Prof. Dr. Dmitri Kuzmin
Dipl.-Technomath. Christopher Basting
Numerische Mathematik für Physiker und Ingenieure

Abgabe der **Theorieaufgaben bis Donnerstag, 09.06.2016, 16:15 Uhr** (einzeln oder zu zweit) in den Briefkasten Ihrer Übungsgruppe. Bitte vermerken Sie auf Ihrer Abgabe Ihre Übungsgruppe sowie Name und Matrikelnummer. Heften Sie mehrere Blätter zusammen.
Abgabe der **Programmieraufgaben bis Donnerstag, 09.06.2016, 16:15 Uhr** digital im Moodle-Arbeitsraum der Veranstaltung. Bei Abgabe zu zweit bitte nur einmal einreichen und im Kommentar den Namen und E-Mail Adresse des/der Koautors/Koautorin nennen. Sie können die Aufgaben in MatLab bzw. Octave schreiben. Alternativ kann auch in C oder C++ abgegeben werden, allerdings ohne mögliche Hilfestellung. Es kann nur lauffähiger Code bewertet werden.
Bei Fragen zu Vorlesung und Übungen können Sie neben der Übungsgruppe auch das **Tutorium** nutzen, jeweils **Mittwochs von 13:15 bis 14:45** Uhr im CIP-Pool der Mathematik (M946).

In allen folgenden Aufgaben: Geben Sie in jedem Gauss'schen Eliminationsschritt alle durchgeführten Zeilenumformungen explizit an (z. B. "$-2(\mathrm{II})$; $\cdot(-7)$" für: "ziehe zwei mal Zeile (II) ab, multipliziere anschließend mit -7.").

Aufgabe 6.1 (Gauss'sches Eliminationsverfahren | 2+6 Punkte)

a) Formulieren Sie für folgendes Problem ein lineares Gleichungssystem und lösen Sie es mit Hilfe des Gauss'schen Eliminationsverfahren:
Wenn vier Ochsen und vier Schafe acht Taels Gold kosten, sowie drei Ochsen und sechs Schafe auch acht Taels, was ist dann der Preis eines jeden Tieres?

b) Lösen Sie die beiden folgenden Gleichungssysteme (Darstellung in erweiterter Koeffizientenmatrix: das lineare Problem $Ax = b$ wird als $(A|b)$ dargestellt) mit Hilfe des Gauss'schen Eliminationsverfahren:

$$\left(\begin{array}{ccc|c} 3 & 3 & 7 & -1 \\ 4 & 4 & 9 & 3 \\ 5 & 5 & 11 & 6 \end{array}\right) \quad \text{und} \quad \left(\begin{array}{cccc|c} 2 & -1 & -3 & -1 & 9 \\ -4 & 2 & 8 & -3 & -23 \\ 0 & -1 & 2 & 2 & -2 \\ -4 & 5 & 10 & 14 & -10 \end{array}\right)$$

Aufgabe 6.2 (3+2+2 Punkte)

Betrachte die Matrix $A \in \mathbb{R}^{n\times n}$ mit

$$A := \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & \ddots & \vdots & 1 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ -1 & \cdots & \cdots & -1 & 1 \end{bmatrix}$$

a) Berechnen Sie die obere Dreiecksmatrix R der LR-Zerlegung.

b) Berechnen Sie $\|\cdot\|_1$, $\|\cdot\|_\infty$ von A und R.

c) Ändern Sie den Eintrag $a_{1n} := 1 + \varepsilon$. Wie lautet jetzt R?

Prof. Dr. Dmitri Kuzmin
Dipl.-Technomath. Christopher Basting
Numerische Mathematik für Physiker und Ingenieure

Programmieraufgabe 6.1 (LR-Zerlegung | 8 + 2 + 3 + 1 Punkte)

Implementieren Sie die LR-Zerlegung (ohne Pivotisierung) mit Vorwärts- und Rückwärtssubstitution. Schreiben Sie dazu die folgenden Routinen, ohne dabei enstprechende Funktionen in Matlab zu nutzen (wie etwa den Backslash-Löser \):

a) `function LU = LU_decompose(A)`
Diese Funktion soll die Matrix $A \in \mathbb{R}^{n \times n}$ in $A = L * U$ zerlegen, wobei L eine untere Dreiecksmatrix ist, bei der die Diagonaleinträge immer 1 sind und U eine obere Dreiecksmatrix ist. Die Matrizen L und U sollen platzsparend in der Matrix LU gespeichert werden (die Diagonale von L soll nicht mit abgespeichert werden, sie ist ja sowieso bekannt).

b) `function z = forward_solve(LU,b)`
Diese Funktion soll das Gleichungssystem $Lz = b$ lösen, wobei LU die oben beschriebene Matrix ist. Die Struktur von L soll dabei natürlich werden. Sie können hierzu Ihre Implementierung aus Programmieraufgabe 5.1 oder eine Musterimplementierung (siehe Moodle) erweitern.

`function x = backward_solve(LU,z)`
Diese Funktion soll das Gleichungssystem $Ux = z$ lösen, wobei LU die oben beschriebene Matrix ist. Sie können hierzu Ihre Implementierung aus Programmieraufgabe 5.1 oder eine Musterimplementierung (siehe Moodle) erweitern.

c) `function LU_Test()`
Testen Sie in dieser Funktion Ihre Implementierung aus Teilen **a)** und **b)** jeweils für $n = 10, 15, 20, 25$ für das lineare $n \times n$ Gleichungssystem $Vy = b$ mit der Vandermondematrix V bezüglich $x_i = \frac{n-i}{n}$, $i = 1, \ldots, n$, sowie der rechten Seite $b_i = 1$ für $i < \frac{n}{2}$ und $b_i = 2$ sonst. Geben Sie insbesondere das Residuum $\|Vy - b\|_2$ aus.
Hinweis: Zur Berechnung der Vandermonde-Matrix können Sie Ihre Implementierung aus Programmieraufgabe 5.1 nutzen oder den Matlab-Befehl `vander(x)`.

d) Was können Sie in **c)** beobachten? Schreiben Sie Ihre Beobachtung und Erklärung dieses numerischen Verhaltens in eine Text-Datei `PA_6.txt`.

Die Routinen sollen Schleifen möglichst vermeiden und natürlich nicht die in Matlab bereits vorhandene LU-Zerlegung benutzen.