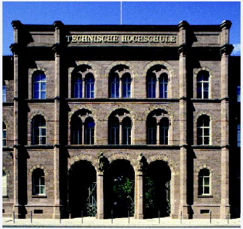
Übung zum Fach Rechnerunterstütze Mechanik I

J. Ruck, H. Erdle, T.-A. Langhoff, T. Böhlke

Chair for Continuum Mechanics
Institute of Engineering Mechanics
Department of Mechanical Engineering
Karlsruhe Institute of Technology (KIT)

WS 2017/2018



KIT
Karlsruher Institut für Technologie

Ü3: Lösung linearer Gleichungssysteme

Themen der 3. Übung

- LU-Zerlegung mit Spaltenpivotsuche (LUP-Zerlegung)
- Gauss-Seidel-Verfahren
- Cholesky Zerlegung (1.Testat)

Idee

Jede reguläre Matrix A lässt sich schreiben als $A = LUP$, mit:

$$L = \begin{pmatrix} 1 & 0 & \dots & \\ l_{21} & 1 & 0 & \dots \\ l_{31} & l_{32} & 1 & 0 \\ \vdots & \vdots & \ddots & \ddots \end{pmatrix}, U \quad = \begin{pmatrix} u_{11} & u_{12} & \dots & \dots \\ 0 & u_{22} & u_{23} & \dots \\ 0 & 0 & u_{33} & u_{34} \\ 0 & \dots & 0 & \ddots \end{pmatrix},$$

$$P \in \mathcal{P}(\mathbb{R}^n)$$

Damit lässt sich die lineare Gleichung $Ax = LUPx = y$ wie folgt lösen:

- Vorwärtseinsetzen: $z_i = (UPx)_i = y_i - \sum_{j=1}^{i-1} z_j l_{ij}$;
- Rückwärtseinsetzen: $w_i = (Px)_i = \left(z_i - \sum_{j=i+1}^{N} u_{ij} w_j\right) / u_{ii}$;
- Permutation der Einträge: $x = P^{\mathsf{T}} w$;

👍 Rechenaufwand und Speicherbedarf kleiner als bei Matrixinversion

👍 Zerlegung kann mehrfach zur Gleichungslösung verwendet werden

👍 L und U können in einer (i.A. voll besetzten) Matrix gespeichert werden

$\rightarrow$ Niedriger Speicherbedarf

Algorithmus

Zunächst: Ohne Permutationsmatrix P. Dann gilt

$$LU = \begin{pmatrix} u_{11} & u_{12} & \dots & \dots \\ u_{11}l_{21} & l_{21}u_{12}+u_{22} & l_{21}u_{13}+u_{23} & \dots \\ \vdots & \vdots & \ddots & \ddots \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & \ddots & \ddots \end{pmatrix}$$

- Erste Zeile: $u_{1j} = a_{1j}\ (j = 1, \dots, n)$
- Einsetzen in nächste Zeile: $l_{21} = a_{21}/u_{11}$; l_{21} einsetzen: $u_{2j} = a_{2j} - l_{21}u_{1j}\ (j = 2, \dots, n)$

  $\vdots$
- Erste k-Zeilen einsetzen, ...

Problem: Es kann nicht garantiert werden, dass $u_{ii} \neq 0$ gilt

Einbeziehung der Pivotsuche

Problem: Es kann nicht garantiert werden, dass $u_{ii} \neq 0$ gilt

Lösung: Nach Bearbeitung der i-ten Zeile das Element u_{ii} prüfen; falls (betragsmäßig) zu klein bzgl. anderer Komponenten:

- Finde maximales Element u_{ik} mit $(k > i)$, so dass:

$$|u_{ik}| > |u_{ij}| \quad \forall j > i \qquad \text{(mit Index } k\text{)}$$

- Vertausche die k-te und i-te Spalte der Matrix U und die k-te und i-te Zeile der Permutationsmatrix P

Vorteile der Pivotsuche

- Robustheit gegenüber Abschneidefehlern verbessert
- Faktorisierung kann für jede reguläre Matrix A berechnet werden
- Rechenzeit und Speicherplatzbedarf (fast) unverändert

Gauss-Seidel Algorithmus

Gesucht: Lösung x von

$$Ax = y \quad (A \in \mathrm{Sym}(\mathbb{R}^{n\times n}))$$

Ansatz: $A = L + D + U$ mit

- $L \in \mathbb{R}^{n\times n}$: $l_{ij} = 0 \ (\forall j \geq i)$,
- $D = \mathrm{diag}(d_1, \dots, d_n)$,
- $U \in \mathbb{R}^{n\times n}$: $u_{ij} = 0 \ (\forall i \geq j)$;

Dann gilt:	$(L + D)x = y - Ux$
Approximation:	$(L + D)x^{(j+1)} = y - Ux^{(j)}$
mit Startwert:	$x^{(0)} = 0$, $j = 0$;

$(L + D)$ ist untere Dreiecksmatrix $\rightarrow$ Lösung durch Vorwärtseinsetzen
$\rightarrow$ sehr niedriger Rechenaufwand!

Konvergenz, falls $\frac{\|Ax^{(j)} - y\|}{\|y\|} < \delta_{\mathrm{num}}$,
Fehler, falls $j > N_{\max}$.

Cholesky-Zerlegung (TESTAT 1)

Für jede symmetrische positive Matrix A (spd-Matrix) existiert eine untere Dreiecksmatrix L, so dass folgende Zerlegung gilt

$$A = LL^{\mathsf{T}}.$$

Algorithmus: Siehe Übungsblatt

Vorteile der Cholesky-Zerlegung

- Berechnung von L sehr einfach und schnell
- Effiziente Lösung des linearen Gleichungssystems
(etwa Faktor 2 gegenüber dem Gauss-Verfahren)
- Keine Spalten-/Zeilenvertauschungsoperationen
- Robust gegenüber Abschneidefehler (Maschinengenauigkeit)