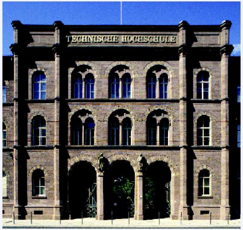
Übung zum Fach Rechnerunterstütze Mechanik I

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



Karlsruher Institut für Technologie

Ü2: Lineare Gleichungssysteme

Themen der 2. Übung

- Eigenschaften linearer Abbildungen
- Berechnen von Matrix- und Vektornormen
- Lösung linearer Gleichungssysteme
- Thomas-Algorithmus

Ausgabe der 1. Programmieraufgabe mit Abgabe
Abgabetermin: FIXME

Eigenschaften linearer Abbildungen

`det(A)`	$\det(A)$
`rank(A)`	$\operatorname{rank}(A)$
`trace(A)`	$\operatorname{tr}(A)$
$Au_i = \lambda_i u_i$	`eig`

Vektor- und Matrixnormen

Definition *(Vektornormen)*

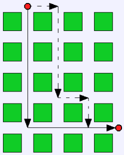
$$\|x\|_k = \begin{cases} \left(\sum_{i=1}^{d} |x_i|^k\right)^{1/k} & (k < \infty) \\ \max\limits_{i=1,\dots,d} |x_i| & (k = \infty) \end{cases}$$

Definition *(induzierte Matrixnorm)*

$$\|A\|_k = \sup_{\|x\|_k = 1} \|Ax\|_k$$

Definition *(Frobenius-Norm)*

$$\|A\|_F = \sqrt{\sum_{i,j=1,\dots,d} \sum A_{ij}^2} = \sqrt{A_{ij} A_{ij}}$$



Die 1-Norm $\| \bullet \|_1$ wird auch als Mannheimer Metrik bezeichnet. Man beachte: Es ist egal, welchen der möglichen Wege man hier wählt, so lange die x- und y-Koordinaten monotan fallen/steigen (*also: nie umdrehen*)!

Funktionen in Matlab

Definition einer Funktion in Matlab:

- Dateiname: FktName.m
- `function returnValue = FktName( Arg1, Arg2, ...)`

  Beispiele:

  ```
  function quad = Quadrat( a )
  quad = a .* a; % komponentenweise die Einträge multiplizieren
  ```

  Wichtig: Durch Verwendung von `.*` können auch Vektoren als Eingabe übergeben werden. Dies ist häufig sinnvoll und merklich schneller.
- Mehrere Rückgabewerte:

  `function [ r1, r2, r3 ] = FktName( Arg1, Arg2, ...)`

Der Thomas-Algorithmus

Gegeben:

$$T = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{pmatrix},$$

Gesucht:

$$x \in \mathbb{R}^n : \quad Tx = y \qquad (y \in \mathbb{R}^n).$$

Algorithmus

- Vorwärtseinsetzen

$$c_i^* = \frac{c_i}{b_i - c_{i-1}^* a_i} \text{ für } (1 < i \leq n-1), \quad c_1^* = c_1/b_1;$$

$$y_i^* = \frac{y_i - y_{i-1}^* a_i}{b_i - c_{i-1}^* a_i} \text{ für } (1 < i \leq n), \quad y_1^* = y_1/b_1;$$

- Rückwärtseinsetzen $(i = n, \ldots, 1)$

$$x_n = y_n^*,$$

$$x_i = y_i^* - c_i^* x_{i+1} \quad (1 \leq i < n);$$

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)