Technische Universität Berlin
Institut für Mathematik
Prof. Dr. Jörg Liesen
Dr. Agnieszka Międlar
André Gaul

WS 11/12
To be submitted in the lecture on 23.11.2011

Numerische Mathematik für Ingenieure II
Homework 5

Programming exercise 8: (8 points)
Consider the Poisson equation with *Neumann* boundary values for the domain $\Omega =]0,1[\times]0,1[$ and the functions $f : \Omega \to \mathbb{R}, \quad g : \partial\Omega \to \mathbb{R}$:

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega \\ \frac{\partial u}{\partial \nu} &= g \qquad \text{on } \partial\Omega. \end{aligned}$$

Implement the finite difference method with a uniform grid and one-sided differences for the Neumann boundary as in Example 4.19 in the script. Therefore, write a MATLAB `function [Lh,fh] = p08getLS(n,f,g)` that returns the matrix $\texttt{Lh} \in \mathbb{R}^{\texttt{n}^2,\texttt{n}^2}$ and right hand side $\texttt{fh} \in \mathbb{R}^{\texttt{n}^2,1}$ for the grid size $h = \frac{1}{\texttt{n+1}}$ by using the lexicographical order of the unknowns. `f` und `g` are function handles (for example `f(x,y)` returns the function's value at $[\texttt{x},\texttt{y}]^\mathsf{T} \in \Omega$). Notice that the discretized right hand side `fh` is constructed from the right hand side function `f` and the Neumann boundary values `g`:

$$\texttt{fh((j-1)*n+i)} = f(ih, jh) + \frac{1}{h} \begin{cases} g(h,0) + g(0,h), & \text{for } i = j = 1 \\ g(nh,0) + g(1,h), & \text{for } i = n, j = 1 \\ g(1,nh) + g(nh,1), & \text{for } i = j = n \\ g(h,1) + g(0,nh), & \text{for } i = 1, j = n \\ g(ih,0), & \text{for } 1 < i < n, j = 1 \\ g(ih,1), & \text{for } 1 < i < n, j = n \\ g(0,jh), & \text{for } i = 1, 1 < j < n \\ g(1,jh), & \text{for } i = n, 1 < j < n \\ 0, & \text{else.} \end{cases}$$

Hint: use `meshgrid` and `reshape` and verify the correct order of the unknowns.

Programming exercise 9: (8 points)
Verify your implementation of `p08getLS` with the following right hand side and Neumann boundary function:

$$\begin{aligned} f(x,y) &= \cos(2\pi x)\exp(y^3)(4\pi^2 - 6y - 9y^4) \\ g(x,y) &= \begin{cases} 3\cos(2\pi x)\exp(1), & \text{if } y = 1 \\ 0, & \text{else.} \end{cases} \end{aligned}$$

A solution of the PDE in programming exercise 8 with these functions is

$$u(x, y) = \cos(2\pi x) \exp(y^3).$$

Write a `function [lam,err,eoc]=p09test()` that returns three vectors of length 8 by carrying out the following computations for all $\mathtt{p} \in \{1, \dots, 8\}$:

- Determine the matrix `Lh` and right hand side `fh` for the grid size $h_{\mathtt{p}} = 1/(2^{\mathtt{p}} + 1)$ (that is $\mathtt{n} = 2^{\mathtt{p}}$).
- Compute the approximate solution $\mathtt{uh} \in \mathbb{R}^{2^{\mathtt{p}},1}$ by using the extended, nonsingular system of linear equations
$$\begin{bmatrix} \mathtt{Lh} & e \\ e^T & 0 \end{bmatrix} \begin{bmatrix} \mathtt{uh} \\ \mathtt{lam(p)} \end{bmatrix} = \begin{bmatrix} \mathtt{fh} \\ 0 \end{bmatrix}.$$
Use MATLAB's backslash operator to solve this system.
- Compute the given exact solution $\mathtt{uex} \in \mathbb{R}^{2^{\mathtt{p}},1}$ at the grid points.
- Determine the error and experimental order of convergence (EOC):

$$\begin{aligned} \mathtt{err(p)} &= \|\mathtt{uh} - \mathtt{uex}\|_\infty \\ \mathtt{eoc(p)} &= \frac{\log(\frac{\mathtt{err(p)}}{\mathtt{err(p-1)}})}{\log(\frac{h_p}{h_{p-1}})} \qquad \text{for } \mathtt{p} > 1. \end{aligned}$$

Monitor the EOC and verify that it converges to 1. Recall page 41 of the script or your notes from the tutorial on 18.11.2011 to realize which role λ (here `lam(p)`) plays.