

**Correction 3 : Initiation aux éléments finis 2D**

**Ex 1** *Ecoulements de Poiseuille*

1)

La formulation faible

$$\int_0^1 \int_0^1 \left( \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} \right) dx dy = \int_0^1 \int_0^1 f v dx dy + \int_0^1 \left[ \frac{\partial u}{\partial x} v \right]_{x=0}^{x=1} dy + \int_0^1 \left[ \frac{\partial u}{\partial y} v \right]_{y=0}^{y=1} dx$$

Sur chaque élément, on obtient le système élémentaire  $\mathbf{N} \mathbf{u} = \mathbf{F}$  suivant

$$\mathbf{A}^k \mathbf{U}^k = \mathbf{F}^k \tag{1}$$

où :

$$a_{ij}^k = \int \int_{\mathbf{K}} [\nabla \psi_j^k]^T [\nabla \psi_i^k] dx dy \quad \text{avec} \quad [\nabla \psi_j^k] = \begin{pmatrix} \frac{\partial \psi_j^k}{\partial x} \\ \frac{\partial \psi_j^k}{\partial y} \end{pmatrix}$$

$$f_i^k = \int \int_{\mathbf{K}} \psi_i^k dx dy$$

2)

La matrice Jacobienne

$$DT_k = \begin{pmatrix} \sum_{j=1}^4 x_j^k \frac{\partial L_j}{\partial \xi} & \sum_{j=1}^4 x_j^k \frac{\partial L_j}{\partial \eta} \\ \sum_{j=1}^4 y_j^k \frac{\partial L_j}{\partial \xi} & \sum_{j=1}^4 y_j^k \frac{\partial L_j}{\partial \eta} \end{pmatrix}$$

Ce qui donne le programme

```
x = COOR(CONNEC(iel,:),1); y = COOR(CONNEC(iel,:),2);
[F,DxiFP,DetaFP] = Q1(xg,yg); // on evalue les fonctions de Lagrange au point (xg,yg)

j11 = x(1:4)'*DxiFP; // = sum_j x_j^k dL_j/dxi
j12 = x(1:4)'*DetaFP;
j21 = y(1:4)'*DxiFP;
j22 = y(1:4)'*DetaFP;
//
DT = [j11 j12; j21 j22];
```

3)

Listing du programme

```
1  clear; xdel(0); xdel(1);
2  // on charge les programmes dont on a besoin
3  //
4  exec('Q1.sci'); exec('Q2.sci'); exec('gauss.sci');
5  exec('maillage_2D_Q2.sci');
6  exec('cal_indice.sci'); exec('interpol.sci');
7  exec('trace_maillage_2D.sci'); exec('trace_u_2D.sci');
8
9  //
10 // nbre de points pour l'integration numerique
11 Ng = 3;
12 [XG,PG] = gauss(Ng);
13 //
14 nx = 10; ny = 10; // nbre d'element dans les directions x et y
15 lx = 1.; ly = 1.; // taille du carre
16
17 [CONNEC,COOR,Nelt,Nno,Npt] = maillage_2D_Q2(1,1,nx,ny);
18 trace_maillage_2D(CONNEC,COOR,0);
19
20 //
21 // creation de la matrice NUMER et ADRESS
22 //
23 // Conditions aux limites
24 // u = 0 sur tout le bord
25 NUMER=zeros(Npt,1);
26 NI = 0;
27 nc = 0;
28 for i=1:Npt
29     x = COOR(i,1); y = COOR(i,2);
30     if x <> 0 & x <> 1 & y <> 0 & y <> 1 then
31         NI = NI+1;
32         NUMER(i) = NI;
33     else
34         NUMER(i) = Npt-nc;
35         nc = nc + 1;
36     end;
37 end;
38 //
39 // creation de la matrice ADRESS
40 //
41 ADRESS = zeros(Nelt,Nno);
42 for i =1:Nelt
43     for j = 1:Nno
44         ADRESS(i,j) = NUMER(CONNEC(i,j));
45     end;
46 end;
47 //
48 // creation du vecteur UC
```

```

49 //
50 UC = zeros(Npt-NI,1);
51
52 //
53 //   Fin initialisation des tableaux
54
55 A_glob = spzeros(Npt,Npt); F_glob = zeros(Npt,1);
56
57 for iel = 1:Nelt // boucle sur les elements
58 //
59 // -----
60 // PARTIE   A ECRIRE
61 //-----
62 // calcul du sytème elementaire sur chaque element
63 //
64   A_el = zeros(Nno,Nno);
65   F_el = zeros(Nno,1);
66
67   x = COOR(CONNEC(iel,:),1); y = COOR(CONNEC(iel,:),2);
68
69
70   for ig=1:Ng // double boucle pour l'integration numerique
71     xg = XG(ig);
72     for jg=1:Ng
73       yg = XG(jg);
74     //
75     // calcul de la matrice jacobienne
76     //
77       [F,DxiFP,DetaFP] = Q1(xg,yg);
78       j11 = x(1:4)'*DxiFP;
79       j12 = x(1:4)'*DetaFP;
80       j21 = y(1:4)'*DxiFP;
81       j22 = y(1:4)'*DetaFP;
82     //
83       DT = [j11 j12; j21 j22];
84       jac = det(DT);
85       B = inv(DT)';
86     //
87       [FP,DxiFP,DetaFP] = Q2(xg,yg);
88
89       for i=1:Nno
90         DFPi = B*[DxiFP(i); DetaFP(i)];
91         F_el(i) = F_el(i) + PG(ig)*PG(jg)*FP(i)*jac;
92         for j=1:Nno
93           DFPj = B*[DxiFP(j); DetaFP(j)];
94           A_el(i,j) = A_el(i,j) + PG(ig)*PG(jg)*( DFPj' * DFPi) *jac;
95         end;
96       end;
97     end;
98   end;
99 //
100 // fin calcul systeme elementaire pour l'element iel
101 //

```

```

102 //-----
103 //  FIN PARTIE A ECRIRE
104 //-----
105 // on doit assembler
106
107     for i=1:Nno
108         i_glob = ADRESS(iel,i);
109         F_glob(i_glob) = F_glob(i_glob) + F_el(i);
110     end;
111
112     A_el=sparse(A_el);
113     [ij,v,mn] = spget(A_el);
114     for i = 1:length(v)
115         ij(i,1) = ADRESS(iel,ij(i,1));
116         ij(i,2) = ADRESS(iel,ij(i,2));
117     end;
118     A_el = sparse(ij,v,[Npt Npt]);
119     A_glob = A_glob + A_el;
120
121 end; // fin boucle sur les elements
122
123
124 A = A_glob(1:NI,1:NI); F = F_glob(1:NI);
125 M12 = A_glob(1:NI,NI+1:$);
126 F = F - M12*UC;
127
128 [h,rk] = lufact(A);
129 U = lusolve(h,F);
130 ludel(h);
131
132 u_cal = [U; UC];
133
134 [u,x,y]=trace_u_2D(u_cal,COOR,NUMER,1,20,20);

```

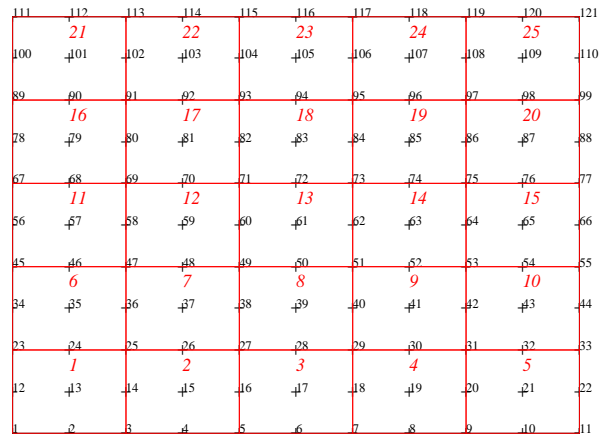


FIG. 1 – Exemple de maillage avec  $N_{elt} = 25$  et  $N_{pt} = 121$ .

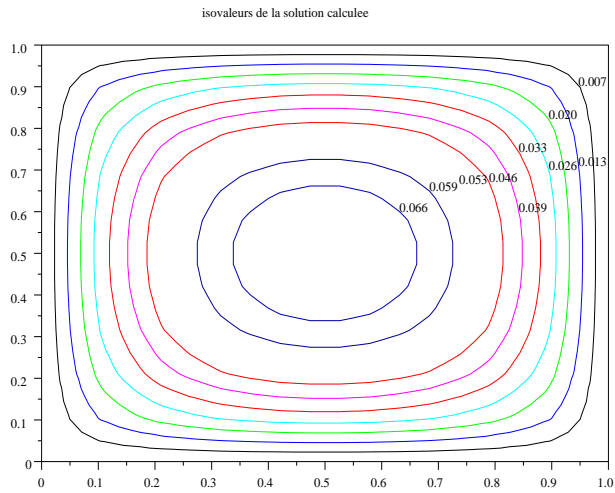


FIG. 2 – Solution du problème pour avec  $N_{elt} = 100$  et  $N_{pt} = 441$ .

4)

La formule qui donne le débit est :

$$q = \int_0^1 \int_0^1 u(x,y) dx dy = \sum_{k=1}^{N_{elt}} \left( \sum_{j=1}^{N_{no}} u_j^k \int_{-1}^1 L_j(\xi, \eta) Jac^k d\xi d\eta \right)$$

avec  $u|_{\mathbf{K}_k} = \sum_{j=1}^{N_{no}} u_j^k \psi_j^k(x, y)$ .

On trouve  $q = .0351$  avec le programme suivant :

```
function q = cal_debit_2D(U,COOR, CONNEC, NUMER, ADRESS)
//
[Nelt, Nno] = size(CONNEC);
Npt = length(COOR);

q = 0.;
for iel = 1:Nelt

    x = COOR(CONNEC(iel,:),1); y = COOR(CONNEC(iel,:),2);
    u_el = U(ADRESS(iel,:))

    for ig=1:Ng // double boucle pour l'integration numerique
        xg = XG(ig);
        for jg=1:Ng
            yg = XG(jg);
//
// calcul de la matrice jacobienne
//
            [F,DxiFP,DetaFP] = Q1(xg,yg);
            j11 = x(1:4)'*DxiFP;
            j12 = x(1:4)'*DetaFP;
            j21 = y(1:4)'*DxiFP;
            j22 = y(1:4)'*DetaFP;
//
            DT = [j11 j12; j21 j22];
            jac = det(DT);
//
            [FP,DxiFP,DetaFP] = Q2(xg,yg);
//
            for i=1:Nno
                q = q + PG(ig)*PG(jg) * u_el(i)*FP(i) * jac;
            end;
        end;
    end;
end; // fin boucle sur les elements

endfunction
```

5)

Il faut changer le calcul de la matrice NUMER à la ligne 28. Il faut "libérer" les degrés de liberté qui sont situés sur les bords  $x = 0$  et  $1$ . on remplace la condition

```

if x <> 0 & x <> 1 & y <> 0 & y <> 1 then
par
if y <> 0 & y <> 1 then

```

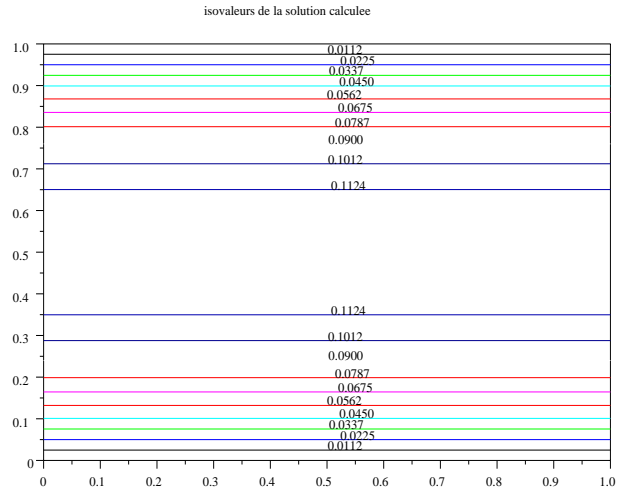


FIG. 3 – Solution de la question 5) pour avec  $N_{elt} = 100$  et  $N_{pt} = 441$ .

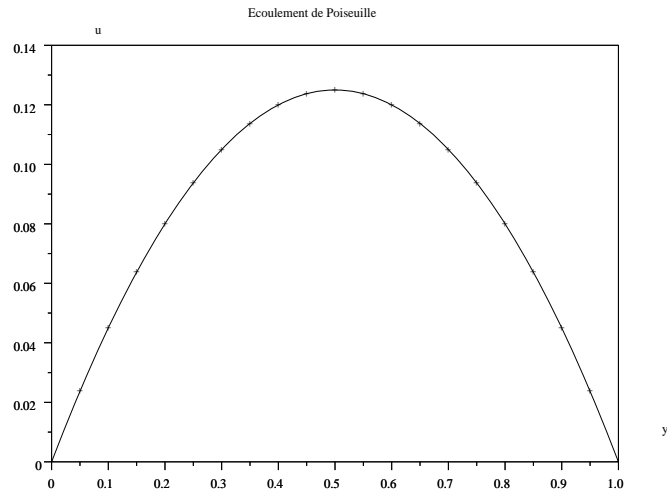


FIG. 4 – Comparaison entre la solution calculée (+) et la solution analytique  $u_e = y(y - 1)/2$  pour une valeur de  $x$  fixée.

**Ex 2** *Ecoulement analytique*

1)

La formulation faible

$$\int_0^1 \int_0^1 \left( \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} \right) dx dy + 3 \int_0^1 \int_0^1 u v dx dy = 0$$

2)

Le programme est essentiellement le même que pour l'exercice précédent. Il suffit d'imposer les conditions aux limites de Diriclet correctement

```
UC = zeros(Npt-NI,1);
for i=1:Npt
  x = COOR(i,1); y = COOR(i,2);
  if y == 0 | y == 1 | x == 0 | x == 1 then
    UC(NUMER(i)-NI) = exp(2*x) * sin(y);
  end;
end;
```

et de calculer correctement la matrice élémentaire

```
A_el(i,j) = A_el(i,j) + ...
  PG(ig)*PG(jg)*( DFPj(1)*DFPi(1) + DFPj(2)*DFPi(2) + 3*FP(j)*FP(i) ) *jac;
```

Si on utilise des éléments  $\mathbf{P}_1$ , il faut utiliser la quadrature de Hammer et la fonction `hammer.sci` pour avoir les points d'intégration.

Quelques exemples de calcul sont rassemblés dans le tableau suivant :

	Npt	Nelt	Nno	$\ u_c - u_a\ /\ u_a\ $
$\mathbf{Q}_1$	81	64	4	$5.05 \cdot 10^{-4}$
$\mathbf{Q}_2$	81	16	9	$1.40 \cdot 10^{-5}$
$\mathbf{P}_1$	81	128	3	$5.15 \cdot 10^{-4}$

TAB. 1 – Erreur relative  $\|u_c - u_a\|/\|u_a\|$  en fonction des éléments choisis.



