Project 6 – Numerics for Fluids, Structures and Electromagnetics
Pollution of a drinking water reservoir

Benoît Müller

Fall Semester 2021

Question 1

– The Darcy equation

The Darcy equation can be written $\begin{cases} \frac{1}{k}\mathbf{u} + \nabla p = 0 & \text{in } \Omega \\ \operatorname{div}\mathbf{u} = f & \text{in } \Omega \\ p = d & \text{in } \Gamma_D \\ \mathbf{u}\cdot\mathbf{n} = 0 & \text{in } \Gamma_N \end{cases}$

with $d = p_{\text{in}}\mathbb{1}_{\Gamma_4} + p_{\text{out}}\mathbb{1}_{\Gamma_2}$, $\Gamma_D = \Gamma_2 \cup \Gamma_4$ and $\Gamma_N = \Gamma_1 \cup \Gamma_3$ We search a mixed-form weak formulation for the Darcy equation by multiplying by a smooth functions test $\mathbf{v}$,p and integrate over Ω. We impose $\mathbf{v}\cdot\mathbf{n} = 0$ to stay in the space where $\mathbf{u}$ should be. We obtain with integration by part

$$\begin{aligned} 0 &= \int_\Omega (\frac{1}{k}\mathbf{u} + \nabla p)\mathbf{v} = \int_\Omega \frac{1}{k}\mathbf{u}\cdot\mathbf{v} + \int_\Omega \mathbf{v}\cdot\nabla p = \frac{1}{k}\int_\Omega \mathbf{u}\cdot\mathbf{v} + \int_{\partial\Omega} p\mathbf{v}\cdot\mathbf{n} - \int_\Omega p\operatorname{div}\mathbf{v} \\ &= \frac{1}{k}\int_\Omega \mathbf{u}\cdot\mathbf{v} + \int_{\Gamma_D} d\mathbf{v}\cdot\mathbf{n} - \int_\Omega p\operatorname{div}\mathbf{v} \end{aligned}$$

for the first equation, and

$$\int_\Omega q\operatorname{div}\mathbf{u} = \int_\Omega fq$$

for the second. For the integrals to be defined, we need $\mathbf{v}$ and $\operatorname{div}\mathbf{v}$ to be $L^2(\Omega)$, so we define the space

$$V = H^1(\operatorname{div},\Omega) = \{\mathbf{v} \in L^2(\Omega) : \operatorname{div}\mathbf{v} \in L^2(\Omega)\}$$

which is a Hilbert space endowed with the scalar product $(\mathbf{u},\mathbf{v})_V = (\mathbf{u},\mathbf{v})_{L^2(\Omega)} + (\operatorname{div}\mathbf{u},\operatorname{div}\mathbf{v})_{L^2(\Omega)}$ and its induced norm. We would like to define a V_0 to encapsulate the condition $\mathbf{u}\cdot\mathbf{n} = 0$ but for that we need a notion of trace in V. Let's show that $\mathbf{v}\cdot\mathbf{n}$ is a bounded linear form on $H^{1/2}(\Omega)$, namely a element of $H^{-1/2}(\Omega)$. For any function $g \in H^{1/2}(\partial\Omega)$, we can by definition find a $q_g \in H^1(\Omega)$ such that the trace of q_g is g and $\|q_g\|_{H^1(\Omega)} \leq \gamma\|g\|_{H^{1/2}(\partial\Omega)}$. Then we see that

$$\begin{aligned} \int_{\partial\Omega} g\mathbf{u}\cdot\mathbf{n} &= \int_\Omega q_g\operatorname{div}\mathbf{u} + \int_\Omega \mathbf{u}\nabla q_g \\ &\leq \|q_g\|_{L^2(\Omega)}\|\operatorname{div}\mathbf{u}\|_{L^2(\Omega)} + \|\mathbf{u}\|_{L^2(\Omega)}\|\nabla q_g\|_{L^2(\Omega)} \\ &\leq \|\mathbf{u}\|_V\|q_g\|_{H^1(\Omega)} \quad \text{(by Cauchy-Schwarz)} \\ &\leq \|\mathbf{u}\|_V\gamma\|g\|_{H^{1/2}(\partial\Omega)} \end{aligned}$$

and $\mathbf{u}\cdot\mathbf{n}$ is indeed a bounded functional for any $\mathbf{u}$. We can now define the subspace

$$V_0 = H_{\Gamma_N}(\operatorname{div},\Omega) = \{\mathbf{v}\in V : \mathbf{v}\cdot\mathbf{n} = 0 \text{ on } \Gamma_N\}.$$

The weak formulation becomes then:
find $\mathbf{u}\in V_0 = H_{\Gamma_N}(\operatorname{div},\Omega)$ and $p\in Q = L^2(\Omega)$ such that

$$\begin{cases} \frac{1}{k}\int_\Omega \mathbf{u}\cdot\mathbf{v} - \int_\Omega p \operatorname{div}\mathbf{v} = -\int_{\Gamma_D} d\mathbf{v}\cdot\mathbf{n} & \forall \mathbf{v}\in V_0 = H_{\Gamma_N}(\operatorname{div},\Omega) \\ \int_\Omega q\operatorname{div}\mathbf{u} = \int_\Omega fq & \forall q\in Q = L^2(\Omega) \end{cases}$$

or: find $\mathbf{u}\in V_0$ and $p\in Q$ such that

$$\begin{cases} a(\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) = D(\mathbf{v}) & \forall \mathbf{v}\in V_0 \\ b(\mathbf{v},q) = F(q) & \forall q\in Q \end{cases}$$

where we have defined the linear forms

$$\begin{array}{cc} a(\mathbf{u},\mathbf{v}) = \frac{1}{k}\int_\Omega \mathbf{u}\cdot\mathbf{v} & b(\mathbf{v},q) = -\int_\Omega q\operatorname{div}\mathbf{v} \\ D(\mathbf{v}) = -\int_{\Gamma_N} d\mathbf{v}\cdot\mathbf{n} & F(q) = \int_\Omega fq. \end{array}$$

First, we see that the data f and d are clearly regular enough and are respectively $L^2(\Omega)$ and $L^{1/2}(\Omega)$ We want to show continuity, coercivity, and inf-sup conditions on the linear forms, to use the theorem of of the course about well-posedness for mixed-forms. Continuity of a:

$$|a(\mathbf{u},\mathbf{v})| = |(\mathbf{u},\mathbf{v})_{L^2(\Omega)}| \le \frac{1}{k}\|\mathbf{u}\|_{L^2(\Omega)}\|\mathbf{v}\|_{L^2(\Omega)} \le \frac{1}{k}\|\mathbf{u}\|_{V_0}\|\mathbf{v}\|_{V_0}$$

Continuity of b:

$$|b(\mathbf{u},p)| \le \|\operatorname{div}\mathbf{u}\|_{L^2(\Omega)}\|p\|_{L^2(\Omega)} \le \|\mathbf{u}\|_{V_0}\|p\|_Q$$

Continuity of F:

$$|F(q)| \le \|f\|_{L^2(\Omega)}\|q\|_{L^2(\Omega)=Q}$$

The continuity of D is a consequence of the trace theorem:

$$|D(\mathbf{v})| \le \|d\|_{H^{1/2}(\partial\Omega)}\|\mathbf{v}\cdot\mathbf{n}\|_{H^{-1/2}(\partial\Omega)} \le \|d\|_{H^{1/2}(\partial\Omega)}\gamma\|\mathbf{v}\|_{V_0}$$

The Coercivity of a on $V^0 = \ker B = \{\mathbf{v}\in V : \operatorname{div}\mathbf{v} = 0\}$ where B is the linear operator such that $(\mathbf{u}, B\mathbf{v})_{L^2(\Omega)} = b(\mathbf{u},\mathbf{v})$, is immediate since

$$a(\mathbf{v},\mathbf{v}) = 1/k\|\mathbf{v}\|^2_{L^2(\Omega)}. = 1/k\big(\|\mathbf{v}\|^2_{L^2(\Omega)} + \|\operatorname{div}\mathbf{v}\|^2_{L^2(\Omega)}\big) = 1/k\|\mathbf{v}\|^2_V.$$

Now come the inf-sup condition, and we see that for

$$\frac{|b(\mathbf{v},q)|}{\|\mathbf{v}\|_V} = \frac{\int_\Omega q\operatorname{div}\mathbf{v}}{\|\mathbf{v}\|_V}$$

to be big, one should find a $div\mathbf{v} = q$. We can do that by resolving the Dirichlet problem $\begin{cases} \Delta\psi = q & \text{in } \Omega \\ \psi = 0 & \text{in } \partial\Omega \end{cases}$ which has a $H^2(\Omega)$-solution since $q\in L^2(\Omega)$, and hence $\nabla\psi$, as well as $\operatorname{div}\nabla\psi = q$ are in $L^2(\Omega)$. We have then $\nabla\psi\in V$ with

$$\|\nabla\psi\|^2_V = \|\nabla\psi\|^2_{L^2(\Omega)} + \|q\|^2_{L^2(\Omega)} \le (1+C)\|q\|^2_{L^2(\Omega)}$$

by Poincaré inequality, and taking $\nabla\psi$ as a value for $\mathbf{v}$ we get

$$\sup_{\mathbf{v}\in V_0} \frac{|b(\mathbf{v},q)|}{\|\mathbf{v}\|_V} \geq \frac{|b(\nabla\psi,q)|}{\|\nabla\psi\|_V} = \frac{\|q\|_Q^2}{\|\nabla\psi\|_V} \geq (1+c)^{1/2}\|q\|_Q,$$

so the inf-sup condition is satisfied with $\beta = (1+c)^{1/2}$.

All the conditions of the theorem are satisfied and the problem admit a unique solution with $\|\mathbf{u}\|_V + \|p\|_Q \leq C_2\|D\|'_V + \|F\|'_Q$, so the problem is well posed.

– The diffusion-transport equation

The diffusion-transport equation can be written $\begin{cases} -\nu\Delta c + \mathbf{u}\cdot\nabla c = g & \text{in } \Omega \\ c = 0 & \text{on } \Lambda_D = \Gamma_1\cup\Gamma_3\cup\Gamma_4 \\ \nu\nabla c\cdot\mathbf{n} = 0 & \text{on } \Lambda_N = \Gamma_2 \end{cases}$ This is an elliptic equation and we search for a weak form by multiplying by a smooth enough w and integrating over Ω. We take w such that its trace on Γ_D is zero, so that it doesn't affect the Dirichlet condition for of the solution. By integration by part we obtain:

$$\begin{aligned}
\int_\Omega(-\nu\Delta c + \mathbf{u}\cdot\nabla c)w &= -\nu\int_\Omega w\Delta c + \int_\Omega w\mathbf{u}\cdot\nabla c \\
&= \nu\int_\Omega \nabla w\nabla c - \int_{\partial\Omega} w\nabla c\cdot\mathbf{n} + \int_\Omega w\mathbf{u}\cdot\nabla c \\
&= \nu\int_\Omega \nabla w\nabla c - \int_{\Lambda_N} w\mathbf{0} + \int_{\Lambda_D} 0\nabla c\cdot\mathbf{n} + \int_\Omega w\mathbf{u}\cdot\nabla c \\
&= \nu\int_\Omega \nabla w\nabla c + \int_\Omega w\mathbf{u}\cdot\nabla c
\end{aligned}$$

where we have used $w = 0$ on Λ_D and $\nu\nabla c\cdot\mathbf{n} = 0$ on Λ_N. Doing the same for the right hand side we get

$$\nu\int_\Omega \nabla w\nabla c + \int_\Omega w\mathbf{u}\cdot\nabla c = \int_\Omega gw$$

For all the integral to be well defined, we need at least $w, c \in H^1(\Omega) =: W$ and $g \in L^2(\Omega)$; we define $W_0 = \{w \in W : w = 0 \text{ on } \Gamma_D\}$ so that actually $c, w \in W_0$ for the hypothesis we used. This is a Hilbert subspace of $H^1(\Omega)$, seen as the kernel of the trace function on Γ_D and endowed with the same scalar product as in $H^1(\Omega)$.

We want to apply the Lax-Milgram theorem, we define the functions

$$\alpha(c,w) := \nu\int_\Omega \nabla c\nabla w + \int_\Omega w\mathbf{u}\cdot\nabla c \quad\text{and}\quad G(w) := \int_\Omega gw$$

so that the problem is: find $c \in W_0$ such that $\alpha(c,w) = G(w) \quad \forall w \in W_0$. The continuity of α follow from

$$\begin{aligned}
|\alpha(c,w)| &\leq \nu\|\nabla c\|_{L^2(\Omega)}\|\nabla w\|_{L^2(\Omega)} + \|w\|_{L^2(\Omega)}\|\mathbf{u}\cdot\nabla c\|_{L^2(\Omega)} \\
&\leq
\end{aligned}$$

il faudrait que une des 3 soit essentiellement bornée

Question 2

For the Darcy problem, as it is a mixed problem, we choose the Raviart-Thomas finite element space on triangles, $\mathbb{RT}_r$ (of degree r). It is adapted because it use a non standard finite element space for the velocity fields $\mathbf{u}$ and take care of the divergence, as well as the continuity in edges of triangles with respect to normal direction. For the advection-diffusion problem, we choose the space of continuous finite element of degree r, $\mathbb{P}_r$. The choice of the degree is $r = 2$. We notice that the data is not continuous, and therefore the space chosen should theoretically allow discontinuous functions. However, the discontinuity of the data is just a consequence of a simple choice, to models the pump and the presence of pollution. Therefore we can take a higher method, we could have chosen a continuous data but the finite elements will smooth it for us, and we will have a way better convergence. Namely, both methods has a rate of convergence of $r + 1$ (assuming that the solution has regularity $H^{r+1}(\mathrm{div}, \Omega)$).

Question 3

The code is listed in the appendix. Here are the results:

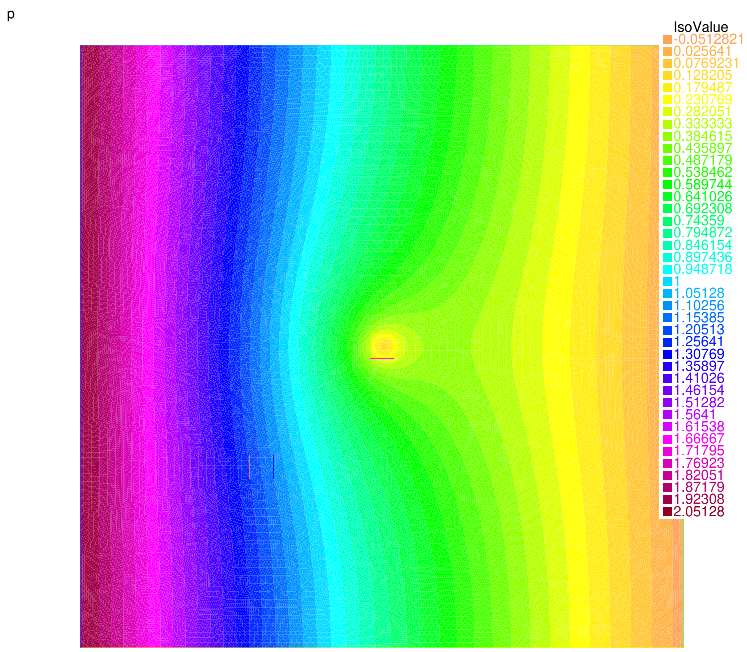

Figure 1: The value of the pressure p.

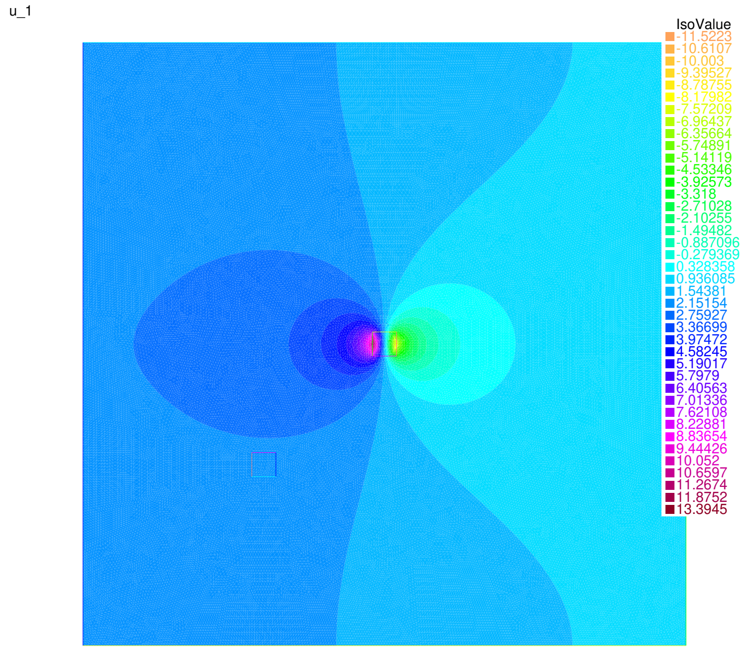

Figure 2: The value of the first coordinate of the velocity u.

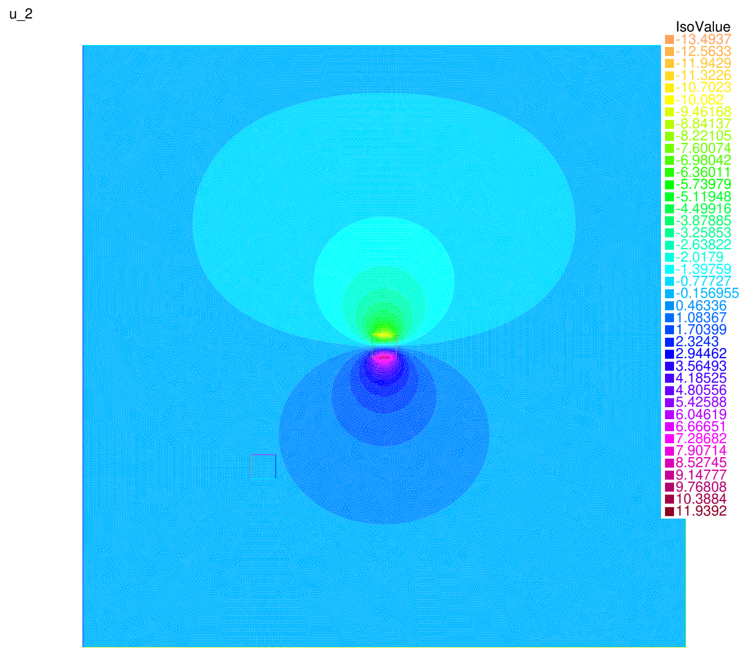

Figure 3: The value of the second coordinate of the velocity u.

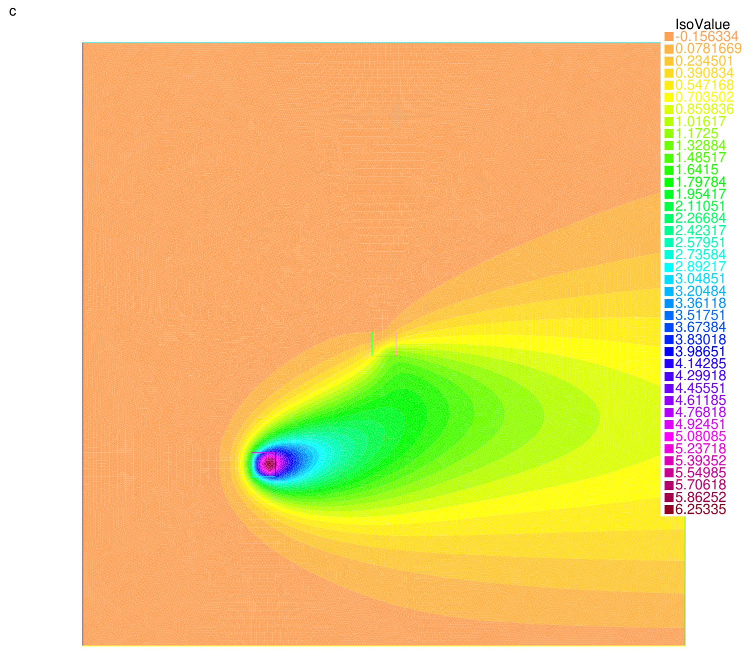

Figure 4: The value of the pollution c.

Question 4

We copy the code of question 3 and add a loop for the mesh generation, where we refine it at each time. We use the function "trunc" and divide by two the edges each time, (meaning that we multiply by four the number of triangles). The results are not quite as regular as we expected. We see that the code we have written doesn't converge really precisely as we augment the refinement. The value oscillate around 0.0009 and 0.005 and the relative error is biggest than 0.1. The method take a lot of time as soon as we arrive the third iteration. We conclude that the uniform mesh is very costly, and doesn't need to be as precise everywhere, since the only irregularities appear around the two little squares. An adaptive method which take into account the shape of the current c should give better results. We will do it in the next question.

Question 5

We do as in the last question but instead of using a uniform refinement, we use the funcion "adaptmesh" and giv it the current function c. As expected the results are wy better and we see a convergence appear.

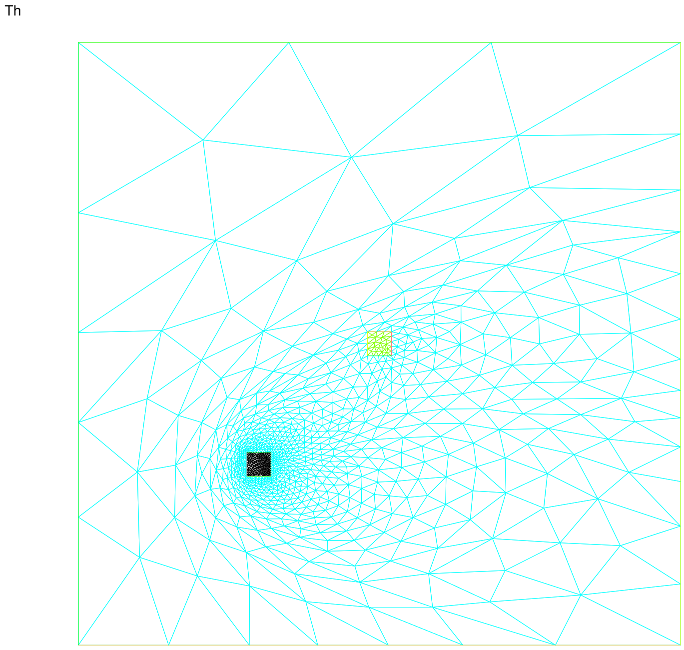

Figure 5: The non uniform mesh

The output is the following:
The resulting polution in the pump is
At iteration 1: phi= 0.000324542
At iteration 2: phi= 0.0010733 with relative error 0.697622
At iteration 3: phi= 0.00106978 with relative error 0.00329416
The relative error is attained with few iteration so we can't show with confidence that the error has the wanted rate.

Appendix

A Code of question 3

```
load "Element_Mixte"
bool waits=true; //display parameter

// Constants:
real k=1.;
real pin=2.;
real nu=0.05;

// Borders:
// the main border:
border g1(t=0,1){x=t; y=0;}
border g2(t=0,1){x=1; y=t;}
```

```
border g3(t=0,1){x=1-t; y=1;}
border g4(t=0,1){x=0; y=1-t;}
// the ransformations for the two squares:
func real transf1(real s) { return (s-0.5)*0.04 + 0.5; }
func real transf2(real s) { return (s-0.5)*0.04 + 0.3; }
//C1:
border c01(t=0,1){x=transf1(t); y=transf1(0);}
border c02(t=0,1){x=transf1(1); y=transf1(t);}
border c03(t=0,1){x=transf1(1-t); y=transf1(1);}
border c04(t=0,1){x=transf1(0); y=transf1(1-t);}
//C2:
border c11(t=0,1){x=transf2(t); y=transf2(0);}
border c12(t=0,1){x=transf2(1); y=transf2(t);}
border c13(t=0,1){x=transf2(1-t); y=transf2(1);}
border c14(t=0,1){x=transf2(0); y=transf2(1-t);}

// Mesh:
int i=3; // Refinement coeficient
int j=i;
mesh Th = buildmesh(g1(25*i)+g2(25*i)+g3(25*i)+g4(25*i)
                    + c01(i)+c02(i)+c03(i)+c04(i)
                    + c11(j)+c12(j)+c13(j)+c14(j));
//plot(Th,wait=waits);
//plot(Th2,wait=waits);
int regionomega = Th(0.1,0.1).region; //the big region ~= Omega
int regionc1 = Th(0.5,0.5).region; //C1
int regionc2 = Th(0.3,0.3).region; //C2

// Finite elements spaces and some elements:
fespace Vh(Th,RT2);
Vh [uh1, uh2],[vh1, vh2];
fespace Qh(Th,P2);
Qh ph,qh;
Qh fh = -1000.*(region==regionc1);
fespace Wh(Th,P2);
Wh ch, wh;
Wh gh = 1000.*(region==regionc2);

// The problems:
problem DarcyRT([uh1,uh2,ph],[vh1,vh2,qh]) =
         int2d(Th)(1./k*(uh1*vh1+uh2*vh2))
         - int2d(Th)(ph*(dx(vh1) + dy(vh2)))
         + int1d(Th,4)(-pin*vh1)
         + int2d(Th)(qh*(dx(uh1) + dy(uh2)))
         - int2d(Th)(fh*qh)
         + on(1,3,uh2=0)
         + on(4,ph=pin) + on(2,ph=0); //Neuman condition as a Dirichlet one
DarcyRT;
problem DiffusionP(ch,wh) =
```

```
        int2d(Th)(nu*(dx(ch)*dx(wh) + dy(ch)*dy(wh)))
        + int2d(Th)(wh*(uh1*dx(ch) + uh2*dy(ch)))
        - int2d(Th)(gh*wh)
        + on(1,3,4,ch=0); //Dirichlet condition
DiffusionP;

// Plots:
plot(ph,fill=true,nbiso=40,value=true,cmm="p",ps="graphics/ph.eps",wait=waits);
plot(uh1,fill=true,nbiso=40,value=true,cmm="u_1",ps="graphics/uh1.eps",wait=waits);
plot(uh2,fill=true,nbiso=40,value=true,cmm="u_2",ps="graphics/uh2.eps",wait=waits);

plot(ch,fill=true,nbiso=40,value=true,cmm="c",ps="graphics/c.eps",wait=waits);

// Pollution
real phi = int2d(Th,regionc1)(ch);
cout<<"-----------------------------------------------------------------"<<endl;
cout<<"The resulting polution in the pump is: "<< phi <<endl;
cout<<"-----------------------------------------------------------------"<<endl;
```

B Code of question 4

```
load "Element_Mixte"
bool waits=false; //display parameter

// Constants:
real k=1.;
real pin=2.;
real nu=0.05;

// Borders:
border g1(t=0,1){x=t; y=0;}
border g2(t=0,1){x=1; y=t;}
border g3(t=0,1){x=1-t; y=1;}
border g4(t=0,1){x=0; y=1-t;}

func real transf1(real s) { return (s-0.5)*0.04 + 0.5; }
func real transf2(real s) { return (s-0.5)*0.04 + 0.3; }
border c01(t=0,1){x=transf1(t); y=transf1(0);}
border c02(t=0,1){x=transf1(1); y=transf1(t);}
border c03(t=0,1){x=transf1(1-t); y=transf1(1);}
border c04(t=0,1){x=transf1(0); y=transf1(1-t);}

border c11(t=0,1){x=transf2(t); y=transf2(0);}
border c12(t=0,1){x=transf2(1); y=transf2(t);}
border c13(t=0,1){x=transf2(1-t); y=transf2(1);}
border c14(t=0,1){x=transf2(0); y=transf2(1-t);}

// Mesh:
int i=1; // Refinement coeficient
```

```
int j=i;
mesh Th= buildmesh(g1(25*i)+g2(25*i)+g3(25*i)+g4(25*i)
                   + c01(i)+c02(i)+c03(i)+c04(i)
                   + c11(j)+c12(j)+c13(j)+c14(j));

//plot(Th2,wait=waits);

bool condition = true;
int iter = 0;
real phi=0.;
real lastphi=0.;
real err;
cout<<"The resulting polution in the pump is"<<endl;
while(condition * iter<4){
  iter = iter + 1;
  //the current mesh:
  plot(Th,wait=waits);
  // compute the phi:
  int regionomega = Th(0.1,0.1).region; //the big region ~= Omega
  int regionc1 = Th(0.5,0.5).region; //C1
  int regionc2 = Th(0.3,0.3).region; //C2

  // Finite elements spaces and some elements:
  fespace Vh(Th,RT2);
  Vh [uh1, uh2],[vh1, vh2];
  fespace Qh(Th,P2);
  Qh ph,qh;
  Qh fh = -1000.*(region==regionc1);
  fespace Wh(Th,P2);
  Wh ch, wh;
  Wh gh = 1000.*(region==regionc2);

  // The problems:
  problem DarcyRT([uh1,uh2,ph],[vh1,vh2,qh]) =
          int2d(Th)(1./k*(uh1*vh1+uh2*vh2))
          - int2d(Th)(ph*(dx(vh1) + dy(vh2)))
          + int1d(Th,4)(-pin*vh1)
          + int2d(Th)(qh*(dx(uh1) + dy(uh2)))
          - int2d(Th)(fh*qh)
          + on(1,3,uh2=0)
          + on(4,ph=pin) + on(2,ph=0); //Neuman condition as a Dirichlet one
  DarcyRT;
  problem DiffusionP(ch,wh) =
          int2d(Th)(nu*(dx(ch)*dx(wh) + dy(ch)*dy(wh)))
          + int2d(Th)(wh*(uh1*dx(ch) + uh2*dy(ch)))
          - int2d(Th)(gh*wh)
          + on(1,3,4,ch=0); //Dirichlet condition
  DiffusionP;
```

```
  // Pollution
  lastphi = phi;
  phi = int2d(Th,regionc1)(ch);
  cout<<"At iteration "<< iter <<": phi= "<< phi;
  if (iter>1){
    err = abs(phi - lastphi) / phi;
    condition = err > 0.1;
    cout<<" with relative error "<< err;
  }
  cout<<endl;
  Th = trunc(Th,1,split=2);
}
```

C Code of question 5

```
load "Element_Mixte"
bool waits=false; //display parameter

// Constants:
real k=1.;
real pin=2.;
real nu=0.05;

// Borders:
border g1(t=0,1){x=t; y=0;}
border g2(t=0,1){x=1; y=t;}
border g3(t=0,1){x=1-t; y=1;}
border g4(t=0,1){x=0; y=1-t;}

func real transf1(real s) { return (s-0.5)*0.04 + 0.5; }
func real transf2(real s) { return (s-0.5)*0.04 + 0.3; }
border c01(t=0,1){x=transf1(t); y=transf1(0);}
border c02(t=0,1){x=transf1(1); y=transf1(t);}
border c03(t=0,1){x=transf1(1-t); y=transf1(1);}
border c04(t=0,1){x=transf1(0); y=transf1(1-t);}

border c11(t=0,1){x=transf2(t); y=transf2(0);}
border c12(t=0,1){x=transf2(1); y=transf2(t);}
border c13(t=0,1){x=transf2(1-t); y=transf2(1);}
border c14(t=0,1){x=transf2(0); y=transf2(1-t);}

// Mesh:
int i=1; // Refinement coeficient
int j=i;
mesh Th= buildmesh(g1(25*i)+g2(25*i)+g3(25*i)+g4(25*i)
                   + c01(i)+c02(i)+c03(i)+c04(i)
                   + c11(j)+c12(j)+c13(j)+c14(j));
//plot(Th2,wait=waits);
```

```
bool condition = true;
int iter = 0;
real phi=0.;
real lastphi=0.;
real err;
cout<<"The resulting polution in the pump is"<<endl;
//Loop with refinement until relative error is attained:
while(condition){
  iter = iter + 1;
  //split the mesh:
  plot(Th, wait=waits);
  // compute the new phi
  int regionomega = Th(0.1,0.1).region; //the big region ~= Omega
  int regionc1 = Th(0.5,0.5).region; //C1
  int regionc2 = Th(0.3,0.3).region; //C2

  // Finite elements spaces and some elements:
  fespace Vh(Th,RT2);
  Vh [uh1, uh2],[vh1, vh2];
  fespace Qh(Th,P2);
  Qh ph,qh;
  Qh fh = -1000.*(region==regionc1);
  fespace Wh(Th,P2);
  Wh ch, wh;
  Wh gh = 1000.*(region==regionc2);

  // The problems:
  problem DarcyRT([uh1,uh2,ph],[vh1,vh2,qh]) =
          int2d(Th)(1./k*(uh1*vh1+uh2*vh2))
          - int2d(Th)(ph*(dx(vh1) + dy(vh2)))
          + int1d(Th,4)(-pin*vh1)
          + int2d(Th)(qh*(dx(uh1) + dy(uh2)))
          - int2d(Th)(fh*qh)
          + on(1,3,uh2=0)
          + on(4,ph=pin) + on(2,ph=0); //Neuman condition as a Dirichlet one
  DarcyRT;
  problem DiffusionP(ch,wh) =
          int2d(Th)(nu*(dx(ch)*dx(wh) + dy(ch)*dy(wh)))
          + int2d(Th)(wh*(uh1*dx(ch) + uh2*dy(ch)))
          - int2d(Th)(gh*wh)
          + on(1,3,4,ch=0); //Dirichlet condition
  DiffusionP;

  // Pollution
  lastphi = phi;
  phi = int2d(Th,regionc1)(ch);
  cout<<"At iteration "<< iter <<": phi= "<< phi;
  if (iter>1){
    err = abs(phi - lastphi) / phi;
```

```
    condition = err > 0.01;
    cout<<" with relative error "<< err;
  }
  cout<<endl;
  Th = adaptmesh(Th,ch);
}
plot(Th,cmm="Th",ps="graphics/Th.eps",wait=waits);
```