Prof. Dr. Stefan Funken
Kristina Steih
Institut für Numerische Mathematik
Universität Ulm

Numerik 1
WS 2011/12

Matlab-Blatt 1

(Abgabe bis spätestens Dienstag, den 08.11.2011, 10:00 Uhr per Mail (s.u.).)

Aufgabe 1 *(Dreiecksmatrizen)* (8 Punkte)

In dieser Aufgabe wollen wir uns mit den Operationen Rx und $R^{-1}x$ ($R \in \mathbb{R}^{n\times n}$ obere Dreiecksmatrix und $x \in \mathbb{R}^n$) beschäftigen.

(a) Schreiben Sie eine Funktion matVec, welche als Input eine obere Dreiecksmatrix R und einen Vektor x erhält und einen Vektor b, der das Produkt Rx enthält, ausgibt.

(b) Schreiben Sie eine Funktion invMatVec welche als Input eine obere Dreiecksmatrix R und einen Vektor x erhält und einen Vektor b, der das Produkt $R^{-1}x$ enthält, ausgibt (Rückwärtseinsetzen).

(c) Laden Sie das Skript main1.m von der Vorlesungshomepage herunter und führen Sie darin die folgenden Schritte aus:

Zeile 9-15 Verwenden Sie R und x um ihre Funktionen aus Teil a und b zu testen, indem Sie das Ergebnis ihrer Funktion mit dem der Matlab-builtins $*$ bzw. $\backslash$ vergleichen.

Zeile 20-44 Führen Sie für beide Funktionen Zeitmessungen durch. Legen Sie dabei für $n \in \{2^i \mid i = 1..12\}$ jeweils eine zufällige obere Dreiecksmatrix Matrix $R \in \mathbb{R}^{n\times n}$ und einen zufälligen Vektor $x \in \mathbb{R}^n$ an, und berechnen Sie Rx bzw. $R^{-1}x$. Achtung: Je nach Computer können die Rechenzeiten für $n = 2^{10}, 2^{11}, 2^{12}$ groß werden. Für ältere Computer genügt es, die Rechnungen für $n \in \{2^i | i = 1..10\}$ auszuführen (im Skript *main1.m* anpassen).

(d) Betrachten Sie die Ergebnisse im loglog-plot und beschreiben und erklären Sie kurz was Sie sehen. Wie kann das Verhalten für kleine n erklärt werden?

Aufgabe 2 *(Memory considerations)* (8 Punkte)

In order to store a full $n \times n$ matrix, n^2 floats have to be stored. For a 1000×1000 matrix 10^6 floats have to be stored, which corresponds to a storage size of 8MB. Since triangular matrices have only at most $\frac{n^2+n}{2}$ non zero entries, only these entries have to be stored. Doing this one can save almost 50% of storage.
One possibility to store only the neccessary entries of an upper triangular matrix is to write the entries into a vector in the following way:

$$A = \begin{pmatrix} a_{11} & a_{12} & & \cdots & a_{1n} \\ & a_{22} & & \cdots & a_{2n} \\ & & \ddots & & \vdots \\ & & & & a_{nn} \end{pmatrix} \quad \rightarrow \quad vec = \begin{pmatrix} a_{11}, & a_{12}, & \cdots & a_{1n}, & a_{22}, & \cdots & a_{2n}, & \cdots & a_{nn} \end{pmatrix}$$

1. Download the file *full2half.m* from the lecture homepage. The file contains the function *vec = full2half(A)* which transforms the matrix A into a vector *vec* as described above.
2. Modify your functions *matVec* and *invMatVec* such that they work for matrices which are given in the vector format. Name your new functions *halfMatVec* and *halfInvMatVec*, respectively.
3. Download the script *main2.m* from the homepage and proceed as in Aufgabe 1.
4. Compare your time measurements with the ones from Aufgabe 1 and describe and explain what you see.

Senden Sie alle Dateien (*m*-Files, Plots, Erklärungen) in einer Email mit dem Betreff `Num1-BlattM1` an kristina.steih@uni-ulm.de (Am Besten alle Dateien in ein zip-File packen und das schicken). Aus der Email sollte klar hervorgehen, von welchen beiden Studenten die Lösung ist. Bitte schreiben Sie außerdem dazu, in welcher Übungsgruppe (Wochentag und A bzw. B) Sie jeweils sind.