Technische Universität München
Institut für Informatik
Dr. Miriam Mehl
Markus Brenk
Tobias Neckel

WT 2006/2007

Practical Course Scientific Computing and Visualization

Worksheet 3

distributed: 22.11.2006
due to: 04.12.2006, 18:00 pm (per email to `neckel@in.tum.de` and `brenk@in.tum.de`)
personal presentation: 05.12.2006 (exact slots will be announced)

In this worksheet, we switch to differential equations for functions with more than one independent variable, for example functions depending on several and space dimensions. Such differential equations are called partial differential equations. We start with a first simple example, the two-dimensional stationary heat equation

$$T_{xx} + T_{yy} = -2\pi^2 \sin(\pi x)\sin(\pi y) \tag{1}$$

on the unit square $]0;1[^2$ with the temperatur $T(x,y)$, the two-dimensional coordinates x and y, and homogeneous Dirichlet boundary conditions

$$T(x,y) = 0 \text{ forall } (x,y) \text{ in } \partial]0;1[^2. \tag{2}$$

The boundary value problem (1),(2) has the analytical solution

$$T(x,y) = \sin(\pi x)\sin(\pi y). \tag{3}$$

a) The discretization of (1) and (2) leads to a large system of linear equations for the values $T_{i,j}, i \in \{1,\ldots,N_x\}, j \in \{1,\ldots,N_y\}$ denoting the approximative values of the temperature T at the discrete grid points $\left(i \cdot \frac{1}{N_x+1}, j \cdot \frac{1}{N_y+1}\right)$.

Use the finite difference approximation of the second derivatives

$$T_{xx}|_{i,j} \approx \frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{h_x^2},$$

$$T_{yy}|_{i,j} \approx \frac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{h_y^2}$$

with $h_x = \frac{1}{N_x+1}$ and $h_y = \frac{1}{N_y+1}$.

Sketch a few lines of the resulting sytem in the following scheme

$$\underbrace{\begin{pmatrix} \quad & & & & & & & & & & \quad \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{pmatrix}}_{=:A} \begin{pmatrix} T_{1,1} \\ T_{2,1} \\ \vdots \\ T_{N_x,1} \\ T_{1,2} \\ \vdots \\ T_{N_x,2} \\ \vdots \\ T_{1,N_y} \\ \vdots \\ T_{N_x,N_y} \end{pmatrix} = \underbrace{\begin{pmatrix} \quad \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{pmatrix}}_{=:b} .$$

b) Implement a function creating the matrix from **a)** as a function of N_x and N_y.

c) Implement a Gauss-Seidel solver for the system in **a)** as a function of the right hand side b, N_x, and N_y. The output of the solver is a matrix with dimensions $N_x + 2$ and $N_y + 2$ containing the computed approximate values of T at the grid points $(i \cdot h_x, j \cdot h_y)$ in entry (i, j). As a termination criterion for the iteration use an accuracy limit of 10^{-4} for the residual norm.

Remark 1: Do not(!!!) use the explicit system matrix from **b)** in your Gauss-Seidel solver. Use your knowledge about the particular, constant form of the lines of the linear system to completely avoid any storage of matrix entries!

Remark 2: The residual norm for a general system $Ax = b$ is defined as

$$R = \sqrt{\frac{1}{N} \sum_k \left(b_k - \sum_m a_{k,m} x_m \right)^2},$$

where N denotes the number of unknowns. In the Gauss-Seidel solver, $N = N_x * N_y$, $a_{i,j}$ are not stored (but known), and the entries of x are the temperature values $T_{i,j}$ at the grid points $(i \cdot h_x, j \cdot h_y)$.

d) Solve the system from **a)**

1) storing the system matrix as a normal (full) $N_x \times N_y$ matrix and using the matlab direct solver,
2) storing the system matrix as a sparse matrix and using the matlab direct solver,

  3) without storing the system matrix (use Gauss-Seidel with zero as initial guess for T!).

e) Visualize the solutions as a

  1) a coloured surface,
  2) a contour plot.

f) Compare the runtimes and the storage requirements (measured by the number of entries of the arrays and/or vectors needed) for 1) – 3) in **c)** and for $N_x = N_y = 7, 15, 31, 63$:

direct solution with full matrix

N_x, N_y	7	15	31	63
runtime				
storage				

direct solution with sparse matrix

N_x, N_y	7	15	31	63
runtime				
storage				

iterative solution with Gauss-Seidel

N_x, N_y	7	15	31	63
runtime				
storage				

g) Compute the solutions for $N_x = N_y = 7, 15, 31, 63, 127$ with the Gauss-Seidel solver and fill in the resulting errors

$$e = \sqrt{\frac{1}{N_x \cdot N_y} \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} (T_{i,j} - T(x_i, y_j))^2}$$

in the following tabular:

$N_x = N_y$	7	15	31	63	128
error					
error red.	—				

Questions:

1) How many non-zero entries do you achieve in the system matrix from **a)**?
2) Compare the number of non-zero entries to the number of entries of a full matrix with the same size. What conclusion concerning the methods to be used for the storage of the system matrix can you draw for increasing N_x and N_y?
3) Which solver would you suggest to use for very big N_x and N_y?
4) Use the results of **g)** to guess the convergence order of the discretization used for the Laplacian operator.