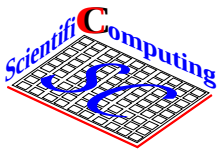
Institute of Scientific Computing
Technical University Braunschweig
Prof. Hermann G. Matthies, Ph.D.



Simulation of Fluid Dynamics - Introduction to MATLAB

This introduction should give a first impression of the program MATLAB (Abbreviation for „Matrix laboratory“), which comes from the Mathworks company (WWW-Address: http://www.mathworks.com/). Matlab can be started directly with the command `matlab`. After that you get some hints (i.e. help, demo, etc.) about commands you can try in the beginning. First you should get some help about the help system with `help help`. The help system is nearly as good as the printed manuals for Matlab (which can be borrowed from the consulting office of the computer centre). After that you can get a list of the different help topics offered by Matlab with the command `help` (To prevent matlab from scrolling the text enter `more on`). If you enter the command `help topic`, where `topic` is a possible subsection (like general, elmat, funfun, etc.), you can find specific information in the help system. Other things like direct search (`lookfor ...` or similar commands) are explained under `help help`. If you want to find some information about the numerical solution of differential equations, you will first choose `help funfun` and then `help ode23` which gives your detailed information about a Runge-Kutta method. A first introduction to Matlab can be started with `intro`. This introduction shows and explains the basic properties and possibilities of Matlab. Furthermore you can start the so called *Expomap* with `expo` or `demo`, which offers loads of information and examples to the installed toolboxes.
During this practical course you should learn how to solve sparse systems of linear equation, which means a system of equations $Ax = b$, where the matrix A has only a few nonzero elements. The following exercises should provide some examples for sparse matrices and show some ways to solve these sparse linear equations. Matlab stores sparse matrices very effective and provides several operations and functions to work with these matrices (`help sparfun`).

Exercise 1:
If you discretise a partial differential equation with a Finite-Difference method, you'll get

often linear systems of equations, where the matrix $A \in I\!R^{n\times n}$ has often the following form

$$A = \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & \ddots & & \vdots \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2 \end{pmatrix}$$

.

Create a Matlab function which generates this Matrix A with an arbitrary dimension $n \in I\!N$. Compute for $n = 50$, 100, 250 the eigenvalues of A and the Cholesky decomposition of A, and solve the system of linear equations $Ax = b$, letting $b \in I\!R^n$ be $\|b\|_\infty = 1$ a random vector.

Exercise 2:

Use the command `load waerme.mat` to load the matrix A and the vector b, which come from the discretisation of a heat equation. With `spy(A)` you can visualise the non-zero structure of the matrix A.

1. Determine the Cholesky factor for the sparse matrix A. How many new nonzero elements (Fill-ins) are created during the Cholesky decomposition? Explain your observations.

2. To reduce the fill-in the rows and columns of the matrix A become reordered with some Graph algorithms. Test the reordering algorithms already implemented in Matlab by applying them to A and computing the Cholesky factor of the reordered matrix afterwards. How do the reordered matrix A and it's Cholesky factor look like now?

3. Solve the system of equations $Ax = b$ with the Cholesky factor from A. This is equivalent to a direct solution of the linear system of equations.

4. Ofter iterative methods are used for very large sparse systems of linear equations. A simple iterative procedure is the SOR-method or a more effective the conjugate gradient method with preconditioning (PCG-method). During the practical course you should use both methods (and in the last assignment also a Multigrid method). Solve the given system of linear equations $Ax = b$ with the SOR- and PCG-method by implementing a small Matlab script for it.