2D1260 Finite Element Methods Project Work

Johan Hoffman

October 18, 2006

1 Introduction

The project is divided into two parts; the first part concerns FEM approximation of Poisson's equation, and the second part concerns FEM for convection-diffusion-reaction equations. The purpose of the project is to get familiar with two important partial differential equations, and to gain an understanding of FEM approximation for these equations.

1.1 Puffin - a simple FEM solver

Puffin is a simple 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) a file describing the solution algorithm together with the definition of the PDE in variational form, for example `PoissonSolver.m` and `Poisson.m`.

Puffin is part of *FEniCS*, an open source software project for computational technology for solving differential equations. To download and install Puffin from the FEniCS homepage `www.fenics.org/puffin`:

1. Go to `www.fenics.org/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`

Puffin is a minimal implementation of the FEniCS FEM solver *DOLFIN*, which is available for download at `www.fenics.org/dolfin`. DOLFIN is

1

implemented in C++/Python and is suitable for large scale problems in 2D and 3D.

1.2 Puffin tutorial

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

1.3 Examination

The project can be done individually or in groups of two. Each group should hand in two written reports: for Project A and Project B, each including a first page containing: name, email address and educational program for all group members.

- Report A should be handed in no later than Friday September 22.

- Report B should be handed in no later than Friday October 20.

A combined grade 3-5 will be given for the projects A+B, which is part of the final grade of the course:

- For grade 5: Problem 1-4 have to be solved for both Project A and B.

- For grade 4: Problem 1-3 have to be solved for both Project A and B.

- For grade 3: Problem 1-2 have to be solved for both Project A and B.

1.4 Report layout

The reports should contain a description of each problem, a plot of the FEM solution of each problem, and (in an appendix) the matlab code used to solve the problem (only the new code written, not the existing Puffin code).

For plotting use `pdesurf` or `pdemesh`, where you can set the view by the command `view`. The standard 3D view is set by:

```
>> view(-37.5,30)
```

For the report use plots by `pdemesh`.

2

2　Meshes

In Puffin we use the following data structures to represent a finite element 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 details, see:
www.mathworks.com/access/helpdesk/help/toolbox/pde/ug/
index.html?/access/helpdesk/help/toolbox/pde/ug/pdetrg.html For

the project we consider the 2D 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`.

3　Project A

Consider Poisson's equation:

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

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

$$u(x) = 0, \quad \forall x \in \partial\Omega. \tag{2}$$

This equation is implemented in Puffin as 2 m-files: `PoissonSolver.m` and `Poisson.m`. To use the solver in Matlab/Octave:

```
>> PoissonSolver
```

3

3.1   Problem 1

Write down a FEM method for this problem using piecewise linear basis functions in space, then solve equation (1) using Puffin for 2 different forces f_1, f_2, given by:

$$f_1(x) = 32x_1(1 - x_1) + 32x_2(1 - x_2), \tag{3}$$

$$f_2(x) = 10\pi^2 \sin(\pi x_1) \sin(2\pi x_2), \tag{4}$$

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

$$u_1(x) = 16x_1(1 - x_1)x_2(1 - x_2), \tag{5}$$

$$u_2(x) = 2\sin(\pi x_1) \sin(2\pi x_2). \tag{6}$$

Plot the solution and the error using the two different meshes $\mathcal{T}^1$ and $\mathcal{T}^2$.

3.2   Problem 2

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:

$$\frac{\partial u}{\partial n} = \nabla u \cdot n = \frac{\partial u}{\partial x_1} n_1 + \frac{\partial u}{\partial x_1} n_2 = 0, \tag{7}$$

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

$$f_3(x) = 5\pi^2 \cos(\pi x_1) \sin(2\pi x_2), \tag{8}$$

then corresponds to the exact solution

$$u_3(x) = \cos(\pi x_1) \sin(2\pi x_2). \tag{9}$$

Plot the solution and the error using the two different meshes $\mathcal{T}^1$ and $\mathcal{T}^2$.

3.3   Problem 3

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*: Mark a number of cells for refinement, then
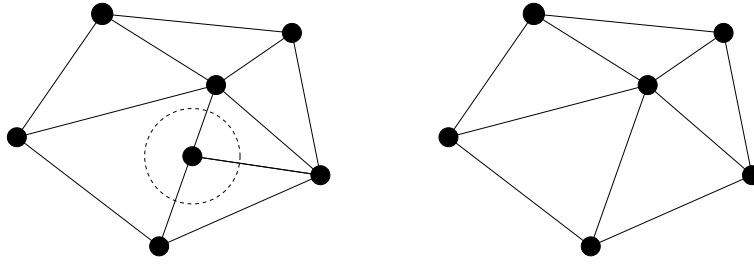
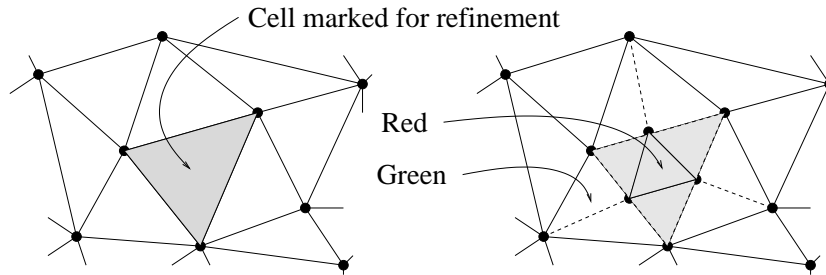Figure 1: To the left: a hanging node that lies on the edge on a triangle.

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

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

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

Implement the red-green mesh refinement algorithm as a Matlab/Octave function in an m-file. 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.

3.4   Problem 4

Compute the residual $R(U) = f + \Delta U$ for the solutions in Problem 1. Use the approximation $\Delta U \approx \Delta_h U$, with $\Delta_h U$ the *discrete Laplacian*, defined as in the pages 358-359 in the CDE book.

Plot the error, and compare the L_2-norm of the error, using 2 different mesh refinement algorithms:

1. 3 uniform mesh refinements (refine all cells 3 times).

2. 5 local mesh refinements, 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.

4   Project B

Now consider the convection-diffusion-reaction equation:

$$\dot{u} - \epsilon\Delta u + \beta \cdot \nabla u + \alpha u = f, \quad (x,t) \in \Omega \times (0,T], \qquad (10)$$

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

4.1   Problem 1

Set $\dot{u} = 0$, $\alpha(x) = 0.1$, $\epsilon = 0.01$,

$$\beta(x) = 5(-(x_2 - 0.5), x_1 - 0.5),$$

for the same uniform mesh $\mathcal{T}^1$ as in Project A.

Write down a FEM method for this problem using piecewise linear basis functions in space.

Now compute the stationary solution using the same uniform mesh $\mathcal{T}^1$ as in Project A, with homogeneous Neumann boundary conditions: $\partial u/\partial n = 0$, and with the source term:

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

and $f(x) = 0$ else.

A physical interpretation of this problem is that $u(x)$ is the concentration of a contaminant in the domain Ω with diffusivity ϵ, which experience stirring by $\beta(x)$, and where the contaminant is absorbed at a rate given by the absorption coefficient $\alpha(x)$.

4.2 Problem 2

Write down a FEM method using piecewise linear basis functions in space and implicit Euler time stepping (This is the method used in *F3: Problem 2*). 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

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

4.2.1 Problem 2.1

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

$$\|u(T)\|^2 + 2 \int_0^T \|\nabla u\|^2 \, dt = \|u^0\|^2, \quad \forall t > 0, \tag{11}$$

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

The energy balance (11) 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.

4.2.2 Problem 2.2

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. This is a time dependent variant of Problem 1. Describe what you see.

4.3 Problem 3

Solve the dual problem corresponding to Problem 1:

$$-\epsilon\Delta\varphi - \beta \cdot \nabla\varphi + \alpha\varphi = \psi, \quad x \in \Omega, \tag{12}$$

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.

Plot the dual solution $\varphi(x)$, and give an interpretation with respect to the error of the FEM solution in Problem 1 in the point $(0.25, 0.25)$.

4.4 Problem 4: Choose one of the following

4.4.1 Problem 4.1

Consider the stationary Problem 1. Compute the residual $R(U)$ of the approximate FEM solution. Refine the mesh using 3 different mesh refinement algorithms:

1. 3 uniform mesh refinements.

2. 5 local mesh refinements where in each step you refine 50% of the elements with largest residual $R(U)$.

3. 5 local mesh refinements where in each step you refine 50% of the elements with largest product $h^2 R(U) \times D^2\varphi$.

How many nodes are used in the 3 approaches? What are the differences in the refined meshes for approach 2 and 3?

4.4.2 Problem 4.2

Solve the stationary and time dependent problems (Problems 1 + 2.2) on $\mathcal{T}^1$ and $\mathcal{T}^2$, now with $\epsilon = 10^{-4}$. What happens with the solutions? Modify the method using least squares stabilization. What changes?

4.4.3 Problem 4.3

Change the FEM basis in Puffin to using piecewise quadratic basis functions in space. Solve the stationary and the time dependent problem (Problems 1 + 2.2) with the new basis on $\mathcal{T}^1$ and $\mathcal{T}^2$.

4.4.4   Problem 4.4

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