

Universität Bielefeld Fakultät für Physik	Computerphysik SS 2012	Prof. Dr. Jürgen Schnack jschnack@uni-bielefeld.de
--	---------------------------	---

## Aufgabenblatt 11

### 11.1 Lineare Gleichungssysteme

In der Vorlesung wurde die Bibliothek LAPACK besprochen. Aus dieser Bibliothek haben wir die Unteroutine DGESV verwendet, um lineare Gleichungssysteme der Form

$$A \cdot X = B \quad (1)$$

zu lösen. Dabei ist  $A$  eine Matrix,  $B$  und  $X$  haben die gleiche Dimension, sind also z.B. Vektoren oder Matrizen gleicher Größe.

- a. Informieren Sie sich über die Bibliothek LAPACK. Dazu können Sie zum Beispiel bei [www.netlib.org/lapack](http://www.netlib.org/lapack) nachschauen. Informieren Sie sich auch über die Parameter, die an die Routine DGESV übergeben werden müssen. Entsprechende Informationen finden Sie unter „Individual Routines“.

Bei LAPACK handelt es sich um eine Bibliothek, die ursprünglich in Fortran 77 geschrieben und dann später für andere Programmiersprachen portiert wurde. Informieren Sie sich unter [www.netlib.org/clapack/readme](http://www.netlib.org/clapack/readme) darüber, was alles zu beachten ist, wenn eine LAPACK-Funktion innerhalb eines C-Programms aufgerufen werden soll. Wichtig ist insbesondere der Unterschied zwischen Fortran und C bei der Speicherung zweidimensionaler Felder („column-major order“ im Gegensatz zu „row-major order“). Wiederholen Sie, wie man auf die Komponenten einer Matrix statt (wie gewohnt) mit zwei Indizes mit einem linearen Index zugreifen kann.

- b. Erstellen Sie ein Programm in C/C++ (oder der Programmiersprache Ihrer Wahl), das die Matrizen  $A$  und  $B$  einliest und das lineare Gleichungssystem durch Aufruf der Routine DGESV löst. Verwenden Sie die auf meiner Webseite hinterlegten Matrizen.

Wenn Sie zur Lösung der Aufgabe ein C-Programm auf einer Linux-Maschine schreiben, dann ist (neben den obigen Hinweisen) Folgendes zu beachten. Bis jetzt haben Sie nur einfache „externe“ Funktionen wie z.B. `gaussLegendre.h` kennengelernt, deren Source-Code in das Hauptprogramm durch einen `#include`-Befehl eingebunden werden musste, um die Funktion nutzen zu können. Kompliziertere Bibliotheken (wie z.B. LAPACK) liegen dagegen typischerweise in Form von bereits kompilierten Objekt-Dateien (Endung z.B. `.a` oder `.so`) vor. Die zugehörige Header-Datei (Endung `.h`) enthält dann nicht den kompletten Source-Code (bei kommerziellen Bibliotheken werden Sie diesen niemals zu Gesicht bekommen), sondern nur Deklarationen der Funktionen, d.h. die benötigten Informationen über Rückgabewerte und Argumente der Funktionen.

Um in Ihrem Programm LAPACK-Funktionen verwenden zu können, müssen Sie zunächst die Header-Dateien `f2c.h` und danach (!) `clapack.h` einbinden. Beim Kompilieren müssen dann die drei Bibliotheken `liblapack.a`, `libblas.a` und `libf2c.a` korrekt in Ihr Programm integriert („gelinkt“) werden. Kopieren Sie dazu die drei Dateien in dasselbe Verzeichnis, in dem sich auch Ihr Programm befindet. Kompilieren Sie das Programm dann folgendermaßen:

```
gcc -o <Name> <Source Code> -L. -llapack -lblas -lf2c -lm -Wall
```

Fortgeschrittene können auch die zentral gespeicherten Bibliotheken (meist irgendwo in `/usr/lib`) nutzen und den Compiler-Befehl entsprechend abändern.

- c. Wie kann man das Inverse einer Matrix bestimmen?
- d. **Zusatzaufgabe:** Schreiben Sie selbst ein Programm, das für eine beliebige Matrix  $A$  eine LU-Zerlegung durchführt. Testen Sie es anhand des Beispiels aus b. Welchen Vorteil bietet eine LU-Zerlegung im Vergleich zu anderen Methoden?