University of Konstanz
Department Mathematics and Statistics
D. Beermann, L. Mechelli, Prof. Dr. S. Volkwein

Winter semester 2016/2017

Exercises for
Theory and Numerics of Partial Differential Equations

`http://www.math.uni-konstanz.de/numerik/personen/beermann/teaching`

Sheet 7

(first sheet in the Numerics part of the lecture)

Deadline: Thursday, 22/12, in the lecture

Exercise 7.1 (Matlab) (8 points)

Assorted files: `main71.m` and `plot_grid.m`
Please follow the *programming guidelines* available under the above url.

The task is to write a program which generates a two-dimensional Finite-Differences (FD) grid of a set $\Omega \subset \mathbb{R}^2$. This is supposed to be done by calling a function

`grid = compute_fd_grid(box, indFunc)`

which takes two arguments

1. `box` is a variable of the type `struct` and contains information about a square $[\underline{x}_1, \overline{x}_1] \times [\underline{x}_2, \overline{x}_2] \subset \mathbb{R}^2$, which is a superset of the original set: $\Omega \subset [\underline{x}_1, \overline{x}_1] \times [\underline{x}_2, \overline{x}_2]$. From the input within `box`, two vectors `x1`, `x2` are supposed to be generated which represent equidistant discretizations of the two dimensions. The function should be able to handle two combinations of input:[1]

   (a) Input of the corner points of the square in a matrix `range`, of the shape

   $$\texttt{range} = \begin{bmatrix} \underline{x}_1 & \overline{x}_1 \\ \underline{x}_2 & \overline{x}_2 \end{bmatrix}$$

   Additional input of a `2x1` vector `Nx` such that `x1` will be an equidistant discretization of $[\underline{x}_1, \overline{x}_1] \subset \mathbb{R}$ with `Nx(1)` points. Same for `x2`.

   (b) Input of `range` from above and a `2x1` vector `dx` such that `x1` is an equidistant discretization of $[\underline{x}_1, \overline{x}_1] \subset \mathbb{R}$ with the step width `dx(1)`. Same for `x2`.

   From `x1` and `x2` and by use of the Matlab comand `ndgrid`, generate two matrices `X1` and `X2` which represent the square numerically. Based on these, build a coordinate list `p` such that `p(:,i)` contains the (x_1, x_2)-coordinates of the i-th grid point and `size(p,2)` is identical to the total number of grid points in the discretized square $[\underline{x}_1, \overline{x}_1] \times [\underline{x}_2, \overline{x}_2]$.

2. `indFunc` stands for a function $\mathbb{1}_\Omega : \mathbb{R}^2 \to \{0, 1\}$ which indicates for every point $x \in \mathbb{R}^2$ if it is contained in Ω or not, meaning that it holds for all $x \in \mathbb{R}^2$: $\mathbb{1}_\Omega(x) = 1 \Leftrightarrow x \in \Omega$. `indFunc` is of the class `function_handle` and has the form `ind = indFunc(x1,x2)`, expecting two vectors `x1`, `x2` of same length and returning a logical vector `ind` of same size such that `ind(i)` indicates if the point with the coordinates `x1(i)` and `x2(i)` lies in Ω. Now, walk over the already computed rectangle points in `p` and check by using `indFunc` if they are part of the set Ω. If not, eliminate them from `p`.[2]

[1] This means that the function should check the fields of `box` to determine what type of input is passed: If the variables `a` and `b` form a valid input combination, `box` should contain the fields `box.a` and `box.b`. If a `box` is passed that contains invalid input, the function should be terminated. Useful Matlab commands: `isfield` and `error`. To familiarize yourself with the possible inputs, call the `main1.m` script first to see the produced input types.

[2] Hint: By iterating over the points in `p` in reverse order - i.e. from `p(:,end)` to `p(:,1)` - you can eliminate the appropriate columns directly by the command `p(:,i) = []`. This would not be possible in forward order (why?).

The output variable `grid` is supposed to be a `struct` containing the fields `range`, `X1`, `X2`, `p` and `Np` (Number of grid points). Test your generated function by calling the script `main71.m`.

Exercise 7.2 (Theory) (6 points)

Let $\Omega \subset \mathbb{R}^n$, $n \in \mathbb{N}$, be a bounded domain with C^1-Boundary $\partial\Omega$. The scalar product between two functions $u, v \in C^0(\overline{\Omega})$ is defined as:

$$\langle u, v \rangle = \int_\Omega uv \, \mathrm{d}x.$$

For a function $\psi \in C(\partial\Omega)$, let be

$$D_\psi = \left\{ u \in C^1(\overline{\Omega}) \text{ s.t. } u(x) = \psi(x) \text{ for } x \in \partial\Omega \right\}$$

and consider the Differential Operator L:

$$L : C^2(\Omega) \cap D_0 \to C(\Omega) \; , \; Lu = -\Delta u.$$

Prove that:

i) $\langle Lu, v \rangle = \langle u, Lv \rangle$ for all $u, v \in C^2(\Omega) \cap D_0$,

ii) L is a positive-definite operator.

Exercise 7.3 (Theory) (6 points)

Let $\{u_n\}_{n\in\mathbb{N}}$ be a succession of functions $u_n : \overline{\Omega} \to \mathbb{R}$, where Ω is a bounded domain of $\mathbb{R}^d$. Each function u_n is the solution of a Laplace Problem of this type:

$$\begin{cases} \Delta u_n = 0 & \text{in } \Omega \\ u_n = \frac{1}{n}\sin(|x|) & \text{on } \partial\Omega \end{cases} \tag{1}$$

where $|\cdot|$ denotes the Euclidean Norm. Answer[3] the following questions:

i) When is problem (1) solvable and what is the lowest regularity you expect from solutions? (i.e. in what spaces does u_n live in?)

ii) Which is the value, for all $x \in \overline{\Omega}$, of the following limit[4]:

$$\lim_{n\to\infty} u_n(x) = ?$$

[3] As ever in mathematics the proof is the main part of an answer.

[4] This limit is called Pointwise limit.