Technical University of Berlin
Faculty II - Mathematics and Natural Sciences
Institute of Mathematics
Dr. D. Peschka, A. Selahi

Numerical Mathematics II for Engineers

Homework Assignment 4

Submitted on November 18th, 2019

by **Group 5**

Kagan Atci	338131	Physical Engineering, M.Sc.
Navneet Singh	380443	Scientific Computing, M.Sc.
Riccardo Parise	412524	Scientific Computing, M.Sc.
Daniel V. Herrmannsdoerfer	412543	Scientific Computing, M.Sc.

Exercise 1

Considered is the Laplace operator $Lu = -\Delta$ in $\Omega = (0, L)^2$ and $u \in C^4(\bar{\Omega})$.

a) Let $\underline{x} = [x_1, x_2]^T \in \Omega_h$. Applying the Laplace operator in the form of finite difference on $u(\underline{x})$ takes place by differentiating $u(\underline{x})$ with respect to x_1 and x_2 one at a time. The D^-D^+ stencil is employed as the 2nd order finite difference operator with one neighbor on either side at the distance h.

$$\frac{\partial^2 u}{\partial x_1^2} = \frac{u(x_1 + h, x_2) - 2u(x_1, x_2) + u(x_1 - h, x_2)}{h^2} + \mathcal{O}(h^2) \tag{1}$$

$$\frac{\partial^2 u}{\partial x_2^2} = \frac{u(x_1, x_2 + h) - 2u(x_1, x_2) + u(x_1, x_2 - h)}{h^2} + \mathcal{O}(h^2) \tag{2}$$

With remainders neglected in Equations (1) and (2), the Laplace operator can be approximated by adding both equations

$$\Delta u = \frac{u(x_1 + h, x_2) + u(x_1, x_2 + h) - 4u(x_1, x_2) + u(x_1 - h, x_2) + u(x_1, x_2 - h)}{h^2} + \mathcal{O}(h^2). \tag{3}$$

Since u is a four times continuously differentiable function in $\bar{\Omega}$ and the remainder is $\mathcal{O}(h^2)$, it holds $||f_h - L_h R_h u||_h \in \mathcal{O}(h^2)$ as $h \to 0$, where f_h is the right hand side of the BVP, L_h the finite difference matrix and R_h the restriction operator within $\bar{\Omega}$. Hence, Equation (3), also called as the 5-point stencil, is considered as a 2nd order consistent approximation of Δu.

b) Unlike in the Exercise 1c of Assignment 3, a more practical way exists in building the $\underline{\underline{L_h}}$ in a sparse form using row & column indices and the element value. The command `sparse(ii, jj, aa)` in MATLAB serves exactly to this purpose, with `ii` and `jj` representing the vectors containing the row and column indices of the non-zero elements and `aa` the value of those elements respectively, or $L_h(ii(k), jj(k)) = aa(k)$ with k being the integer in the range between 1 and the number of non-zero elements. The key advantage of `sparse` is the handling of indices in an arbitrary sequence within the aforementioned vectors, such that the stencil coefficients can be written in lexicographically form one under the other at any arrangement.

c) Please refer to online submitted `a04ex01_Lh5.m` file.

d & e) The exercises d and e are combined in in the online submitted `a04ex01_solve.m` file. The function is called with a single input parameter `FLAG` that refers to the form of the numerical differentiation with

- `FULL`: Solves the problem with full matrix,
- `RED`: Solves the problem with reduced differentiation matrix and accordingly modified right hand side.

The solutions of the boundary value problems $L_h u_h = \sin(\pi x)\sin(\pi y)$ and $L_h u_h = \sin(\pi x + \pi/8)\sin(\pi y)$ with homogeneous Dirichlet boundary conditions utilizing either differentiation forms was simulated with 512×512 points. The result is illustrated in Figure 1, where Figure 1a is showcasing a zero condition on the entire boundary, and Figure 1b is showcasing a non-zero condition on two edges of the boundary. It reveals that the error in both methods remain same, whereas the number of DOFs dropped significantly in the reduced method, thus nearly halving the process duration in the former problem, whereas reducing the process duration by 61% in the latter problem.

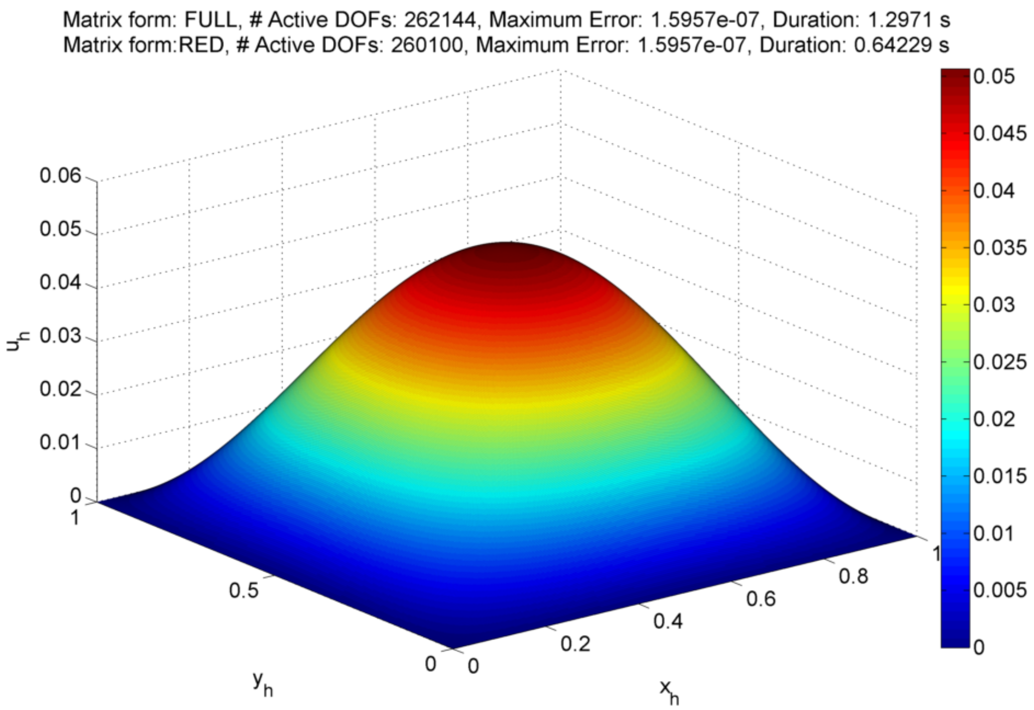
Exercise 2

Given is the boundary value problem

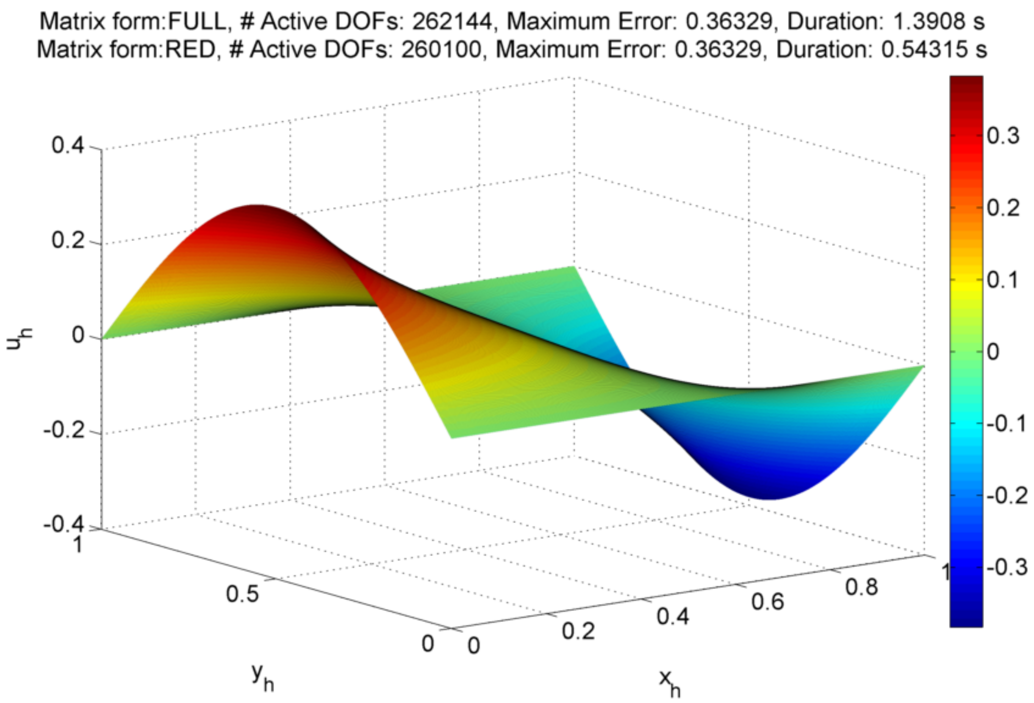
$$-au''(x) + bu'(x) + cu(x) = f(x), \ x \in (0,1), \tag{4}$$
$$u(0) = \alpha, \ u(1) = \beta \tag{5}$$

for a non-uniform mesh $\bar{\Omega}_h$ with constants $a, b, c, d, \alpha, \beta \in \mathbb{R}$.

a) The right handside f_h is now defined by the restriction of f to the grid points, which now can be non uniform, and thus we can better write $f_h = f_x = (f(x_1), ..., f(x_N))^T$.



(a) $L_h u_h = \sin(\pi x)\sin(\pi y)$



(b) $L_h u_h = \sin(\pi x + \pi/8)\sin(\pi y)$

Figure 1 *Solutions of two different* $L_h u_h = f(x,y)$ *problems with remarks regarding the active dof number, maximum error* $\max_{x,y\in\Omega_h}|u_h(x,y) - u(x,y)|$ *and process duration for both flags. Since the error is same in both flags, the plot of the latter flag is omitted and the remark title is added instead.*

We then have to add the vector

$$(\alpha(a\frac{2}{h_0(h_0+h_1)} - b\frac{-1}{h_0+h_1}), 0, .., 0, \beta(a\frac{2}{h_{N+1}(h_N+h_{N+1})} - b\frac{1}{h_0+h_1}))^T$$

If we use the D^+ stencil the middle summand of the expression of L_h turns into:

$$b(0, \frac{-1}{h_{i+1}}, \frac{1}{h_{i+1}})$$

with right hand side a sum of f_h and the boundary term:

$$(\alpha(a\frac{2}{h_0(h_0+h_1)}), 0, .., 0, \beta(a\frac{2}{h_{N+1}(h_N+h_{N+1})} - b\frac{1}{h_{N+1}}))^T$$

If we use D^-:

$$b(\frac{-1}{h_i}, \frac{1}{h_i}, 0)$$

with right hand side a sum of f_h and the boundary term:

$$(\alpha(a\frac{2}{h_0(h_0)} - b\frac{-1}{h_0+h_1}), 0, .., 0, \beta(a\frac{2}{h_{N+1}(h_N+h_{N+1})}))^T$$

b) Please refer to the online submitted `a04ex02getPDE.py` file. Figure 2 shows the graph produced if the last section of the program is uncommented, which is the analytical solution from a03ex03 restricted to a uniform grid compared to the numerical solution on a uniform grid vector transformed by the function $T(x_i) = x_i^2$. We can see that the two curves are quite similar to each other, and the difference grows smaller the bigger the number of grid points.

Exercise 3

a) Please refer to `a04ex03solve(eps,xh,flag)` function in the online submitted `a04ex03experiment.py` file.

b) Please refer to `a04ex03error(eps,xh,uh)` function in the online submitted `a04ex03experiment.py` file.

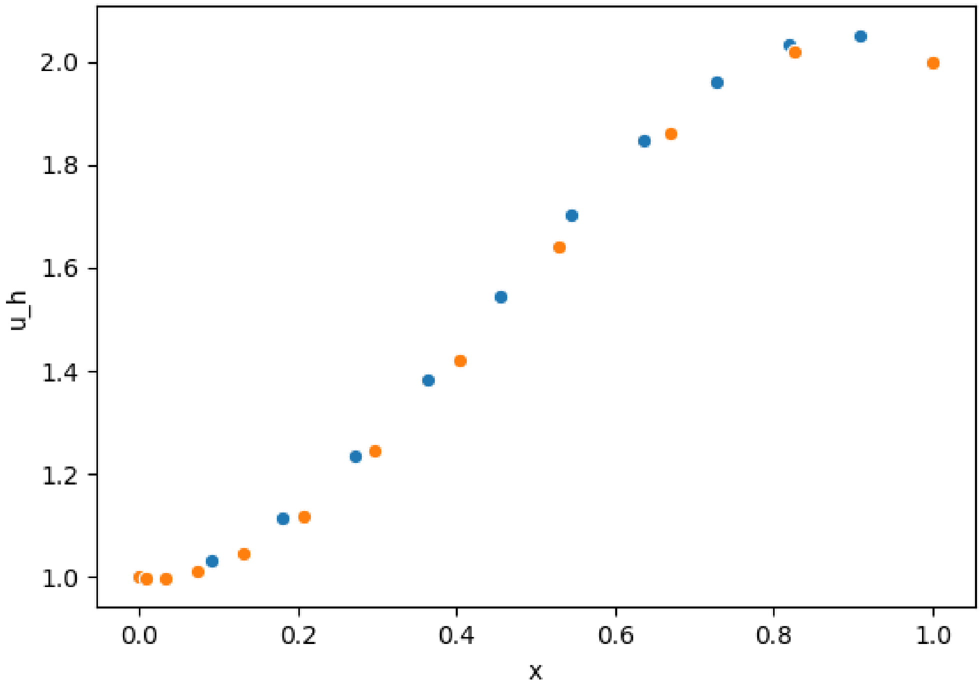


Figure 2 *Solution of the BVP with uniform and non-uniform meshes*

c) Please refer to `a04ex03shiskin(N, sigma)` function in the online submitted `a04ex03experiment.py` file.

d) Please refer to online submitted `a04ex03experiment.py` file.