbb{R}^n$.Weiter sei $\| . \|$eine der in Kap. 4.6.1 eingeführten Normen und $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ erfülle $\overline{\boldsymbol{x}} = \boldsymbol{B}\overline{\boldsymbol{x}} + \boldsymbol{b} = F(\overline{\boldsymbol{x}})$. Dann heisst

  - $\overline{\boldsymbol{x}}$ anziehender Fixpunkt, falls $\| \boldsymbol{B} \| < 1$ gilt
  - $\overline{\boldsymbol{x}}$ abstossender Fixpunkt, falls $\| \boldsymbol{B} \| > 1$ gilt.

Konvergenz

- Unter Anwendung des Banachschen Fixpunktsatzes haben wir dann die folgende Fehlerabschätzung zur Verfügung:

Satz 4.5: Abschätzungen [1]

- Gegeben sei wie in obiger Definition eine Fixpunktiteration

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{B}\boldsymbol{x}^{(n)} + \boldsymbol{b} =: F(\boldsymbol{x}^{(n)})$$

und $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ sei ein bezüglich der Norm $\|\cdot\|$ anziehender Fixpunkt. Dann konvergiert die Fixpunktiteration für alle Startvektoren $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ gegen $\bar{\boldsymbol{x}}$ und es gelten die Abschätzungen

$$\begin{aligned}
\| \boldsymbol{x}^{(n)} - \bar{\boldsymbol{x}} \| &\leq \frac{\| \boldsymbol{B} \|^n}{1 - \| \boldsymbol{B} \|} \| \boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)} \| \quad \text{a-priori Abschätzung} \\
\| \boldsymbol{x}^{(n)} - \bar{\boldsymbol{x}} \| &\leq \frac{\| \boldsymbol{B} \|}{1 - \| \boldsymbol{B} \|} \| \boldsymbol{x}^{(n)} - \boldsymbol{x}^{(n-1)} \| \quad \text{a-posteriori Abschätzung}
\end{aligned}$$

Konvergenz

Bemerkung: Der Vergleich mit den Definitionen für das Gesamt- und Einzelschrittverfahren liefert die Matrix $\boldsymbol{B}$:

- für das Gesamtschrittverfahren (Jacobi) ist

$$\boldsymbol{B} = -\boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{R}),$$

- für das Einzelschrittverfahren (Gauss-Seidel) ist

$$\boldsymbol{B} = -(\boldsymbol{D} + \boldsymbol{L})^{-1}\boldsymbol{R}.$$

Ausserdem gilt mit der folgenden Definition:

Definition 4.8: Diagonaldominanz [1]

- $\boldsymbol{A}$ ist eine **diagonaldominante Matrix**, falls eines der beiden folgenden Kriterien gilt:
  - für alle $i = 1, ..., n$: $\mid a_{ii} \mid > \sum_{j=1, j \neq i}^{n} \mid a_{i,j} \mid$ (Zeilensummenkriterium)
  - für alle $j = 1, ..., n:$ $\mid a_{jj} \mid > \sum_{i=1, i \neq j}^{n} \mid a_{i,j} \mid$ (Spaltensummenkriterium)

Satz 4.6: Konvergenz [1]

- Falls $\boldsymbol{A}$ diagonaldominant ist, konvergiert das Gesamtschrittverfahren (Jacobi) und auch das Einzelschrittverfahren (Gauss-Seidel) für $\boldsymbol{Ax} = \boldsymbol{b}$.

Konvergenz

- Die Bedingung $\| \boldsymbol{B} \| < 1$ für einen anziehenden Fixpunkt $\bar{\boldsymbol{x}}$ impliziert, dass $\boldsymbol{A}$ diagonaldominant ist.
- Diagonaldominanz ist nur ein hinreichendes Kriterium. Es gibt durchaus nicht diagonaldominante Matrizen, für die die Verfahren trotzdem konvergieren kann. Ein notwendiges und hinreichendes Kriterium ist, dass der Spektralradius $\rho(\boldsymbol{B}) < 1$.

Konvergenz
Aufgabe 4.6

- Prüfen Sie, ob das Jacobi-Verfahren in Beispiel 4.13 konvergiert. Schätzen Sie den Fehler des Vektors $\boldsymbol{x}^{(5)}$ ab. Wie viele Schritte sollten Sie rechnen, damit der berechnete Näherungsvektor in jeder Komponente um max. 10^{-4} von der exakten Lösung $\boldsymbol{x} = (1, 2, 3)^T$ abweicht? Vergleichen Sie Ihre Fehlerabschätzung mit den wirklichen Gegebenheiten.
- Bearbeiten Sie die Aufgabenstellung nochmals, aber mit dem Gauss-Seidel-Verfahren und dem Näherungsvektor $\boldsymbol{x}^{(4)}$ aus Beispiel 4.14.

Konvergenz
Aufgabe 4.6: Lösung Teil 1

Konvergenz

Aufgabe 4.6: Lösung Teil 1