Project Work

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`:

1. Go to the *FEniCS* homepage, `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. `Puffin` is a minimal implementation of the *FEniCS* FEM solver *DOLFIN*, available for download at `www.fenics.org/dolfin`. *DOLFIN* is 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:
course code (DN2260), name, email address and educational program for all group members.
• Report A should be handed in no later than Friday, September 28.
• Report B should be handed in no later than Friday, October 19.
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: Problems 1-4 have to be solved for both Project A and B.
• For grade 4: Problems 1-3 have to be solved for both Project A and B.
• For grade 3: Problems 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 the `matlab pdetoolbox` functions `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 Meshes

`Puffin` (and the `Matlab PDE ToolBox`) uses the following data structures to represent a

triangular finite element mesh with `nel` elements and `nnodes` nodes:

- `p(2,nnodes)` - coordinates (x_1, x_2) of the nodes
- `e` - edge information
- `t(4,nel)` - elements (triangles): global node numbers for local nodes 1,2,3. The fourth number in `t(:,k)` is a subdomain numbering.

For details, see:
`www.mathworks.com/access/helpdesk/help/toolbox/pde/ug/index.html`
`www.mathworks.com/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.
- (`p,e,t`) of a uniform 2 x (8 x 8) triangular mesh $\mathbf{T}_1$ of the unit square is available in `Puffin` as `square.m`, `nel` = 128, `nnodes` = 81
- `square refined.m`. gives a uniformly refined mesh $\mathbf{T}_2$ from $\mathbf{T}_1$.

You can generate a mesh for a(nother) four-sided domain Q by:
0. Let `p` be the mesh generated by `square`;
1. Define the function $\mathbf{g}(\mathbf{x})$ which maps Ω onto Q;
2. Compute the coordinates of the mesh on Q by `p` := $\mathbf{g}$(`p`)`.`

3 Project A

Consider Poisson's equation:

$$-\Delta u(\mathbf{x}) = f(\mathbf{x}),\ \mathbf{x} \in \Omega, \qquad (3.1)$$

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

$$u(\mathbf{x}) = 0,\ \forall \mathbf{x} \in \partial\Omega. \qquad (3.2)$$

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

```
>> PoissonSolver
```

3.1 Problem A1

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

$$f_1(\mathbf{x}) = 32\, x_1\, (1 - x_1) + 32\, x_2\, (1 - x_2), \qquad (3.3)$$

$$f_2(\mathbf{x}) = 10\pi^2 \sin(\pi\, x_1) \sin(2\pi\, x_2), \qquad (3.4)$$

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

$$u_1(\mathbf{x}) = 16\, x_1\, (1 - x_1)\, x_2(1 - x_2), \qquad (3.5)$$

$$u_2(\mathbf{x}) = 2 \sin(\pi\, x_1) \sin(2\pi\, x_2). \qquad (3.6)$$

Plot the solution and the error using the two different meshes $\mathbf{T}_1$ and $\mathbf{T}_2$.

3.2 Problem A2

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

$$\frac{\partial u}{\partial n} = \nabla u \cdot \mathbf{n} = \frac{\partial u}{\partial x_1} n_1 + \frac{\partial u}{\partial x_2} n_2 = 0 \qquad (3.7)$$

with $\mathbf{n} = (n_1, n_2)$ the unit outward normal of the boundary. The source function

$$f_3(\mathbf{x}) = 5\pi^2 \cos(\pi\, x_1) \sin(2\pi\, x_2), \qquad (3.8)$$

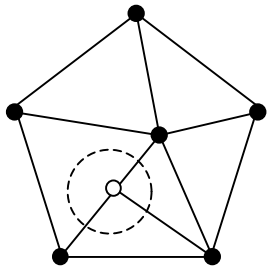
then corresponds to the exact solution

$$u_3\,(\mathbf{x}) = \cos(\pi\, x_1) \sin(2\pi\, x_2). \qquad (3.9)$$

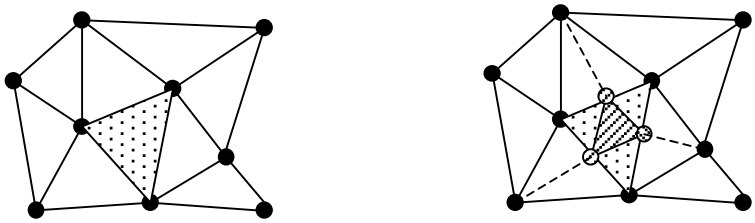
Plot the solution and the error using the two different meshes $\mathbf{T}_1$ and $\mathbf{T}_2$.

3.3 Problem A3

A FEM mesh $\mathbf{T} = \{K\}$ 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 right.



Local refinement of a FEM mesh must avoid hanging nodes. One such algorithm in 2D is the red-green mesh refinement algorithm:
Mark a number of cells for refinement, then
1. loop over all marked cells, for each cell create new nodes at the midpoints of its edges, which gives 4 new red cells, and then
2. loop over all all hanging nodes, and connect each hanging node with the corresponding opposite nodes in each cell. Note different cases dependent on how many edges have hanging nodes!
The new cells thus created are labeled green. The angles of (some of) the new triangles are about half their original size. If the refinement is repeated, unacceptably small angles may result. Therefore, the green cells should first be re-combined and then split into four if they have neighbors which should be refined.



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 $\mathbf{T}_1$, marking all cells for refinement with at least one node inside the circle defined by $(x_1-0.5)^2 + (x_2-0.5)^2 \leq 0.05^2$. Plot the 3 refined meshes, zoomed so the details of the mesh at the refinement front are clear.

3.4 Problem A4

Compute the residual $R(U) = f + \Delta U$ for the solutions in Problem A1. Use the approximation $\Delta U \approx \Delta_h U$, with $\Delta_h U$ the discrete Laplacian, defined as in CDE pp. 358-359. Plot the errors, and compare their L_2 -norms, using two 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, those 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:

$$\partial u / \partial t - \varepsilon \Delta u + \beta \cdot \nabla u + \alpha u = f, (\mathbf{x}, t) \in \Omega \times (0, T] \qquad (4.1)$$

with $\mathbf{x} = (x_1, x_2)$ and $\Omega = [0, 1] \times [0, 1]$, and initial solution $u(\mathbf{x}, 0) = u_0(\mathbf{x})$.

4.1 Problem B1

Set

$$\partial u / \partial t = 0,\ \alpha(\mathbf{x}) = 0.1,\ \varepsilon = 0.01,\ \boldsymbol{\beta}(\mathbf{x}) = 5(-(x_2 - 0.5), x_1 - 0.5),$$

for the same uniform mesh $\mathbf{T}_1$ as in Project A. Write down a FEM method for this problem using piecewise linear basis functions in space and compute the stationary solution with homogeneous

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

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

$= 0$ elsewhere.

A physical interpretation of this problem is that $u(\mathbf{x})$ is the concentration of a contaminant in the domain Ω with diffusivity ε and stirred by the flow velocity $\boldsymbol{\beta}(\mathbf{x})$. The contaminant source is at (0.75,0.5) and it is destroyed at a rate given by the absorption coefficient $\alpha(\mathbf{x})$. What sort of motion is described by $\boldsymbol{\beta}$? What is its divergence? Where does the flow enter viz. exit Ω?

4.2 Problem B2

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:

$$\frac{1}{k_n}\left(u(t_n) - u(t_{n-1})\right) + \beta \cdot \nabla u(t_n) - \varepsilon \Delta u(t_n) = f(t_n), \mathbf{x} \in \Omega \subset \mathbf{R}^2 \qquad (4.2)$$

4.2.1 Problem B2.1

Solve the problem to final time $T = 0.25$, using a time step $k = 0.01$, on the mesh $\mathbf{T}_2$, using `Puffin` with:

$\alpha(\mathbf{x}) = f(\mathbf{x}) = 0$, $\boldsymbol{\beta}(\mathbf{x}) = (0, 0)$, $\varepsilon = 1$, $u_0(\mathbf{x}) = 16\, x_1(1 - x_1)\, x_2\, (1 - x_2)$,

and homogeneous Dirichlet boundary conditions $u(\mathbf{x}) = 0$ for the whole boundary $\partial\Omega$. This corresponds to the heat equation:

$\partial u / \partial t - \Delta u = 0$,

with homogeneous Dirichlet boundary conditions. Prove the "energy estimate"

$$\left\|u(t)\right\|^2 + 2\int_0^t \left\|\nabla u\right\|^2 dt = \left\|u_0\right\|^2, \forall t > 0 \qquad (4.4)$$

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 estimate (4.4) also says that u will decrease faster if its gradient is large. Is this true in the computation? Try instead with

$u_0(\mathbf{x}) = \sin(4\pi\, x_1)\, \sin(4\pi\, x_2)$.

What do you see? Plot both solutions at the final time.

4.2.2 Problem B2.2

Set

$\alpha(\mathbf{x}) = 0$, $\varepsilon = 0.1$, $\beta(\mathbf{x}) = 5(-(x_2 - 0.5), x_1 - 0.5)$,

and compute the solution on the mesh $\mathbf{T}_2$, 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:

$f(\mathbf{x},t) = 1$, if $|\mathbf{x} - (0.75, 0.5)| < 0.1$ and $|t - \text{round}(t)| < 0.1$, $= 0$ else.

This is a time dependent variant of Problem B1. Describe what you see.

4.3 Problem B3

Solve the dual problem corresponding to Problem B1:

$$-\varepsilon\Delta\varphi - \beta \cdot \nabla\varphi + \alpha\varphi = \psi, \mathbf{x} \in \Omega \qquad (4.5)$$

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

$\psi(\mathbf{x}) = 1$, if $|\mathbf{x} - (0.25, 0.25)| < 0.1$, $= 0$ elsewhere.

Plot the dual solution $\varphi(\mathbf{x})$, and give an interpretation with respect to the error of the FEM solution in Problem B1 in the point (0.25, 0.25).

4.4 Problem B4

Choose one of the following

4.4.1 Problem B4.1

Consider the stationary Problem B1. Compute the residual $R(U)$ of the approximate FEM solution. Refine the mesh using three different mesh refinement algorithms:

1. 3 uniform mesh refinements.
2. 5 local mesh refinements where in each step you refine 50% of the elements, those with largest residual $R(U)$.
3. 5 local mesh refinements where in each step you refine 50% of the elements, those 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 B4.2

Solve the stationary and time dependent problems (Problems B1 + B2.2) on $\mathbf{T}_1$ and $\mathbf{T}_2$, now with $\varepsilon = 10^{-4}$. What happens with the solutions? Modify the method using least squares stabilization. What changes?

4.4.3 Problem B4.3

Change the spatial FEM basis in `Puffin` to piecewise quadratic polynomials. Solve the stationary and the time dependent problem (Problems B1 + B2.2) with the new basis on $\mathbf{T}_1$ and $\mathbf{T}_2$.

4.4.4 Problem B4.4

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`
The triangle can be mapped to another shape with three edges, like a sector of an ellipse. Extra credit for making meshes on unions of mapped squares and triangles (i.e. making unions of the mapped meshes ☺). You can also export the `(p,e,t)` from the `PDE ToolBox` mesh generator to the `matlab` workspace and use in `Puffin`. Figuring out how to set different boundary conditions on all edges, and different materials (heat conduction coefficients, etc) in different subdomains is a challenge. If you want to try your hand, get in touch with course administration for credits, etc.