SF2561 Finite Element Methods
Lab PM

Johan Hoffman

September 5, 2014

1 Introduction

The lab is divided in two parts: the first part A concerns implementation of the finite element method (FEM) in 1D, and part B concerns FEM approximation in 2D.

The purpose of the lab is to get hands-on experience of implementing a FEM program, and also to get familiar with a set of basic partial differential equations (PDEs) and their FEM approximation.

2 Examination

Labs can be done individually or in groups of two. Each group should hand in two compulsory written reports: Lab Report A and Lab Report B.

- Deadline Lab Report A: Friday October 3

- Deadline Lab Report B: Friday October 17

Part A and B consist of a set of compulsory problems, and a set of non-compulsory problems that give bonus points for the written exam if submitted in time for the deadline (submissions after the deadline give no bonus points). The lab is graded by pass/fail (P/F), and maximally 5 bonus points can be obtained for the lab.

2.1 Format of submissions

Each group member should submit individual Lab Reports by email to the teaching assistant (ncde@kth.se) in time for the deadline, containing the following:

1. A written report in pdf-format including a first page with: name, email address and educational program for all group members. The report should answer the questions of the lab specified in this document.

2. A one page summary of the work in the report that describes how the work was carried out:

   - Describe how you distributed the work in the group. Did you split the work? If so, who did what?

     (It is allowed to divide the work, and the source code can be identical for all group members. But the text in the individual reports should be written individually by each group member and should not be copied from each other.)

   - What resources were used to complete the lab assignment (books, websites, etc.)?

     (It is allowed to use any resources you find appropriate, but copying of existing code on the internet or from another group is not allowed.)

   - Who did you collaborate with apart from your group member?

     (It is allowed to collaborate with other groups, but all code and all text in the report should be written independently by the group members, copying code or text from another groups is not allowed.)

   - Describe what you found easy and hard, respectively, with the assignment?

3. A tar-archive of the source files needed to test the program and reproduce the results in Matlab/Octave.

Lab Part A

Consider the Poisson equation in 1D on the unit interval:

$$-u''(x) = f(x), \quad x \in (0,1) \tag{1}$$

with homogeneous Dirichlet boundary conditions

$$u(0) = u(1) = 0 \tag{2}$$

and a force $f(x) = 4\pi^2 \sin(2\pi x)$, with exact solution $u(x) = \sin(2\pi x)$.

Problem A.1 (compulsory)

Derive a weak formulation the Poisson equation (1) [CDE 8.1], and state the FEM method. Write a Matlab/Octave program that computes a finite element approximation U_h with the given boundary conditions and force term, using linear Lagrange basis functions [CDE 5.1] λ_i, $i = 1, 2$, and a uniform subdivision (mesh) $\mathcal{T}^h = \{x_1, x_2, ..., x_{N+1}\}$ of the unit interval, with N subintervals (cells) $I_i = (x_i, x_{i+1})$ of length (mesh size) $h_i = x_{i+1} - x_i$, with $h_i = h = 0.1$ constant. To compute the element integrals $\int_{x_i}^{x_{x+1}} \lambda_i'(x)\lambda_j'(x) \, dx$ and $\int_{x_i}^{x_{x+1}} \lambda_i'(x)f(x) \, dx$, use two different quadrature rules: (i) the composite midpoint rule [CDE 5.5.2], and (ii) the composite two-point Gauss quadrature rule [CDE 5.5.4]. Compare the results using (i) and (ii).

Problem A.2 (compulsory)

Modify the Matlab/Octave program in Problem A.1 to solve equation (1) with a combination of Dirichlet and Neumann boundary conditions:

$$u(0) = u'(1) = 0 \tag{3}$$

and a force $f(x) = (9/16) * 4\pi^2 \sin((3/4) * 2\pi x)$.

Hint: The Matlab interpolation function `interp1` may be useful.

Hint: A typical FEM program takes the following form:

```
% Generate the mesh
nodes = ...
elements = ...

% Construct the local to global map
loc2glob = ...

% Assemble matrix A
for e=1:number_of_elements
  for i=1:number_of_test_functions
    for j=1:number_of_trial_functions
      for q=1:number_of_quadrature_points
        local_element_matrix = ...
      end
      A(loc2glob(e,i),loc2glob(e,j)) += local_element_matrix
    end
  end
end

% Assemble vector b
for e=1:number_of_elements
  for i=1:number_of_test_functions
      for q=1:number_of_quadrature_points
        local_element_vector = ...
      end
      b(loc2glob(e,i)) += local_element_vector
  end
end

% Apply boundary conditions
...

% Solve linear system of equations Ax=b
x=linsolve(A,b)
```

Problem A.3 (1 bonus point)

Compute the L_2-norm [CDE 4.6.1] of the error $e_h = u - U_h$ in Problem A.1 for different uniform mesh sizes $h = \{2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}\}$. To compute the L_2-norm use, e.g., the composite midpoint quadrature rule [CDE 5.5.2] on a reference mesh with a fine resolution $h^{ref} = 10^{-5}$ and N^{ref} number of intervals I_i^{ref}:

$$\|e_h\| = \left(\int_0^1 |e_h|^2 \, dx \right)^{1/2} \approx \left(\sum_{i=1}^{N^{ref}} |e_h(x_m^i)|^2 \, h_i^{ref} \right)^{1/2} \tag{4}$$

with $x_m^i = (x_i + x_{i+1})/2$, the midpoint of interval I_i^{ref}.

- Study the order of convergence of the approximation U_h with respect to the mesh size, by plotting the L_2-norm of the error against the uniform FEM mesh size h in a log-log plot.

- Verify the theoretical second order convergence [CDE 15.5.3]. That is, show that $\|e_h\| \leq Ch^p$ with $p = 2$.

- Plot the error e_h over the fine reference mesh, what can you say about the structure of the error? [CDE Problem 8.9]

Lab Part B

Part B is based on the FEM program Puffin described in the last section of this Lab PM. Consider Poisson's equation in 2D:

$$-\Delta u(x) = f(x), \quad x \in \Omega, \tag{5}$$

with $x = (x_1, x_2)$ and $\Omega = [0, 1] \times [0, 1]$ the unit square, with homogeneous Dirichlet boundary conditions:

$$u(x) = 0, \quad \forall x \in \partial\Omega. \tag{6}$$

This equation is implemented in Puffin as two m-files: `PoissonSolver.m` and `Poisson.m`. Let the two meshes `square.m` and `square_refined.m` be denoted by $\mathcal{T}^1$ and $\mathcal{T}^2$, respectively. To use the solver in Matlab/Octave:

```
>> PoissonSolver
```

Problem B.1 (compulsory)

1. Write down a FEM method for the Poisson problem using piecewise linear Lagrange basis functions [CDE 15.1], then solve equation (5)-(6) using Puffin for two different forces f_1, f_2, given by:

$$f_1(x) = 32x_1(1 - x_1) + 32x_2(1 - x_2), \qquad (7)$$
$$f_2(x) = 20\pi^2 \sin(2\pi x_1) \sin(4\pi x_2), \qquad (8)$$

corresponding to the two exact solutions u_1, u_2, given by

$$u_1(x) = 16x_1(1 - x_1)x_2(1 - x_2), \qquad (9)$$
$$u_2(x) = \sin(2\pi x_1) \sin(4\pi x_2). \qquad (10)$$

2. Plot the solution and (the linear interpolation of) the error using the two different meshes $\mathcal{T}^1$ and $\mathcal{T}^2$.

3. For the two different forces f_1, f_2 the error is different. Which one is largest? Explain why [CDE Theorem 15.7].

4. How does the error change when the mesh is refined? Explain why [CDE Theorem 15.7].

Problem B.2 (compulsory)

1. On the parts of the boundary where $x_1 = 0$ and $x_1 = 1$, respectively, change the Dirichlet boundary conditions to instead homogeneous Neumann boundary conditions:

$$\frac{\partial u}{\partial n} = \nabla u \cdot n = \frac{\partial u}{\partial x_1} n_1 + \frac{\partial u}{\partial x_1} n_2 = 0, \qquad (11)$$

with $n = (n_1, n_2)$ the unit outward normal of the boundary. The force

$$f_3(x) = 5\pi^2 \cos(\pi x_1) \sin(2\pi x_2), \qquad (12)$$

then corresponds to the exact solution

$$u_3(x) = \cos(\pi x_1) \sin(2\pi x_2). \qquad (13)$$

2. Plot the solution and (the linear interpolation of) the error using the two different meshes $\mathcal{T}^1$ and $\mathcal{T}^2$.

3. Describe how the error changes when the mesh is refined. Where in the domain is the error maximal?

Problem B.3.1 (1 bonus point)

Now consider the convection-diffusion-reaction equation:

$$\dot{u} - \epsilon \Delta u + \beta \cdot \nabla u + \alpha u = f, \quad (x,t) \in \Omega \times (0,T], \qquad (14)$$

with $x = (x_1, x_2)$, $\Omega = [0,1] \times [0,1]$, and initial solution $u(x,0) = u^0(x)$.

1. Write down a FEM method using piecewise linear basis functions in space and implicit Euler time stepping. The implicit Euler time discretization for each time interval $I_n = (t_{n-1}, t_n)$, with time step $k_n = t_n - t_{n-1}$, reads: Given $u(t_{n-1})$ find $u(t_n)$ such that

$$\frac{u(t_n) - u(t_{n-1})}{k_n} + \beta \cdot \nabla u(t_n) - \epsilon \Delta u(t_n) + \alpha(x)u(t_n) = f(t_n) \quad x \in \Omega \subset \mathbb{R}^2.$$

2. Solve the problem to final time $T = 0.25$, using a time step $k = 0.01$, on the mesh $\mathcal{T}^2$, using Puffin with: $\alpha(x) = f(x) = 0$, $\beta(x) = (0,0)$, $\epsilon = 1$, $u^0(x) = 16x_1(1-x_1)x_2(1-x_2)$, and use homogeneous Dirichlet boundary conditions $u(x) = 0$ for the whole boundary $\partial\Omega$.

   This corresponds to the *heat equation*: $\dot{u} - \Delta u = 0$, with homogeneous Dirichlet boundary conditions. Prove that

   $$\|u(T)\|^2 + 2\int_0^T \|\nabla u\|^2 \, dt = \|u^0\|^2, \quad \forall t > 0, \qquad (15)$$

   meaning that the L_2-norm of the solution $u(t)$ will decrease as time increases [CDE 16.3]. Can you see the same behavior in the computed FEM solution?

3. The energy balance (15) also says that $\|u(t)\|$ will decrease faster if the norm of the gradient of the solution $\|\nabla u\|$ is large. Is this true in the computation? Try with instead $u^0(x) = \sin(4\pi x_1)\sin(4\pi x_2)$. What do you see? Plot both solutions at final time.

4. Set $\alpha(x) = 0$, $\epsilon = 0.1$, $\beta(x) = 5(-(x_2-0.5), x_1-0.5)$, and compute the solution on the mesh $\mathcal{T}^2$, up to final time $T = 3.5$, using a time step $k = 0.05$, with homogeneous Neumann boundary conditions: $\partial u / \partial n = 0$, and with the source term:

   $$f(x,t) = 1, \quad \text{if } |x - (0.75, 0.5)| < 0.1 \text{ and } |t - round(t)| < 0.1,$$

   and $f(x,t) = 0$ else. Describe what you see.

   Hint: Use the other solvers in Puffin.

Problem B.3.2 (1 bonus point)

1. Solve the dual problem corresponding to a stationary convection-diffusion-reaction problem (14), that is without the time derivative $\dot{u}$:

$$-\epsilon\Delta\varphi - \beta\cdot\nabla\varphi + \alpha\varphi = \psi, \quad x\in\Omega, \tag{16}$$

with homogeneous Neumann boundary conditions: $\partial\varphi/\partial n = 0$, and source term

$$\psi(x) = 1, \quad \text{if } |x - (0.25, 0.25)| < 0.1,$$

and $\psi(x) = 0$ else.

2. For $\alpha(x) = 0$, $\epsilon = 0.1$ and $\beta(x) = 5(-(x_2-0.5), x_1-0.5)$, plot the dual solution $\varphi(x)$, and give an interpretation with respect to the error of the FEM solution of equation (14) [CDE 15.5].

Problem B.3.3 (1 bonus point)

1. Solve a stationary convection-diffusion-reaction problem (14) with $\dot{u} = 0$, $\alpha(x) = 0$, $\epsilon = 0.1$ and $\beta(x) = 5(-(x_2 - 0.5), x_1 - 0.5)$, on $\mathcal{T}^1$ and $\mathcal{T}^2$. Then solve the same problem, now with $\epsilon = 10^{-4}$. What is the difference in the solutions with the smaller ϵ? [CDE Problems 18.7 and 18.9].

2. Modify the method using least squares stabilization [CDE 18.3]. What changes?

Problem B.4.1 (1 bonus point)

A FEM mesh $\mathcal{T} = \{K\}$ is a sub-division of Ω into a non-overlapping set of *elements* (or *cells*) K, with diameter h_K. To preserve continuity over edges, no node (vertex) of one triangle can lie on the edge of another triangle: such nodes are called *hanging nodes*, see figure below.

When locally refining a FEM mesh it is important to preserve the condition of no hanging nodes. One such algorithm in 2D is the *red-green mesh refinement algorithm*.

1. Implement the red-green mesh refinement algorithm as a Matlab/Octave function in an m-file. The algorithms takes the form: Mark a number of cells for refinement, then
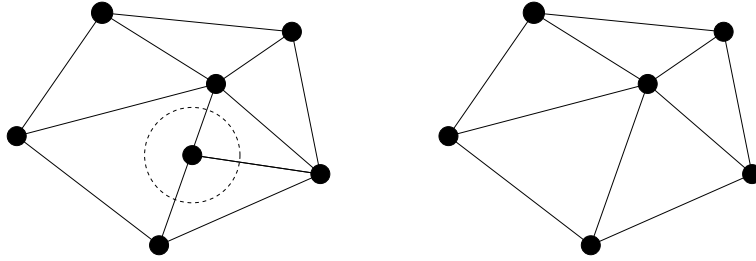
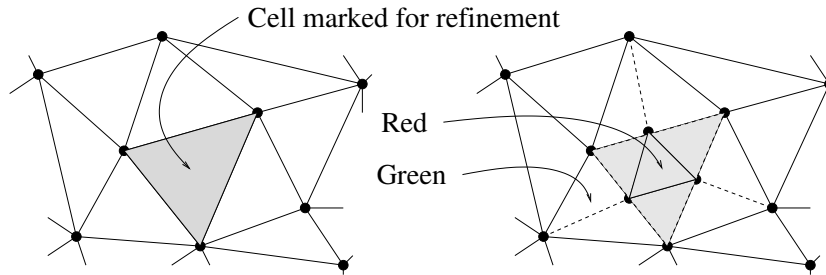Figure 1: To the left: a hanging node that lies on the edge on a triangle.

Figure 2: Illustration of red-green mesh refinement with one cell marked for refinement.

  (a) loop over all marked cells, for each cell subdivide the edges at the midpoints, which gives 4 new cells, and then

  (b) loop over all all hanging nodes, and connect each hanging node with the corresponding opposite nodes in each cell.

2. Illustrate the algorithm by 3 times refining the mesh $\mathcal{T}^1$, marking all cells for refinement with at least one node inside the circle defined by: all $x = (x_1, x_2)$ such that $(x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.05^2$. Plot the 3 refined meshes.

Problem B.4.2 (1 bonus point)

1. Compute the residual $R(U) = f + \Delta U$ for the solutions in Problem B.1. Use the approximation $\Delta U \approx \Delta_h U$, with $\Delta_h U$ the *discrete Laplacian* [CDE 15.1.4].

2. Plot (the linear interpolation of) the error, and compare the L_2-norm of the error, using 2 different mesh refinement algorithms:

   (a) 3 uniform mesh refinements (refine all cells 3 times).

   (b) 5 local mesh refinements with red-green mesh refinement, where in each step you refine 50% of the cells with the largest residual $R(U)$.

   How many nodes are used in each step of the algorithm in the two approaches? Which approach is the most efficient in terms of using as few nodes as possible to obtain as low error as possible? Plot the final meshes for 1 and 2.

Problem B.5 (1 bonus point)

1. Change the FEM basis in Puffin to using piecewise quadratic basis functions in space.

2. Solve equation (5)-(6) with the new basis on $\mathcal{T}^1$ and $\mathcal{T}^2$.

Problem B.6 (1 bonus point)

Use Puffin to solve an engineering problem of your choice. Implement the new equation in Puffin, and use a different mesh (not the unit square). To generate a new mesh you may for example use `Triangle`, available for download at:

   http://www.cs.cmu.edu/~quake/triangle.html

Puffin - a simple FEM solver

Puffin is a simple 2D FEM solver for Matlab/Octave in the form of: (1) two m-files for the assembly of a matrix and a vector (`AssembleMatrix.m` and `AssembleVector.m`), and (2) files describing the solution algorithm and the definition of the PDE in variational form, for example `PoissonSolver.m` and `Poisson.m`. To download and install Puffin:

1. Go to `https://launchpad.net/puffin`.

2. Download Puffin 0.1.6. to your working directory.

3. Unpack the files in `puffin-0.1.6.tar.gz`.

You find the m-files in the `src` directory:

```
$ cd puffin-0.1.6/src
```

2.2 Meshes

In Puffin the following data structures are used to represent a FEM mesh:

- `p` - coordinates (x_1, x_2) of the nodes

- `e` - edge information

- `t` - elements (triangles): global node numbers for local nodes 1,2,3.

- The last number is a subdomain numbering.

For most 2D problems in this lab we consider the computational domain $\Omega = [0, 1] \times [0, 1]$, the unit square.

- A uniform triangular mesh $\mathcal{T}^1$ of the domain Ω is available in Puffin as `square.m`.

- Refining $\mathcal{T}^1$ uniformly one level we get a new mesh $\mathcal{T}^2$ which is available in Puffin as `square_refined.m`.

2.3 Puffin tutorial

A good way to get familiar with Puffin is to complete the Puffin computer sessions F1-F5 at `http://www.bodysoulmath.org/sessions/`.