Date December 20th, 2016 Due February 6th, 2017, 12 noon

Programming Exam

1 In a nutshell

Write a finite element code in your preferred programming language solving one of the following boundary value problems:

$$\begin{aligned} -\Delta u &= \sin(\pi x)\sin(\pi y) && \text{in } \Omega \\ u &= 0 && \text{on } \partial\Omega \end{aligned} \qquad \begin{aligned} -\Delta u + u &= \cos(\pi x)\cos(\pi y) && \text{in } \Omega \\ \partial_n u &= 0 && \text{on } \partial\Omega \end{aligned}$$

Here, $\Omega = [0,1]^2$. Note that in both cases, the solution is a multiple of the right hand side.

2 Minimal requirements

The minimal acceptable version uses a regular mesh, dividing the square Ω into a uniform mesh of n by n (n can be changed either easily in the code or as a runtime parameter) smaller squares. These may or may not be divided into triangles according to your choice.

On this mesh, define a finite element space by linear or bilinear basis functions. Compute the matrix and right hand side vector either by numerical quadrature or by exact integration. Note that the BVP on the left requires that you deal with the boundary condition in some way, while the one on the right has a slightly more complicated matrix. Choose according to your preference.

Choose whatever linear algebra subroutines you like, for instance standard matrices and the backslash operator in Matlab, libraries like LAPACK, Eigen, or UMFPack (Suitesparse) in C or C++.

Generate a graphical output of your solution and verify that it approximates a multiple of the right hand side.

2.1 Rules

- You can prepare the assignment by yourself or in a team of two students.
- Only programs which compile and run are considered. There is no partial credit for a program not producing results.
- Program must be handed in electronically by Feb

- In a short oral interview the authors of each program have a chance to describe its features and their design decisions in order to verify authorship and help determining a grade. You will have to demonstrate the running program, possibly with a change of data.
- Please submit at the oral test a signed declaration: "I/We have prepared the assignment myself/ourselves and I/we have only used the sources declared in comments to the program".
- In the hopefully unlikely event that you do not pass with your program, it counts as one exam attempt. You can choose between one of the two written exams for your second attempt.

3 Options for good grades

- Well readable, modular program structure
- Meaningful tests for correctness
- Fulfilling the minimal requirements exceedingly well
- Compute the error $u - u_h$ and show that it converges as expected on a sequence of meshes with increasingly smaller mesh size h.
- Implement an a posteriori error estimator and evaluate its performance (reliability/efficiency)
- Implement higher order polynomials
- Implement your own matrix class and a conjugate gradient method to solve the discrete linear system.
- Write code for arbitrary triangular meshes and use meshes for instance generated by the program "triangle"