4. Großübung

Lösung linearer Gleichungssysteme

Gesucht $x_1, x_2, x_3, x_4 \in \mathbb{R}$, sodass

$$\begin{aligned} 2x_1 - x_2 + 3x_3 + 2x_4 &= -5 \\ -6x_1 - 3x_2 - 7x - 2x_4 &= 5 \\ 4x_1 + 4x_2 + 5x_3 - 5x_4 &= 13 \\ 8x_1 + 2x_2 + 12x_3 + 2x_4 &= -8 \end{aligned}$$

gilt.

$$\iff \underbrace{\begin{pmatrix} 2 & -1 & 3 & 2 \\ -6 & -3 & -7 & -2 \\ 4 & 4 & 5 & -5 \\ 8 & 2 & 12 & 2 \end{pmatrix}}_{=:A} \cdot \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}}_{=:x} = \underbrace{\begin{pmatrix} -5 \\ 5 \\ 13 \\ -8 \end{pmatrix}}_{=:b} \iff A \cdot x = b$$

mit $A \in \mathbb{R}^{4\times 4},\ x \in \mathbb{R}^4,\ b \in \mathbb{R}^4$

Wiederholung HöMa (nur online):

$A \in \mathbb{R}^{n\times n}$ regulär äquivalent zu:

- $det(A) \neq 0$
- $rang(A) = n$; Rang: Maximale Anzahl linear unabhäniger Spalten bzw. Zeilen.
- Es existiert $A^{-1} \in \mathbb{R}^{n\times n}$, s.t. $A \cdot A^{-1} = A^{-1} \cdot A = I$
- $A \cdot x = b$ hat eine eindeutige Lösung für alle $b \in \mathbb{R}^n$
- $A \cdot x = 0$ hat nur die Lösung $x = 0$

Gauß-Elimination:

Ziel:

- Erzeuge LGS mit rechter oberer Dreiecksmatrix, um dann Rückwärtseinsetzen durchzuführen.

Idee:

- Bilde Linearkombinationen von Zeilen. Manipuliere eine Zeile durch Subtraktion einer skalierten anderen Zeile.

Regeln (Konvention) und Hinweise:

1. Skaliere nicht die eigene Zeile während des Gauss-Algorithmus.
   (Evtl. aber vorher als Zeilenäquilibrierung.)
2. Erzeuge Nullen unterhalb der Diagonalen. Zunächst in der ersten Spalte, dann in der zweiten, etc..
3. Zum Erzeugen der Nullen in der $i$ten (z.B. ersten) Spalte wird die zu manipulierende Zeile ($>i$) nur mit der ($i$ten) erste Zeile kombiniert.
4. Bei allen Operationen auf der Matrix müssen auch die rechten Seiten mit angepasst werden
   $\Rightarrow$ Oft wird LGS dann als *erweiterte Matrix* aufgeschrieben.
5. Jeder einzelne Schritt (und die Summe dieser Schritte) kann als Matrix-Multiplikation aufgefasst werden $A \rightarrow \mathcal{M} \cdot A$, wobei $\mathcal{M} \in \mathbb{R}^{n\times n}$ und $\mathcal{M}$ eine normierte untere Dreiecksmatrix ist.
6. Jeder einzelne Rückwärtsschritt (und die Summe dieser Schritte) kann als Matrix-Multiplikation aufgefasst werden $A \rightarrow \mathcal{L} \cdot A$, wobei $\mathcal{L} \in \mathbb{R}^{n\times n}$ und $\mathcal{L}$ eine normierte untere Dreiecksmatrix ist.
7. Wenn mehr als nur ein Gleichungssystem mit der Matrix A gelöst werden soll, ist es sinnvoll die Koeffizienten ($l_{i,j}$) zu speichern (s. später).

$$\left(\begin{array}{cccc|c} 2 & -1 & 3 & 2 & -5 \\ -6 & -3 & -7 & -2 & 5 \\ 4 & 4 & 5 & -5 & 13 \\ 8 & 2 & 12 & 2 & -8 \end{array}\right) \begin{array}{l} \leftarrow I \\ \\ \\ \\ \end{array}$$

Betrachte Spalte I und Zeilen II, III, IV:

$$\begin{array}{r} \\ -l_{2,1}\cdot I \rightarrow \\ -l_{3,1}\cdot I \rightarrow \\ -l_{4,1}\cdot I \rightarrow \end{array} \left(\begin{array}{cccc|c} 2 & -1 & 3 & 2 & -5 \\ 0 & -6 & 2 & 4 & -10 \\ 0 & 6 & -1 & -9 & 23 \\ 0 & 6 & 0 & -6 & 12 \end{array}\right) \begin{array}{l} \\ \leftarrow II \\ \\ \\ \end{array} \quad \begin{array}{l} \\ 1\ Div. + 3\ Mult. \\ 1\ Div. + 3\ Mult. \\ 1\ Div. + 3\ Mult. \end{array} + \left[\begin{array}{l} 1\ Mult. \\ 1\ Mult. \\ 1\ Mult. \end{array}\right]$$

$$l_{2,1} = \frac{a_{2,1}}{a_{1,1}} = \frac{-6}{2} = -3$$
$$l_{3,1} = \frac{a_{3,1}}{a_{1,1}} = \frac{4}{2} = 2$$
$$l_{4,1} = \frac{a_{4,1}}{a_{1,1}} = \frac{8}{2} = 4$$

Aufwand erster Schritt Matrix: 3 Div. + 9 Mult. [+ 3 Mult.]

$$\begin{array}{r} \\ \\ -l_{3,2}\cdot II \rightarrow \\ -l_{4,2}\cdot II \rightarrow \end{array} \left(\begin{array}{cccc|c} 2 & -1 & 3 & 2 & -5 \\ 0 & -6 & 2 & 4 & -10 \\ 0 & 0 & 1 & -5 & 13 \\ 0 & 0 & 2 & -2 & 2 \end{array}\right) \begin{array}{l} \\ \\ \leftarrow III \quad 1\ Div. + 2\ Mult. \\ \quad 1\ Div. + 2\ Mult. \end{array} + \left[\begin{array}{r} \\ \\ 1\ Mult. \\ 1\ Mult. \end{array}\right]$$

Aufwand zweiter Schritt Matrix: 2 Div. + 4 Mult. [+ 2 Mult.]

$$l_{3,2} = \frac{a_{3,2}}{a_{2,2}} = \frac{6}{-6} = -1$$
$$l_{4,2} = \frac{a_{4,2}}{a_{2,2}} = \frac{6}{-6} = -1$$

$$\begin{array}{r} \\ \\ \\ -l_{4,3}\cdot III \rightarrow \end{array} \left(\begin{array}{cccc|c} 2 & -1 & 3 & 2 & -5 \\ 0 & -6 & 2 & 4 & -10 \\ 0 & 0 & 1 & -5 & 13 \\ 0 & 0 & 0 & 8 & -24 \end{array}\right)$$

Aufwand zweiter Schritt Matrix: 1 Div. + 1 Mult. [+ 1 Mult.]

$$l_{4,3} = \frac{a_{4,3}}{a_{3,3}} = \frac{2}{1} = 2$$

Aufwand j-ter Schritt Matrix: $(n-j)$ Div. + $(n-j)^2$ Mult. [+ $(n-j)$ Mult.] (für $j \in \{1, .., n-1\}$) Gesamtaufwand Gauss-Algorithmus: $\approx \frac{1}{3}n^3$ [+ $\frac{1}{2}n^2$]

Lösung durch Rückwärtseinsetzen:

$$x_4 = \frac{-24}{8} = -3$$
$$x_3 = \frac{13 - (-5x_4)}{1} = -2$$
$$x_2 = \frac{-10 - (4x_4 + 2x_3)}{-6} = -1$$
$$x_1 = \frac{-5 - (2x_4 + 3x_3 - x_2)}{2} = 3$$

Aufwand Rückwärtseinsetzen: $\frac{1}{2}n^2$

Mögliche Variante:

Andere rechte Seite, aber gleiche Matrix:

$$Ax = b\ , b = \begin{pmatrix} -11 \\ 3 \\ 16 \\ -14 \end{pmatrix}$$

- Eine Wiederholung der Gauß-Elimination wäre, abgesehen von der rechten Seite, die selbe Rechnung (nur die rechte Spalte in der *erweiterten Matrix* wäre anders).
- Stattdessen: Die Koeffizienten $l_{i,j}$ definieren eine Zerlegung, die verwendet werden kann, denn es gilt $A = LR$ mit:

$$L := \begin{pmatrix} 1 & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 \\ 2 & -1 & 1 & 0 \\ 4 & -1 & 2 & 1 \end{pmatrix}, \ R := \begin{pmatrix} 2 & -1 & 3 & 2 \\ 0 & -6 & 2 & 4 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 8 \end{pmatrix}$$

Beispiel: Klausuraufgabe VF-3, H2010

Die Musterlösung dieser Aufgabe befindet kann von der NumaMB Webseite heruntergeladen werden.

LR-Zerlegung

Die hier beschriebene Gauß-Elimination bewirkt, sofern durchführbar, eine Faktorisierung der Matrix A mit $A = LR$

- L ist eine normierte untere Dreiecksmatrix

$$L = (l_{i,j})_{i,j=1}^{n} \ mit$$

$$l_{i,i} = 1,\ i = 1, .., n$$

- R ist eine obere Dreiecksmatrix

Die Lösung von $Ax = b$ ergibt sich über die Lösung zweier Dreickssysteme:

$$\begin{aligned} Ax = b \Leftrightarrow L\underbrace{Rx}_{y} &= b \\ \Leftrightarrow Ly &= b \text{ (Vorwärtseinsetzen)} \\ Rx &= y \text{ (Rückwärtseinsetzen)} \end{aligned}$$

$\underline{Ly = b}$ Vorwärtseinsetzen

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & -11 \\ -3 & 1 & 0 & 0 & 3 \\ 2 & -1 & 1 & 0 & 16 \\ 4 & -1 & 2 & 1 & -14 \end{array}\right)$$

$$\begin{aligned} &\Rightarrow y_1 = -11 \\ &\Rightarrow y_2 = 3 - (-3y_1) = -30 \\ &\Rightarrow y_3 = 16 - (2y_1 - y_2) = 8 \\ &\Rightarrow y_4 = -14 - (4y_1 - y_2 + 2y_3) = -16 \end{aligned}$$

$\underline{Rx = y}$ Rückwärtseinsetzen

$$\left(\begin{array}{cccc|c} 2 & -1 & 3 & 2 & -11 \\ 0 & -6 & 2 & 4 & -30 \\ 0 & 0 & 1 & -5 & 8 \\ 0 & 0 & 0 & 8 & -16 \end{array}\right)$$

$$\begin{aligned} &\Rightarrow x_4 = -2 \\ &\Rightarrow x_3 = \frac{8 - (-5x_4)}{1} = -2 \\ &\Rightarrow x_2 = \frac{-30 - (4x_4 + 2x_3)}{-6} = 3 \\ &\Rightarrow x_1 = \frac{-11 - (2x_4 + 3x_3 - x_2)}{2} = 1 \\ &\Rightarrow x = \begin{pmatrix} 1 \\ 3 \\ -2 \\ -2 \end{pmatrix} \end{aligned}$$

Fazit

LR-Zerlegung = 'Gauß-Elimination ohne rechte Seite'.

Im Vergleich zur Gauß-Elimination entsteht kein Zusatzaufwand, da nur die $l_{i,j}$ behalten werden müssen. Das Verfahren ist sehr effizient bei mehreren rechten Seiten, die am Anfang noch nicht alle bekannt sind. Beispiel hierfür ist das (vereinfachtes) Newton-Verfahen.

Zeilenäquilibrierung (Skalierung) und Spaltenpivotisierung

- Problem $\rightarrow$ Kondition $\underset{\text{Ersatzproblem}}{\overset{\text{verbessert durch}}{\longleftarrow}}$ Zeilenäquilibrierung
- Algorithmus $\rightarrow$ Stabilität $\leftarrow$ Spaltenpivotisierung

Pivotisierung

$Problem: \; a_{j,j} \approx 0$

$$\left(\begin{array}{cc|c} 10^{-6} & 1 & 1 \\ 2 & 3 & 5 \end{array}\right) \to \left(\begin{array}{cc|c} 10^{-6} & 1 & 1 \\ \frac{2}{10^{-6}} & 3 - \frac{2}{10^{-6}} & 5 - \frac{2}{10^{-6}} \end{array}\right)$$

z.B. $m = 5$ Stellen
$\Rightarrow x_2 = 1, \; x_1 = 0$ aber exakt gerechnet erhalten wir: $x_2 = 1 - 10^{-6}, \; x_1 = 1 + 1.5 \cdot 10^{-6}$
$\Rightarrow \frac{\triangle x_1}{x}$ ist sehr groß
$\Rightarrow$ Algorithmus ist instabil.

<u>Vertausche Zeilen:</u>

$$\left(\begin{array}{cc|c} 2 & 3 & 5 \\ 10^{-6} & 1 & 1 \end{array}\right) \to \left(\begin{array}{cc|c} 2 & 3 & 5 \\ \frac{10^{-6}}{2} & 1 - \frac{10^{-6}}{2} \cdot 3 & 1 - \frac{10^{-6}}{2} \cdot 5 \end{array}\right)$$

z.B. $m = 5$ Stellen
$\Rightarrow x_2 = 1, \; x_1 = 1 \Rightarrow$ Fehler im Bereich 10^{-6}.

Bemerkung Praxis

Pivotisierung und Zeilenäquilibrierung immer zusammen anwenden! (um 'wahre' Pivotelemente zu finden). Statt $LR = A$ zerlege $LR = PD_zA$ und löse

$$LRx = PD_zb$$

- Optimale Kondition des Problems
- Stabiler Algorithmus (zudem: keine Probleme mit Nulleinträgen)

Berechnung Determinante

$$det(A) = det(D_z^{-1}P^{-1}LR) = \underbrace{det(D_z^{-1})}_{=\prod_{i=1}^{n} d_{i,i}^{-1}} \; \underbrace{det(P^{-1})}_{=(-1)^{\#\text{Vertauschungen}}} \; \underbrace{det(L)}_{=1} \; \underbrace{det(R)}_{=\prod_{i=1}^{n} r_{i,i}}$$

Beispiel: Klausuraufgabe A1, H2010

Die Musterlösung dieser Aufgabe befindet kann von der NumaMB Webseite heruntergeladen werden.

Zusatz: Berechnung Inverse

Folie 48

Gauß-Doolittle

Annahme: Zu A existiert eine LR-Zerlegung: $A = LR$ Dann gilt für jede Komponente

$$a_{i,j} = \sum_{k=1}^{n} l_{i,k} r_{k,j} = \sum_{k=1}^{\min(i,j)} l_{i,k} r_{k,j}$$

Für $i \leq j$ (Diagonale + rechts-oben) folgt damit:

$$\begin{aligned} a_{i,j} &= \sum_{k=1}^{i-1} l_{i,k} r_{k,j} + l_{i,i} r_{i,j} \\ &\Rightarrow \quad r_{i,j} = \left(a_{i,j} - \sum_{k=1}^{i-1} l_{i,k} r_{k,j} \right) / \underbrace{l_{i,i}}_{=1} \end{aligned}$$

Für $i > j$ (links-unten) folgt damit:

$$\begin{aligned} a_{i,j} &= \sum_{k=1}^{j-1} l_{i,k} r_{k,j} + l_{i,j} r_{j,j} \\ &\Rightarrow \quad l_{i,j} = \left(a_{i,j} - \sum_{k=1}^{i-1} l_{i,k} r_{k,j} \right) / r_{j,j} \end{aligned}$$

Eintrag $l_{i,j}$ bzw. $r_{i,j}$ hängt von den vorherigen Zeileneinträgen (i) in L ab und von den Spalteneinträgen (j) in R.

$$\begin{array}{c} \\ \\ \\ i \rightarrow \\ \\ \\ \end{array} \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ * & \cdots & * & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} = L \qquad \begin{array}{c} \begin{matrix} & & & j \downarrow & & \end{matrix} \\ \begin{pmatrix} * & & & * & & \\ & * & & * & & \\ & & \ddots & \vdots & & \\ & & & * & & \\ & & & & \ddots & \\ & & & & & * \end{pmatrix} \end{array} = R$$

Vorgehensweise um die Charakterisierungen von $l_{i,j}$ und $r_{i,j}$ zur Berechnung von L und R zu nutzen:

$$\begin{array}{ccc} \text{1. Zeile von } L \text{ ist gegeben.} & \longrightarrow & \text{1. Zeile von } R \text{ kann komplett berechnet werden.} \\ & \swarrow & \\ \text{1. Spalte von } R \text{ ist gegeben.} & \longrightarrow & \text{1. Spalte von } L \text{ kann komplett berechnet werden.} \\ & \swarrow & \\ \text{2. Zeile von } L \text{ ist gegeben.} & \longrightarrow & \text{2. Zeile von } R \text{ kann komplett berechnet werden.} \quad \text{Gauss-} \\ & \swarrow & \\ \text{2. Spalte von } R \text{ ist gegeben.} & \longrightarrow & \text{2. Spalte von } L \text{ kann komplett berechnet werden.} \\ & \swarrow & \\ \ldots & \longrightarrow & \ldots \end{array}$$

Doolittle hat keine Pivotisierung, funktioniert also nicht für alle reguläre Matrizen. Aber für Spezialfälle.

Sonderfall: symmetrisch positiv definite Matrizen

A s.p.d. $\quad \Rightarrow \quad R = D \cdot L^T$

Setzen wir $R = D \cdot L^T$ in den Gauss-Doolittle-Algorithmus ein, so ergibt sich folgende Abhängigkeit:

$$
\begin{array}{ccc}
\text{1. Zeile von } L \text{ ist gegeben.} & \longrightarrow & d_1 \text{ kann berechnet werden.} \\
 & \swarrow & \\
\text{1. Spalte von } R = DL \text{ ist gegeben.} & \longrightarrow & \text{1. Spalte von } L \text{ kann komplett berechnet werden.} \\
 & \swarrow & \\
\text{2. Zeile von } L \text{ ist gegeben.} & \longrightarrow & d_2 \text{ kann berechnet werden.} \\
 & \swarrow & \\
\text{2. Spalte von } R = DL \text{ ist gegeben.} & \longrightarrow & \text{2. Spalte von } L \text{ kann komplett berechnet werden.} \\
 & \swarrow & \\
\cdots & \longrightarrow & \cdots
\end{array}
$$

Der resultierende Algorithmus (unter Ausnutzung $R = DL^T$) ist der Cholesky-Algorithmus!

$\boxed{\text{Folie 57}}$