

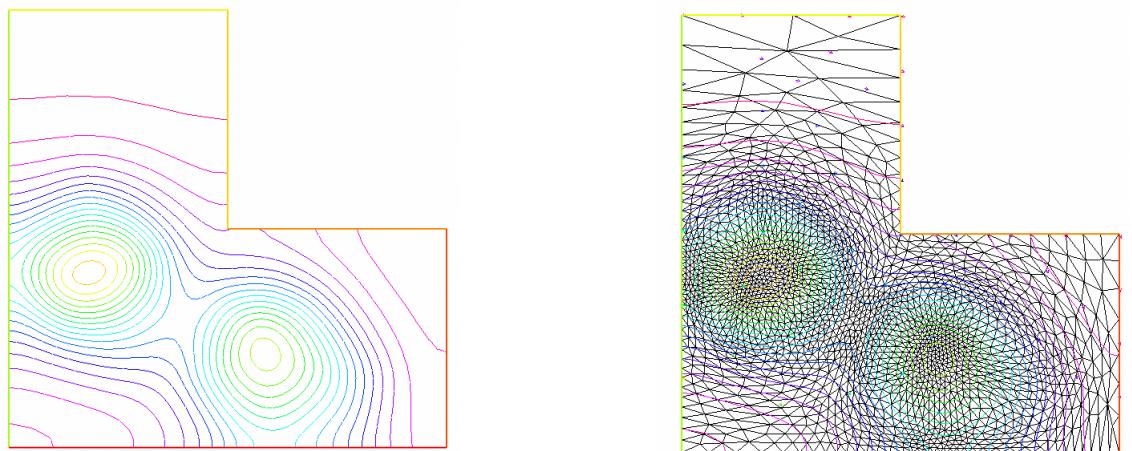
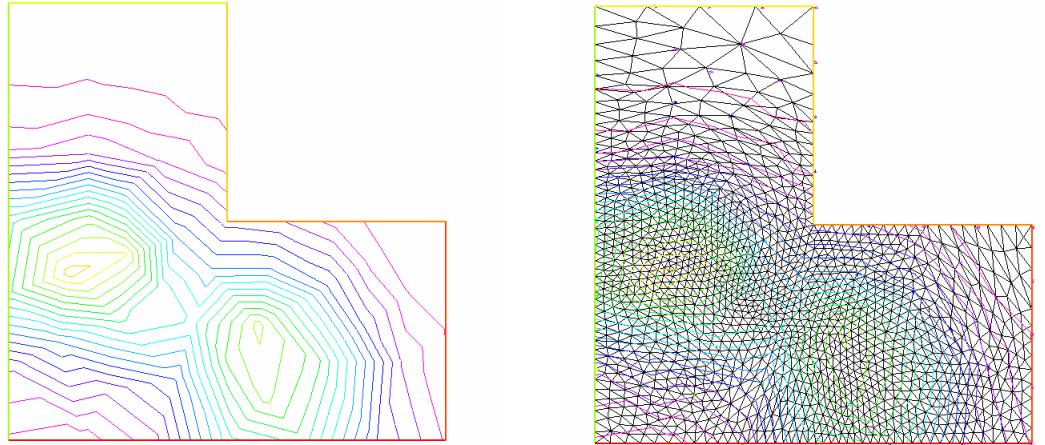


Faculté des sciences de Montpellier  
Rapport séance 5

Étudiant: Giscard Leonel Zouakeu

## 1-COMPRÉHENSION ET EXPLICATION DU CODE: "adapsteady.edp"

```
1 border a(t=0,1.0){x=t;    y=0;    label=1;} // comment
2 border b(t=0,0.5){x=1;    y=t;    label=2;};
3 border c(t=0,0.5){x=1-t; y=0.5;label=3;};
4 border d(t=0.5,1){x=0.5; y=t;    label=4;};
5 border e(t=0.5,1){x=1-t; y=1;    label=5;};
6 border f(t=0.0,1){x=0;    y=1-t;label=6;};
7 mesh Th = buildmesh (a(6) + b(4) + c(4) +d(4) + e(4) + f
(6));
//savemesh(Th,"th.msh");
9 fespace Vh(Th,P1);
10
11
12
13
14 Vh u,v,source;
15 Vh vit1 = y*0.01, vit2 = -x*0.01; // velocity
16 real error=0.005,xnu=0.01,xlam=1;
17 source=exp(-100*((x-0.2)^2+(y-0.4)^2))+exp(-100*((x-0.6)
^2+(y-0.2)^2));
18
19 problem Probem1(u,v,solver=CG,eps=1.0e-6) =
20     int2d(Th,qforder=2)( u*v*1.0e-10+ xnu*( dx(u)*dx(v) +
dy(u)*dy(v) ))
21     + int2d(Th,qforder=2)( (-u*vit1*dx(v) - u*dy(v)*vit2) )
22     + int2d(Th,qforder=2)( source*v)
23     + int2d(Th,qforder=2)( xlam*u*v);
24
25 for (int iadap=0;iadap< 10;iadap++)
26 {
27     Probem1;
28     cout << u[].min << " " << u[].max << endl;
29     plot(u,wait=1);
30     Th=adaptmesh(Th,u,err=error,hmin=0.01,hmax=0.5);
31     plot(Th,u,[vit1,vit2],wait=1);
32     u=u; // interpolation from old to new
33     meshes
34 } ;
```



D'après notre code `adapsteady.edp` , la formulation variationnelle du problème s'écrit:

$$\left( \int_{\Omega} u.v + \nu \int_{\Omega} \nabla u \nabla v - \int_{\Omega} u.V(t, x) \nabla v + \int_{\Omega} f.v + \lambda \int_{\Omega} u.v \right) = 0 \quad (1)$$

avec  $v$  appartenant à l'espace des fonctions test, plus précisément  $v \in \mathbf{H}_0^1(\Omega)$ . Nous avons la fonction source " $f$ " définie dans notre code par:

$$f(x, y) = \exp(-100*((x-0.2)^2 + (y-0.4)^2)) + \exp(-100*((x-0.6)^2 + (y-0.2)^2))$$

et donc le problème lié à cette formulation variationnelle (1) s'écrit:

$$\begin{aligned} -\nu \Delta u - \lambda u + V(t, x) \nabla u &= f \text{ sur } \Omega \\ u &= 0 \text{ sur } \partial\Omega \end{aligned} \tag{2}$$

## 2- PROBLÈME DE STOKES

Soit  $\Omega \in \mathbf{R}^2$  et  $\mathbf{f}, \mathbf{U}_D$  deux fonctions de  $\mathbf{R}^2 \rightarrow \mathbf{R}^2$ . On considère le problème:  
Chercher deux fonctions  $\mathbf{u} : \mathbf{R}^2 \rightarrow \mathbf{R}^2$  et  $p : \mathbf{R}^2 \rightarrow \mathbf{R}^2$  telles que :

$$\begin{aligned} -\Delta \mathbf{U} + \nabla p &= \mathbf{f} \quad \text{dans } \Omega \\ \nabla \mathbf{U} &= 0 \quad \text{dans } \Omega \\ \mathbf{U} &= \mathbf{U}_D \quad \text{sur } \partial\Omega \end{aligned} \tag{3}$$

Ou , pour  $\mathbf{U} = (u_1, u_2)^t$ , on a posé  $\Delta \mathbf{U} = (\Delta u_1, \Delta u_2)^t, \nabla \mathbf{U} = (\frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial x})$ .  
Et pour toute fonction  $v : \mathbf{R}^2 \rightarrow \mathbf{R}$ ,  $\Delta v = \frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_2}{\partial x^2}$ .

### Formulation variationnelle du problème de Stokes

On pose:

$$\mathbf{H}^1(\Omega) = \{v \in \mathbf{L}^2(\Omega), \nabla v \in (\mathbf{L}^2(\Omega))^2\}; \mathbf{H}_0^1(\Omega) = \{v \in \mathbf{H}^1(\Omega); v = 0 \text{ sur } \partial\Omega\}$$

$$\begin{aligned} V &= \mathbf{H}^1(\Omega) \times \mathbf{H}_0^1(\Omega); M = \left\{ q \in \mathbf{L}^2(\Omega); \int_{\Omega} q \cdot \mathbf{d} \mathbf{d}x \mathbf{d}y = 0 \right\} \\ a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \nabla \mathbf{u} \nabla \mathbf{v} \mathbf{d}x \mathbf{d}y; \quad b(\mathbf{u}, p) = - \int_{\Omega} p \nabla \cdot \mathbf{v} \mathbf{d}x \mathbf{d}y \\ l(\mathbf{v}) &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \mathbf{d}x \mathbf{d}y \end{aligned}$$

**Code adapsteady.edp adapté au problème de Stokes.**

```

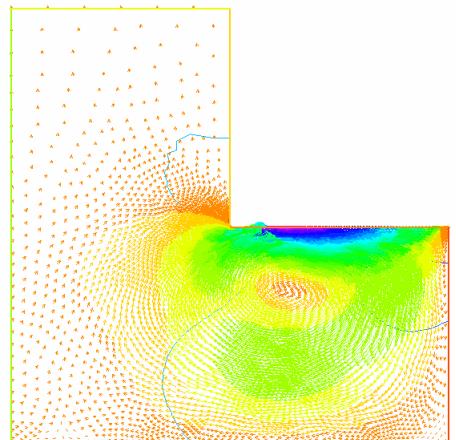
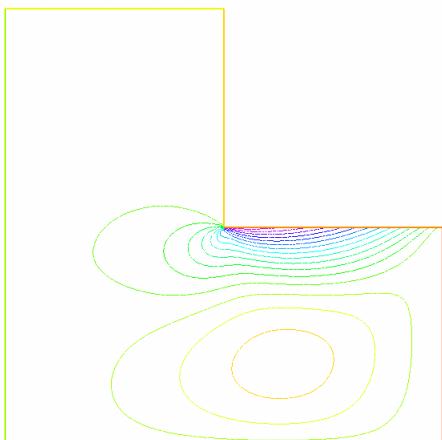
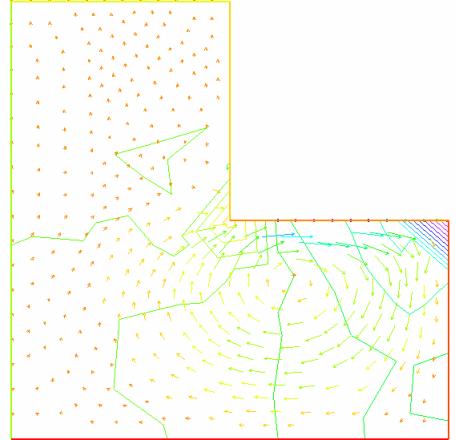
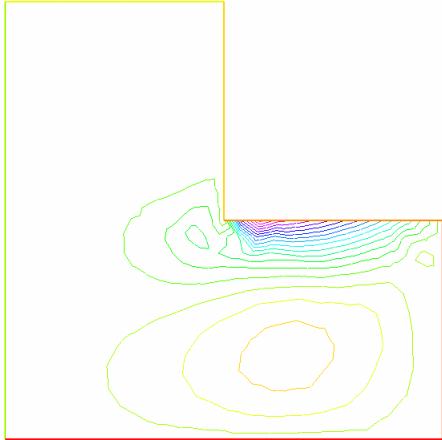
1 border a(t=0,1.0){x=t;    y=0;    label=1;}// comment
2 border b(t=0,0.5){x=1;    y=t;    label=2;};
3 border c(t=0,0.5){x=1-t; y=0.5;label=3;};
4 border d(t=0.5,1){x=0.5; y=t;    label=4;};
5 border e(t=0.5,1){x=1-t; y=1;    label=5;};
6 border f(t=0.0,1){x=0;    y=1-t;label=6;};
7 mesh Th = buildmesh (a(6) + b(4) + c(4) +d(4) + e(4) + f
(6));
8
9 //Espace d'approximation de chaque composante de la
   vitess
10 fespace Vh(Th,P2);
11 //Espace d'approximation de la pression
12 fespace Wh(Th,P1);
13 Vh u1,u2; //composantes de la vitesse

```

```

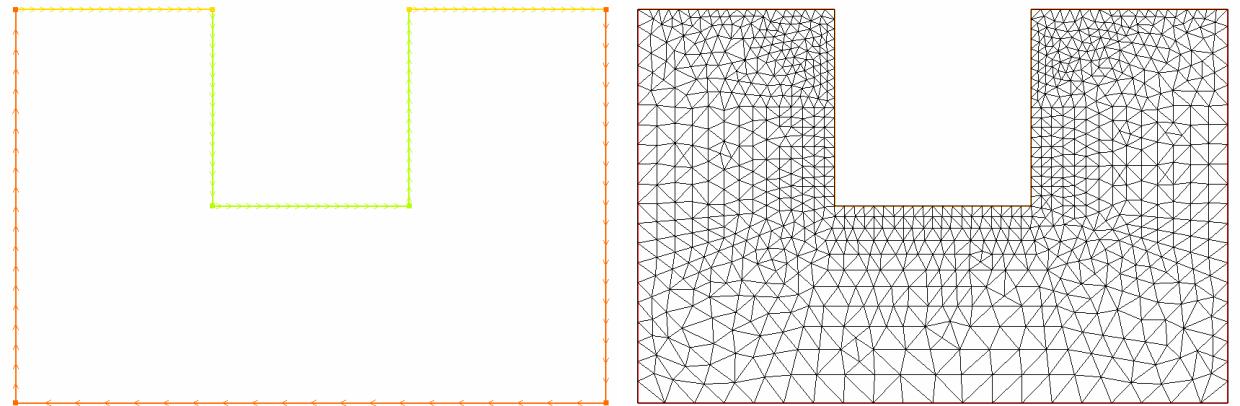
14 Vh v1,v2; //fonctions tests
15 Wh p; //pressionna `calculer
16 Wh q; //fonction test
17
18
19 real nu=1;
20
21 func g=(x)*(1-x)*4;
22 func f1 =exp(-100*((x-0.2)^2+(y-0.4)^2))+exp(-100*((x
-0.6)^2+(y-0.2)^2));
23 func f2 =exp(-100*((x-0.2)^2+(y-0.4)^2))+exp(-100*((x
-0.6)^2+(y-0.2)^2));
24 real error=0.005,xnu=0.01,xlam=1;
25 //e`Probleme variationnel
26 problem Stokes ([u1,u2,p],[v1,v2,q],solver=Crout) =
int2d(Th)(
28 nu *( dx(u1)*dx(v1) + dy(u1)*dy(v1)
29 + dx(u2)*dx(v2) + dy(u2)*dy(v2) )
30 + p*q*(0.000001) /* ne pas oublier ce terme */ - p*dx(v1)
- p*dy(v2)
31 - dx(u1)*q - dy(u2)*q
32 )
33 + int2d(Th) ( -f1*v1 - f2*v2 )
34 + on(3,u1=g,u2=0)
35 + on(1,2,4,u1=0,u2=0);
36 //Resolution
37 //Stokes;
38 //postprocessing
39 //plot(coef=0.2,cmm=" [u1,u2] et p ",p,[u1,u2],ps="
  StokesP2P1.eps",value=1,wait=1);
40
41 for (int iadap=0;iadap< 10;iadap++)
42 {
43   Stokes;
44   cout << u1[].min << " " << u1[].max << endl;
45   plot(u1,wait=1);
46   Th=adaptmesh(Th,u1,err,error,hmin=0.01,hmax=0.5);
47   plot(coef=0.2,cmm=" [u1,u2] et p ",p,[u1,u2],ps="
    StokesP2P1.eps",value=1,wait=1);
48   u1=u1; // interpolation from old to new
        meshes
49   //error = error/2;
50 }

```



## 2- PROBLÈME DE STOKES SOUS FORME DE $U$

Domaine et maillage du domaine sous forme de  $U$ .



### Code équation Stokes avec domaine sous forme de $U$

```

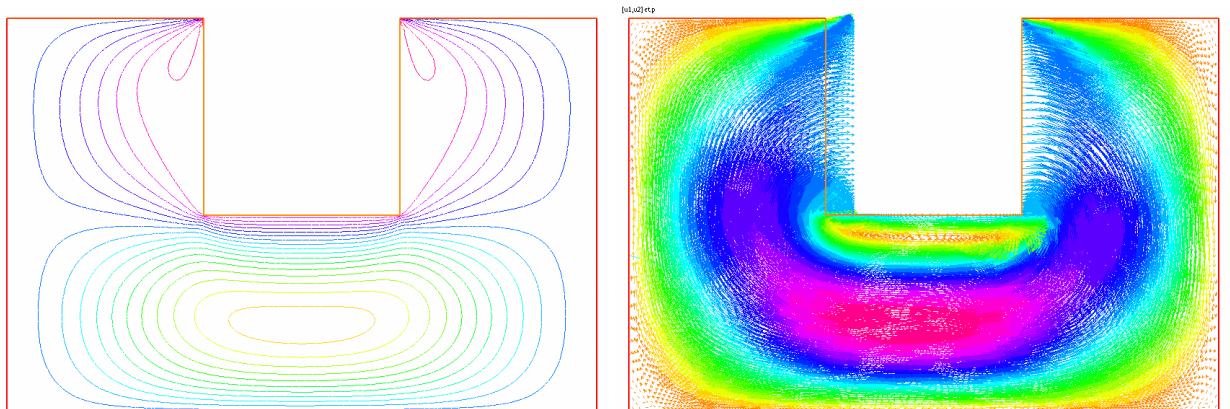
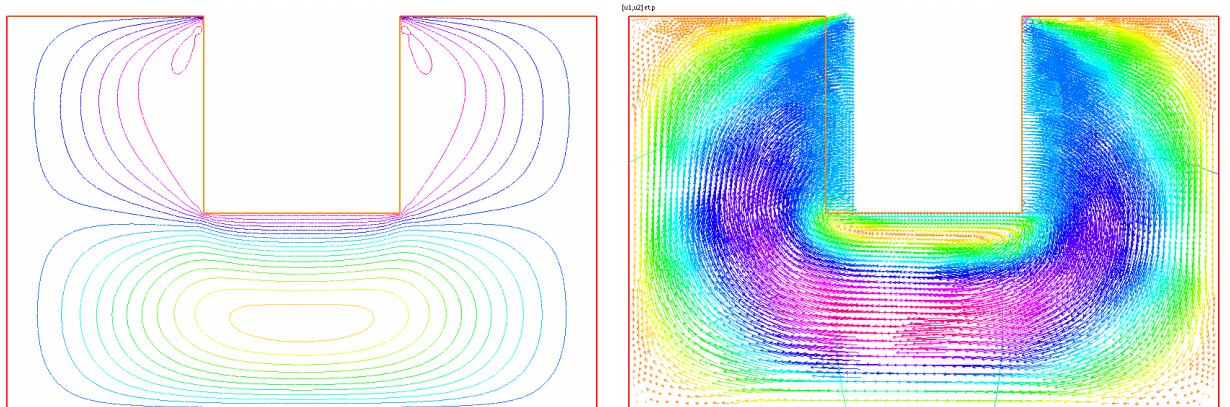
1 int upper = 1;
2 int others = 2;
3 int inner = 3;
4
5 border C01(t=0, 1){x=0; y=-1+t; label=upper;}
6 border C02(t=0, 1){x=1.5-1.5*t; y=-1; label=upper;}
7 border C03(t=0, 1){x=1.5; y=-t; label=upper;}
8 border C04(t=0, 1){x=1+0.5*t; y=0; label=others;}
9 border C05(t=0, 1){x=0.5+0.5*t; y=0; label=others;}
10 border C06(t=0, 1){x=0.5*t; y=0; label=others;}
11 border C11(t=0, 1){x=0.5; y=-0.5*t; label=inner;}
12 border C12(t=0, 1){x=0.5+0.5*t; y=-0.5; label=inner;}
13 border C13(t=0, 1){x=1; y=-0.5+0.5*t; label=inner;}
14
15 int n = 20;
16 plot(C01(-n) + C02(-n) + C03(-n) + C04(-n)
17      + C06(-n) + C11(n) + C12(n) + C13(n), wait=true);
18
19 mesh Th = buildmesh(C01(-n) + C02(-n) + C03(-n) + C04(-n)
20      + C06(-n) + C11(-n) + C12(-n) + C13(-n));
21
22 plot(Th, wait=true);
23
24
25 ///////////////////////////////////////////////////////////////////
26 fespace Vh(Th, P2); //definition of the velocity
27   component space
28 fespace Wh(Th, P1); //definition of the pressure space
29 Vh u2, v2;
30 Vh u1, v1;
31 Wh p, q;
32 func g= 1;

```

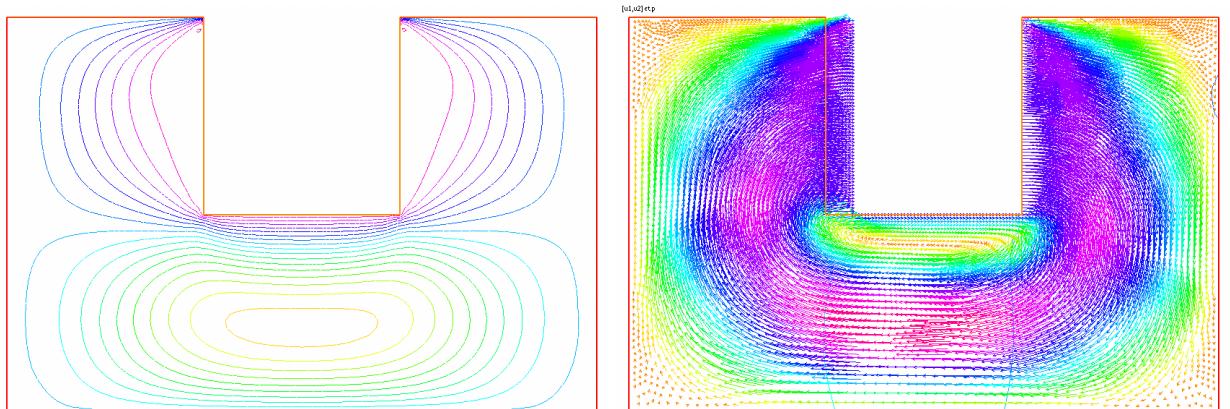
```

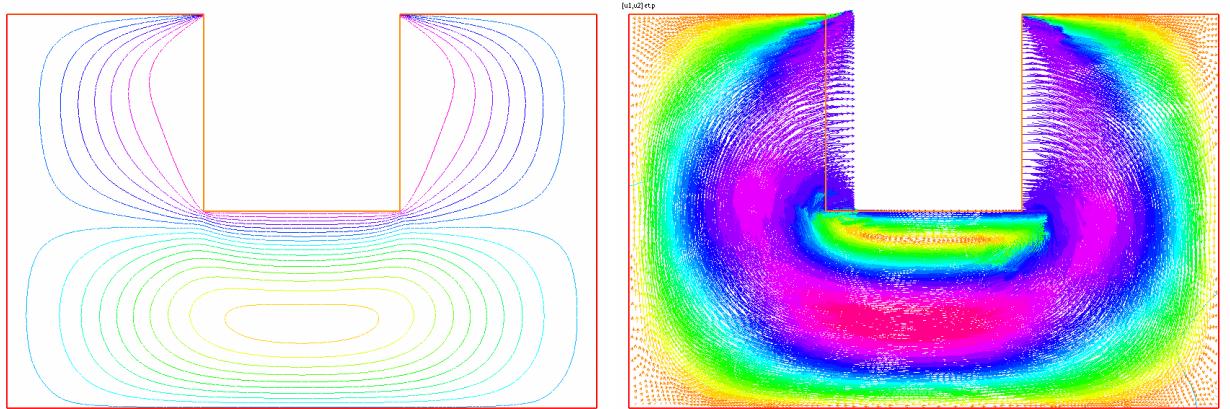
32 func f1 =exp(-100*((x-0.2)^2+(y-0.4)^2))+exp(-100*((x
33 -0.6)^2+(y-0.2)^2));
34 func f2 =exp(-100*((x-0.2)^2+(y-0.4)^2))+exp(-100*((x
35 -0.6)^2+(y-0.2)^2));
36 real error=0.005,xnu=0.01,xlam=1;
37
38 solve Stokes ([u1,u2,p],[v1,v2,q],solver=Crout)
39 = int2d(Th)(
40 (
41 dx(u1)*dx(v1)
42 + dy(u1)*dy(v1)
43 + dx(u2)*dx(v2)
44 + dy(u2)*dy(v2)
45 )
46 - p*q*(0.000001)
47 - p*dx(v1) - p*dy(v2)
48 - dx(u1)*q - dy(u2)*q
49 )
50 + int2d(Th) (-f1*v1 - f2*v2)
51
52 + on(3,u1=g,u2=0)
53 + on(1,2,4,u1=0,u2=0);
54
55 for (int iadap=0;iadap< 10;iadap++)
56 {
57   Stokes;
58   cout << u1[].min << " " << u1[].max << endl;
59   plot(u1,wait=1);
60   Th=adaptmesh(Th,u1,err,error,hmin=0.01,hmax=0.5);
61   plot(coef=0.2,cmm=" [u1,u2] et p ",p,[u1,u2],ps="
62     StokesP2P1.eps",wait=1);
63   u1=u1; // interpolation from old to new
64   meshes
65   //error = error/2;
66 }

```



Probleme Stokes pour un niveau d'erreur de  $\epsilon = 0,01$  et  $0,005$





## 2- PROBLÈME ADRS SOUS FORME DE U

```

1 int upper = 1;
2 int others = 2;
3 int inner = 3;
4
5 border C01(t=0, 1){x=0; y=-1+t; label=upper;}
6 border C02(t=0, 1){x=1.5-1.5*t; y=-1; label=upper;}
7 border C03(t=0, 1){x=1.5; y=-t; label=upper;}
8 border C04(t=0, 1){x=1+0.5*t; y=0; label=others;}
9 border C05(t=0, 1){x=0.5+0.5*t; y=0; label=others;}
10 border C06(t=0, 1){x=0.5*t; y=0; label=others;}
11 border C11(t=0, 1){x=0.5; y=-0.5*t; label=inner;}
12 border C12(t=0, 1){x=0.5+0.5*t; y=-0.5; label=inner;}
13 border C13(t=0, 1){x=1; y=-0.5+0.5*t; label=inner;}
14
15 int n = 20;
16 plot(C01(-n) + C02(-n) + C03(-n) + C04(-n)
17      + C06(-n) + C11(n) + C12(n) + C13(n), wait=true);
18
19 mesh Th = buildmesh(C01(-n) + C02(-n) + C03(-n) + C04(-n)
20      + C06(-n) + C11(-n) + C12(-n) + C13(-n));
21
22 plot(Th, wait=true);
23 //savemesh(Th,"th.msh");
24 fespace Vh(Th,P1);
25
26
27
28
29 Vh u,v,source;

```

```

30 Vh vit1 = y*0.01, vit2 = -x*0.01; // velocity
31 real error=0.005,xnu=0.01,xlam=1;
32 source=exp(-100*((x-0.2)^2+(y-0.4)^2))+exp(-100*((x-0.6)
33 ^2+(y-0.2)^2));
34 problem Probem1(u,v,solver=CG,eps=0.01) =
35     int2d(Th,qforder=2)( u*v*1.0e-10+ xnu*( dx(u)*dx(v) +
36     dy(u)*dy(v) ))
37     + int2d(Th,qforder=2)( (-u*vit1*dx(v) - u*dy(v)*vit2) )
38     + int2d(Th,qforder=2)( source*v )
39     + int2d(Th,qforder=2)( xlam*u*v );
40
41 for (int iadap=0;iadap< 10;iadap++)
42 {
43     Probem1;
44     cout << u[].min << " " << u[].max << endl;
45     plot(u,wait=1);
46     Th=adaptmesh(Th,u,err=error,hmin=0.01,hmax=0.5);
47     plot(Th,u,[vit1,vit2],wait=1);
48     u=u; // interpolation from old to new
49     meshes
//error = error/2;
}

```

