University of Konstanz
Department Mathematics and Statistics
L. Mechelli, J. Lu, Prof. Dr. S. Volkwein

Winter Semester 2017/2018

Exercises for

Theory and Numerics of Partial Differential Equations

http://www.math.uni-konstanz.de/numerik/personen/mechelli/teaching.html

Exercise Sheet 8

Submission[1]: 11th January at 10:00

Exercise 1. (Theory, 4 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

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

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 2. (Theory, 6 points)
We consider a one-dimensional stationary *advection-diffusion-reaction* equation:

$$\begin{gathered} -\underbrace{u''(x)}_{\text{Diffusion}} + \underbrace{b(x)u'(x)}_{\text{Advection}} + \underbrace{c(x)u(x)}_{\text{Reaction}} = \underbrace{f(x)}_{\text{Source term}} \qquad \text{for all } x \in (0,1) \\ u(0) = u(1) = 0 \end{gathered} \tag{1}$$

with $c(x) > 0$ and $b(x) > 0$ for all $x \in (0,1)$. u describes the steady state of a substance contained in a flowing medium. The diffusion term describes the local spreading of the substance, the advection term represents the transport from the medium flow, the reaction term accounts for the reaction between substance and medium and the source term allows for external injection or extraction of the substance.

1. Discretize (1) by introducing an equidistant grid of $[0,1]$, i.e. $\{x_0, x_1, \ldots, x_N\}$ with $x_i = ih = iN^{-1}$, $N \in \mathbb{N}$, and considering the *Discrete Laplacian* for the diffusion term and the *central difference scheme*[2] for the advection term. Write the discretized version of (1) as a linear problem $A\tilde{u} = r$.

2. A matrix $B = (B_{ij})_{i,j=1,\ldots,N} \in \mathbb{R}^{N \times N}$ is called *strictly diagonally dominant* if $|B_{ii}| > \sum_{\substack{i=1 \\ i \neq j}}^{N} |B_{ij}|$ holds for $i = 1, ..., N$. It can be shown that a strictly diagonally dominant matrix is always regular. Which assumption is required on h to be sure that the matrix A from above is strictly diagonally dominant and thus the discretization performed in 1. is well-defined?

3. In 2., you should have realized that the advection velocity field $b(x)$ in (1) may cause problems for the discretization. Now, repeat 1. by using - instead of the central difference scheme - a so-called *upwind difference scheme*[3] for the discretization of the advection part.

[1] The Theory Exercises will be collected at the begin of the lecture. The Programming Exercises has to be sent by email to luca.mechelli@uni-konstanz.de (group of Tuesday) and to hai.nguyen-pham@uni-konstanz.de (group of Wednesday) **before** the submission's deadline.

[2] $u'(x) \approx \frac{u(x+h) - u(x-h)}{2h}$

[3] $u'(x) \approx \frac{u(x) - u(x-h)}{h}$

4. If we denote the problem resulting from 3. by $A^{SU}\tilde{u} = r^{SU}$, check again the diagonal dominance of A^{SU} as in 2. What do you observe?

Exercise 3. (Programming, 6 points)
Let $u \in C^2([0, L], \mathbb{R})$ with $L > 0$ and consider the following one-dimensional Poisson Problem:

$$\begin{cases} -u''(x) = f(x) & \text{for } x \in (0, L) \\ u(0) = u(L) = 0 \end{cases} \tag{2}$$

with $f \in C^0([0, L], \mathbb{R})$. In the main file `main_8_3`, solve (2) numerically with finite difference method following these instructions:

1. Choose the number of nodes $N+1$ and generate finite difference grid of $[0, L]$ corresponding to the $N+1$ nodes $x_0, \ldots, x_N$.
2. Build the matrix A which represents the *Discrete Laplacian* operator on the grid and the right hand side b which represents the values of $f(x)$ on the grid. Solve the linear system with the MATLAB backslash command \ for different $f(x)$:
   i. $f_1(x) = \frac{p(x)}{T}$, where $p(x) = 5 - x(x - L)$ represents the magnitude of a transverse distributed load, to which a string of length L under constant tension $T = 10$ is subjected. The solution $u_1(x)$ represent the transverse displacement at point x.
   ii. $f_2(x) = \frac{q(x)}{k}$, where $q(x) = \frac{100}{\sqrt{2\pi}} e^{-16(2x-L)^2}$ represents a distributed volumetric heat source applied to a homogeneous heat conducting bar of length L, where the bar's temperature $u_2(x)$ is maintained at zero at the end points. $k = 80$ is the constant heat conductivity of the iron at ambient temperature. (To be precise, in the model we have assumed that the transversal section of the rectangular parallelepiped bar is a square.)
3. Plot the solution $u_1(x)$ and $u_2(x)$.

Exercise 4. (Programming, 4 points)
Write a program which generates a two-dimensional finite differences 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)
```

whose output is a variable of the type `struct`, taking two input arguments:

1. `box` is a variable of the type `struct` and contains information about a rectangle $[a_1, b_1] \times [a_2, b_2] \subset \mathbb{R}^2$, which is a superset of the original set: $\Omega \subset [a_1, b_1] \times [a_2, b_2]$. The variable `box` contains the corner points of the rectangle in a matrix `range=`$[a_1, b_1; a_2, b_2]$ and a `2x1` vector `Nx`, for the creation of the `(Nx(1)+1)`$\times$`(Nx(2)+1)` rectangular grid.
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.

In function `compute_fd_grid` you have to:

i. Create the vector `box.x1` and `box.x2` containing the discretization nodes of each interval $[a_1, b_1]$ and $[a_2, b_2]$;

ii. Generate the rectangular grid with the MATLAB command `ndgrid`, which gave in output the grid points coordinates as two matrices `grid.X1` `grid.X2`;

iii. Build a coordinate list `grid.Nodes_List` such that `grid.Nodes_List(:,i)` contains the (x_1, x_2)-coordinates of the i-th grid point and save the number of grid points in to `grid.Num_Nodes`;

iv. According to `indFunc`, walk over `grid.Nodes_List` and remove[4] all the points of the rectangle which are outside of the domain Ω. (Change also `grid.Num_Nodes`)

To resume: The output variable `grid` should contain `box`, `X1`, `X2`, `Nodes_List` and `Num_Nodes`. Test your generated function by calling the script `main_8_4`, downloadable by the above url.

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