



# Scientific Computing I

(Wissenschaftliches Rechnen I)

Winter term 2019/20

Priv.-Doz. Dr. Christian Rieger

Fabian Hoppe



## 8<sup>th</sup> exercise sheet

Submission on December 5, before the lecture

### Exercise 1.

(1 + 4 = 5 points)

Let  $\Omega \subset \mathbb{R}^d$  be a polygonal domain and  $\mathcal{T}_h$  a family of non-degenerate quasi-uniform tetrahedral triangulation of  $\Omega$  such that there is  $N \in \mathbb{N}$  (independent of  $h$ ) such that every node of the triangulation belongs to at most  $N$  tetrahedra of  $\mathcal{T}_h$ .

For piecewise linear finite elements on the triangulation  $\mathcal{T}_h$  we define the stiffness and mass matrices  $\mathbf{A}_h, \mathbf{M}_h \in \mathbb{R}^{N_h \times N_h}$  as usual by

$$(\mathbf{A}_h)_{i,j} := \int_{\Omega} \nabla \varphi_i^h \nabla \varphi_j^h dx \quad \text{and} \quad (\mathbf{M}_h)_{i,j} := \int_{\Omega} \varphi_i^h \varphi_j^h dx,$$

respectively, where  $(\varphi_i^h)_{i=1,\dots,N_h}$  denotes the nodal basis of the finite element space  $V_h \subset H_0^1(\Omega)$  with respect to those nodes of the triangulation  $\mathcal{T}_h$  not being part of the boundary of  $\Omega$ .

- How many non-zero entries does every row/column of  $\mathbf{A}_h$  contain at most?
- Show that there are constants  $c_1, c_2 > 0$  (independent of  $h$ ) such that it holds

$$c_1 \leq \frac{\mathbf{x}^T \mathbf{A}_h \mathbf{x}}{\mathbf{x}^T \mathbf{M}_h \mathbf{x}} \leq c_2 \left( \min_{T \in \mathcal{T}_h} \text{diam}(T) \right)^{-2},$$

and conclude that  $\text{cond}_2(\mathbf{A}_h) = \mathcal{O}(h^{-2})$ .

Hint: As an auxiliary result you may first show that it holds for any  $\phi_h \in V_h$ :

$$|\phi_h|_{H^1(\Omega)} \leq c \left( \min_{T \in \mathcal{T}_h} \text{diam}(T) \right)^{-2} \|\phi_h\|_{L^2(\Omega)}.$$

Further, use exercise 4 from sheet 7.

**Exercise 2.**

(3 + 2 = 5 points)

On  $\Omega = [0, 1]$  we consider

$$\begin{aligned} -\operatorname{div}(a_n(x)\nabla u(x)) &= 1 && \text{on } \Omega, \\ u(0) &= 0, \\ u(1) &= 0, \end{aligned}$$

with  $a_n(x) := (2 + \cos 2\pi nx)^{-1}$ . For every  $n \in \mathbb{N}$  the unique solution  $u = u_n \in H_0^1(\Omega)$  is given by

$$u_n(x) = x - x^2 + \frac{1}{n} \left( \frac{1}{4\pi} \sin 2\pi nx - \frac{1}{2\pi} x \sin 2\pi nx - \frac{1}{4n\pi^2} \cos 2\pi nx + \frac{1}{2n\pi^2} \right).$$

Let  $u_n^h$  be the Galerkin approximation to  $u_n$  obtained with piecewise linear finite elements on a uniform partition of  $\Omega$  into subintervals of length  $h > 0$ ,  $h^{-1} \in \mathbb{N}$ .

- a) Derive an estimate for  $\|u_n^h - u_n\|_{H^1}$  in terms of  $h$  and  $n$ . What can you conclude about a suitable choice of the mesh size  $h$ ?

Hint: Use Céa's Lemma.

- b) Does the sequence  $(u_n)_n$  converge (weakly/strongly) in  $L^2(\Omega)$  resp.  $H^1(\Omega)$  for  $n \rightarrow \infty$ ?

**Exercise 3.**

(2 + 3 = 5 points)

Let  $\Omega \subset \mathbb{R}^d$ ,  $d = 2, 3$ , be a polygonal domain and  $\mathcal{T}_h$  a family of non-degenerate quasi-uniform triangulations of  $\Omega$ . With  $V_h$  we denote the space of piecewise linear finite elements on  $\Omega$  with respect to  $\mathcal{T}_h$ .

- a) Show that there is a constant  $c > 0$  (independent of  $h$ ) such that

$$\|v_h\|_{L^\infty(\Omega)} \leq \begin{cases} ch^{-1} \|v_h\|_{L^2(\Omega)} & d = 2 \\ ch^{-3/2} \|v_h\|_{L^2(\Omega)} & d = 3 \end{cases} \quad \forall v_h \in V_h.$$

- b) Assume further that  $\Omega$  is convex. Prove that the Galerkin approximation  $u_h \in V_h$  to the problem

$$\begin{aligned} -\Delta u &= f, && \text{on } \Omega, \\ u &= 0, && \text{on } \partial\Omega, \end{aligned} \tag{1}$$

with  $f \in L^2(\Omega)$  satisfies the following  $L^\infty$ -error estimate:

$$\|u - u_h\|_{L^\infty(\Omega)} \leq \begin{cases} ch \|f\|_{L^2(\Omega)} & d = 2 \\ ch^{1/2} \|f\|_{L^2(\Omega)} & d = 3 \end{cases}$$

Hint: You may use without proof that due to convexity of  $\Omega$  problem (1) is  $H^2$ -regular, i.e. it admits a unique solution  $u \in H_0^1(\Omega) \cap H^2(\Omega)$  satisfying an estimate  $\|u\|_{H^2(\Omega)} \leq c_\Omega \|f\|_{L^2(\Omega)}$ . Further, note that due to  $H^2(\Omega) \hookrightarrow C(\bar{\Omega})$  the nodal interpolation  $\mathcal{I}_h : H^2(\Omega) \rightarrow V_h$  is welldefined.

## Programming exercise 1.

(0 + 0 + 5 = 5 points)

As we have seen in the last two programming exercises, implementing the finite element method from scratch is quite lengthy and inefficient, even for rather simple examples. This is the reason why we will work with an existing finite element software library, the so called FEniCS (Finite Element Computing System) Project from now on.

- a) Go to <https://fenicsproject.org/> → *Download* → *FEniCS on Anaconda* and follow the instructions to install FEniCS in an Anaconda environment. This provides the easiest way to install FEniCS.

After installation has finished enter the command

```
conda activate fenicsproject
```

to activate the newly created Anaconda environment called “fenicsproject”, that contains your installation of FEniCS. Now, enter

```
conda install -c conda-forge mshr
```

to install mshr, the mesh-generation tool of FEniCS. Further you might install matplotlib separately by running

```
pip install matplotlib
```

Once this has been completed, test whether your installation was successful: Start a python interpreter within the Anaconda environment fenicsproject and import FEniCS and mshr by typing:

```
from fenics import *
from mshr import *
```

Most likely, this will not work in jupyter notebook, so we will directly call python from the terminal.

- b) Once you have installed FEniCS and mshr, run the python file provided on the website: Open the terminal and enter the command

```
source activate fenicsproject
```

to start the Anaconda environment containing the FEniCS-installation. Then run the file by

```
python3 pathofyourfile/nameofthefile.py
```

and try to understand what the different parts of the code do and how they work. In this file the FE solution of the equation

$$\begin{aligned} -\frac{1}{10}\Delta u + u &= f && \text{on } [0, 1]^2, \\ \partial_n u &= 0 && \text{on } \partial[0, 1]^2 \end{aligned}$$

with  $f(x, y) = \mathbf{1}_{\{(x-1/2)^2+(y-1/2)^2<1/9\}}$  and piecewise linear finite elements is implemented.

On <https://fenicsproject.org/documentation/> you may find some helpful information.

c) Modify the given python file in order to solve the following problem:

$$\begin{aligned} -\operatorname{div}(a\nabla u) &= 1 && \text{on } \Omega, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

on  $\Omega = [0, 1] \times [0, 2] \cup [0, 2] \times [0, 1]$  with  $a(x, y) = 1 + 9 \cdot \mathbf{1}_{\{(x-1/2)^2 + (y-1/2)^2 < 1/9\}}$ .

Note that an `Expression`-object for  $a$  can be defined in completely the same way as for  $f$  and that the factor `0.1` in the bilinear form from b) can be replaced by such an `Expression` object. For the incorporation of different geometries and Dirichlet boundary conditions, respectively, see the documentation/tutorial of FEniCS.

For the solution of the problem, first use piecewise linear finite elements, and second, use piecewise quadratic finite elements. Note that you need to change only one line of your code to switch from piecewise linear to piecewise quadratic functions.

Please submit the programming exercise til December 12, before the lecture, **directly to your tutor** via Email.