



Project-oriented Software Lab

Project 1 — Poisson's Equation

Presentation due on: Fri. May 24 2013

Exercise 1:

Consider Poisson's equation with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} -\Delta u(x, y) &= f(x, y), & (x, y) \in \Omega = [0, 1]^2 \\ u(x, y) &= 0 & (x, y) \in \partial\Omega. \end{aligned} \quad (1)$$

- Derive a variational formulation of this problem with trial and test space $H_0^1(\Omega)$.
- What assumptions must be made on u and f for the problem (1) to be well-posed? Compare this to the assumptions required for the variational problem.
- When is a solution of the variational problem also a solution of the boundary value problem (1), and vice versa?
- Find f such that $u(x, y) = \sin(2M\pi x) \sin(2N\pi y)$ satisfies (1) for $M, N \in \mathbb{N}$, $(x, y) \in \Omega$.
- Complete the provided code skeleton under the use of the tutorial on the HiFlow³ homepage to solve the variational problem with linear finite elements on a grid with sixteen elements. Inspect the solution with Paraview. Explain what happens.
- Change the tutorial code to account for another right-hand-side f satisfying

$$u(x, y) = 16x(1-x)y(1-y), \quad (x, y) \in \Omega.$$

Exercise 2:

It is important to try to verify that the solver works correctly. Even if the computed solution “looks correct”, there might be subtle errors in the computation process. In this exercise, you will make some experiments to assure yourself that your program is working correctly.

-Please turn over!-

- (a) The L^2 norm and H^1 seminorm of the error between the computed and the analytical solutions are already implemented in the software tutorial. Add them to your program and compute additionally the H^1 norm! How do you expect these errors to depend on the elements' diameter h and their polynomial degree p ?
- (b) Add a loop that performs uniform mesh refinement and solve the problem at several refinement levels. Compute the errors at each refinement level and compare with your answer to the previous question. Plot the errors in the two norms as functions of the number of degrees of freedom, using for instance MATLAB, Octave or gnuplot.
- (c) Repeat the previous task for different polynomial orders. How does the convergence rate change? How fast does the solution converge if the mesh is kept fixed and polynomial degree is increased (p -FEM)? What behavior do you expect for the different implemented right-hand-sides?
- (d) Investigate the condition number of the matrix by exporting it to MATLAB / Octave and using the function `cond()` or `condest()`. How does the condition number depend on the level of refinement and the polynomial degree?
- (e) How does the number of iterations in the CG method depend on the condition number of the matrix? Verify this experimentally using your results from the previous task.

Exercise 3:

It is easy to forget the theoretical foundations of the computations that one performs. In this exercise, you will see that making even a minor change to the problem formulation can have large consequences on the outcome.

- (a) How does the variational problem change if you replace the Dirichlet boundary condition with an homogeneous Neumann boundary condition: $\partial_n u(x, y) = 0, (x, y) \in \partial\Omega$? What can be said about existence and uniqueness of the solution for this variational problem?
- (b) Modify the solver to solve the Neumann problem instead. Experiment with different initial solutions in the linear solver. What do you observe?
- (c) Implement a problem with mixed boundary conditions. Homogeneous Dirichlet boundary conditions $u(x, y) = 0, (x, y) \in \partial\Omega_{1,2}$ and homogeneous Neumann boundary conditions $\partial_n u(x, y) = 0, (x, y) \in \partial\Omega_{3,4}$.