Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Year 2008

The numerical simulation of complex PDE problems
P. Frey, M. De Buhan
MA691 & CC60X

TP `FreeFem++` N° 4

The finite element method (II)

All problems will be solved using `FreeFem++` software. The documentation can be found at `http ://www.freefem.org/ff++/ftp/freefem++doc.pdf`

Exercice 1 : A Poisson problem.

We start our study by considering the simple boundary-value Laplace problem posed in the unit domain $\Omega = [0,1]^2$:
Find u such that

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega = \bigcup_{1\leq i\leq 4} \Gamma_i \,. \end{cases} \tag{1}$$

0. Write the variational formulation of this problem.

The following `FreeFem++` script solves this problem in about 20 lines for Lagrange $\mathbb{P}_1$ finite elements, using a `LU` solver.

```
 //    Program for solving Poisson problem with homogeneous boundary conditions

int Nbnodes=10;                            //    number of nodes in x and y direction
mesh Th = square(Nbnodes,Nbnodes,[x,y]);        //    uniform mesh of the domain

func f= 1.0;                                                    //    source term

fespace Vh(Th,P1);                         //    FE space (Lagrange P1) defined on Th
Vh uh,vh;                                           //    function of the FE space Vh

problem Poisson(uh,vh,solver=LU)=     //    Definition of the variational problem
   int2d(Th) (dx(uh)*dx(vh)+dy(uh)*dy(vh))                        //    bilinear form
 - int2d(Th)(f*vh)                                                   //    linear form
 + on(1,2,3,4,uh=0);                                //    Dirichlet boundary condition

plot(Th,wait=1);                                                        //    draw mesh

Poisson;                                                     //    Numerical solution

plot(uh,wait=1);                                                   //    Plot solution
```

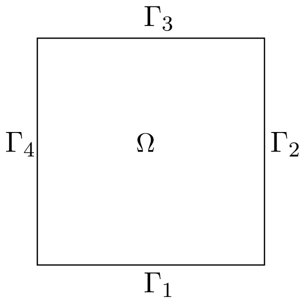

Fig. 1 – *The computational domain* Ω.

1. **Getting started.**
   Make sure you understand all command lines in the script (comments start with the sign //). In particular, notice how the mesh of the domain is defined (the boundary edges are labelled according in the counterclockwise order (Fig. 1), starting from the lower side of the square) and how the finite element approximation space is defined. The qualifier `solver=LU` is not required (by default a multi-frontal `LU` is used).
2. Change the script to use Lagrange $\mathbb{P}_2$ finite elements.
3. **Analysis of the convergence of the numerical solution.**
   Suppose the analytical solution is known for a given function f, sufficiently smooth. To this end, consider for example the solution $u(x,y) = x(1-x)y(1-y)$ associated with the function $f(x,y) = 2x(1-x) + 2y(1-y)$.
   a) Compute the numerical solution u_h of the problem (1) on various meshes corresponding to different levels of refinement, *i.e.* containing more and more grid points.
   b) Verify that $\|u - u_h\|_{L^2(\Omega)}$ and $\|\nabla u - \nabla u_h\|_{L^2(\Omega)}$ are then decreasing functions with respect to the mesh size.

   Hints :

   To compute the error, functions and integrals can be defined as :

```
func f=2.0*x*(1.0-x)+2.0*y*(1.0-y)                        //    function of x,y
func u=x*(1.0-x)*y*(1.0-y);                                //    exact solution
Vh error=u-uh;
                                                          //    L2 and H1 errors
real errL2 =sqrt(int2d(Th)(error^2));
real errH10=sqrt(int2d(Th)(dx(error)^2+dy(error)^2));
```

   it is also possible to define loops and print variabe values as follows :

```
for (int i=0; i<nbiter; i++){
  ... instructions
  cout << "L2 error "<< errL2 << endl;
};
```

   and a caption can be added to the plot of the solution :

```
string caption="L2 error"+err;
plot(error,wait=1,cmm=caption);
```

4. **Variations...**
   Find the variational formulation of the following Neumann and Fourier boundary value problems :
   Given f and g (non null), $\alpha \in \mathbb{R}_+^*$ fixed, find u solving :

$$\left\{ \begin{aligned} -\Delta u + u &= f && \text{in } \Omega \\ \frac{\partial u}{\partial n} &= 0 && \text{on } \partial\Omega\,. \end{aligned} \right. \qquad \text{and} \qquad \left\{ \begin{aligned} -\Delta u &= f && \text{in } \Omega \\ \alpha u + \frac{\partial u}{\partial n} &= g && \text{on } \partial\Omega\,. \end{aligned} \right. \tag{2}$$

   and modify the original `FreeFem++` script to solve this problem numerically on a unit disk.
   Hints :
   To define a unit disk and its discretization :

```
border Gamma(t=0,2*pi) { x=cos(t); y=sin(t); }          //    define the domain
mesh Th = buildmesh (Gamma(50));                   //    discretization of unit disk
```

   In the Fourier boundary-value problem, a boundary integral appears. It is defined in `FreeFem++` using the command :

```
int1d(Th,Gamma)( expression );
```

Exercice 2 A linear elasticity problem.

`FreeFem++` can be also used to solved vector problems. We consider a rectangular domain $\Omega =]0, L[\times]0, 1[$ representing a beam of length L that is fixed on its left boundary, denoted Γ_4 and is subjected to a volumic load $f \in L^2(\Omega)^2$. The corresponding boundary value problem is :
Find u solving :

$$-\operatorname{div} \sigma(u) = f \qquad \text{in } \Omega \tag{3}$$

with the boundary conditions :

$$u = 0 \quad \text{on } \Gamma_4 \qquad \text{and} \qquad \sigma(u)\cdot n = 0\,, \quad \text{on } \partial\Omega\backslash\Gamma_4\,,$$

where the stress tensor $\sigma(u)$ is defined as $2\mu e(u) + \lambda\,\mathrm{tr}(e(u))Id$ and $e(u) = \frac{1}{2}\left(\nabla u + \nabla u^t\right)$ represents the linearized strain tensor and λ and μ are the Lamé coefficients of the material (for example the coefficients of the concrete material are $\lambda = 1.86e6$ Pa, $\mu = 4.34e6$ Pa).

1. **Variational formulation.**
   Show that if $u \in H_0^2(\Omega)^2$ is solution of the problem (3), it is also a solution of the following variational problem :
   Find $u \in H_0^1(\Omega)^2$ such that

$$\int_\Omega \left(2\mu e(u) : e(v) + \lambda(\operatorname{div} u)(\operatorname{div} v)\right)\,dx = \int_\Omega f\cdot v\,dx\,, \qquad \forall v \in H_0^1(\Omega)^2\,. \tag{4}$$

2. **Finite element approximation.**
   We introduce a discretization of the domain Ω using a regular grid T_h. We consider the space of $\mathbb{P}_1$ polynomials with values in $\mathbb{R}^2$ and we introduce the space X_h of piecewise affine continuous functions on Ω :

$$X_h = \left\{v_h \in C^0(\Omega)^2\,,\ \ v|_K \in \mathbb{P}_1(K)\,, \forall K \in T_h\right\}\,.$$

(i) Show that the Galerkin approximation of problem (3) consists in finding

$$u_h = (u_h^1, u_h^2) \in V_h = \left\{ v_h = (v_h^1, v_h^2) \in X_h \,, \ \ v_h^1 = v_h^2 = 0 \text{ on } \Gamma_4 \right\} \,,$$

such that

$$\int_\Omega (2\mu e(u_h) : e(v_h) + \lambda (\operatorname{div} u_h)(\operatorname{div} v_h)) \, dx = \int_\Omega f \cdot v_h \, dx \,, \qquad \forall v_h \in V_h \,. \tag{5}$$

(ii) Write a script `FreeFem++` to solve the problem (5) on a mesh T_h for f, μ and λ arbitrarily fixed.

Hints :

Here, the finite element space will be defined as follows :

```
fespace Xh(Th,[P1,P1]);                              //    definition of FE space
Xh [uh1,uh2];                             //    initialisation of an element of Wh
```

where each element $u_h = (u_h^1, u_h^2)$ of the space X_h is represented by its two components. The variational formulation (5) can be developed as folows :

$$\int_\Omega 2\mu \left(\partial_x u_h^1 \partial_x v_h^1 + \partial_y u_h^2 \partial_y v_h^2 + (\partial_x u_h^2 + \partial_y u_h^1)(\partial_x v_h^2 + \partial_y v_h^1)/2 \right)$$
$$+ \lambda (\partial_x u_h^1 + \partial_y u_h^2)(\partial_x v_h^1 + \partial_y v_h^2) dx = \int_\Omega f^1 v_h^1 + f^2 v_h^2 dx \,.$$

3. **Visualization.**

   Plotting vector functions with `plot` command is not convenient. To overcome this problem, `FreeFem++` offers the possibility of plotting the deformation u of the mesh T_h using the function `movemesh` :

```
real scale=100.;
mesh Sh=movemesh(Th,[x+scale*uh1,y+scale*uh2]);
plot(Sh,wait=1);
```

   It is often more interesting to look at the Cauchy stress tensor $\sigma(u) = (\sigma_{ij}(u))$. Compute this stress tensor σ and plot it.

3. **Mesh adaptation.**

   The accuracy of the solution can be improved by refining the mesh (like we did in the previous exercise). However, this strategy is usually not optimal, as the refinement is uniform and does not account for local variations of the solution. The function `adaptmesh` allows to refine the mesh and adapt the node density to the approximation error. Modify the `FreeFem++` script to adapt the mesh to the solution with an accuracy related to the inverse of the `error` tolerance value.

   Hint :

   mesh adaptation is governed by the commands :

```
real error=0.001;
Th=adaptmesh(Th,uh1,uh2,err=error);
```

Exercice 3 The Stokes problem for viscous flows.

We consider the variational resolution of the Stokes equations posed in the domain $\Omega =]0,1[^2$:

$$\left\{ \begin{aligned} -\Delta u + \nabla p &= f \\ \operatorname{div} u &= 0 \end{aligned} \right. \quad \text{in } \Omega. \tag{6}$$

This set of equations models the flow of an incompressible viscous fluid (in the steady state). Notice that since p is defined by its gradient, it is determined only up to an additive constant. Actually, the discrete pressure can be uniquely determined by setting its value at any mesh node or by setting the constant with imposing its average value on the domain Ω, *i.e.* looking for $p \in L^2_0(\Omega) = \{q \in L^2(\Omega)\,,\ \int_\Omega q\,dx = 0\}$.

1. **Variational formulation.**

   Suppose the set of equations (6) is endowed with homogeneous Dirichlet conditions. Show that the problem (6) is equivalent to the following problem :
   Find $u \in H^1_0(\Omega)^2$ and $p \in L^2_0(\Omega)$ such that

$$\left\{ \begin{aligned} \int_\Omega \nabla u : \nabla v\,dx - \int_\Omega p \operatorname{div} v\,dx &= \int_\Omega f \cdot v\,dx\,, && \forall v \in H^1_0(\Omega)^2 \\ \int_\Omega q \operatorname{div} u\,dx &= 0\,, && \forall q \in L^2_0(\Omega) \end{aligned} \right. \tag{7}$$

2. **Finite element approximation.**

   We introduce the following approximation spaces :

$$\begin{aligned} X_h &= \{v_h \in C^0(\bar{\Omega})^2\,,\ v_h|_K \in \mathbb{P}_2\,,\ \forall K \in T_h\} \\ M_h &= \{q_h \in L^2(\Omega)\,,\ q_h|_K \in \mathbb{P}_0\,,\ \forall K \in T_h\} \end{aligned} \tag{8}$$

   and we want to solve the driven cavity problem for $f = 0$, where the Stokes equations are endowed with homegeneous boundary conditions for the velocity on $\Gamma_2 \cup \Gamma_3 \cup \Gamma_4$ and the velocity profile on the bottom edge defined as $u|_{\Gamma_1} \cdot n = 0$ and $u|_{\Gamma_1} \cdot t = -4x(1-x)$, where n and t denote the normal and the tangent to the boundary.

   (i) Write a script `FreeFem++` to solve the Stokes problem on a mesh T_h for f fixed using the approximation spaces defined above (hint : add a penalization term to allow the resolution using Crout solver).
   Hint :
   the variational formulation and the boundary conditions can be defined as :

```
real epsilon=0.00001;
problem Stokes([ux,uy,p],[vx,vy,q],solver=Crout)=\\
  int2d(Th)( ( dx(ux)*dx(vx) + dy(ux)*dy(vx)
...
               - epsilon*p*q )                           //    penalization term
+on(2,3,4,ux=0.,uy=0.)
+on(1,ux=4.*x*(1.0-x),uy=0.);

plot([ux,uy],wait=1);
```

(ii) Change the `FreeFem++` script to use other approximation spaces : Lagrange $\mathbb{P}_1$ for both the velocity and the pressure variables. Then with the extended $\mathbb{P}_1$ element for the velocity and Lagrange $\mathbb{P}_1$ for the pressure. Please comment on the results.