Fonctions de forme et jacobiens : solutions

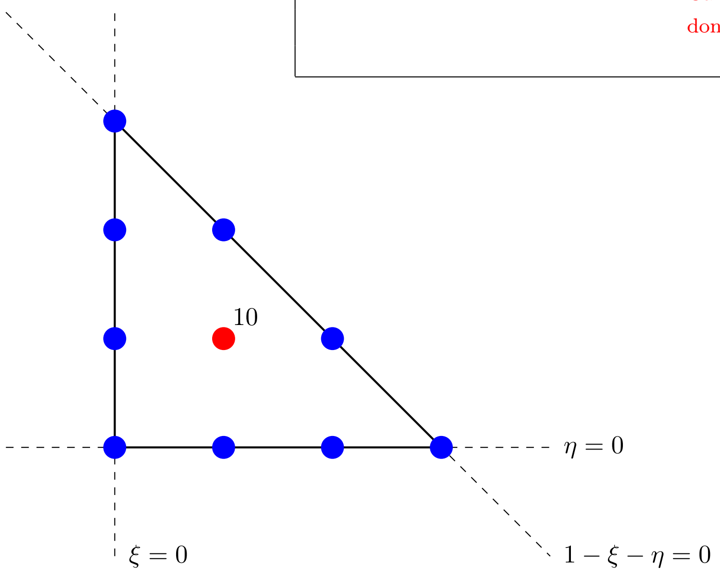
4 A titre d'exemple, calculons la fonction de forme ϕ_{10}.

On obtient très aisément le polynôme en multipliant les équations implicites des trois droites qui définissent le triangle parent. Il reste juste à trouver la constante multiplicative afin que la fontion de forme soit unitaire pour le noeud auquel elle est associée.

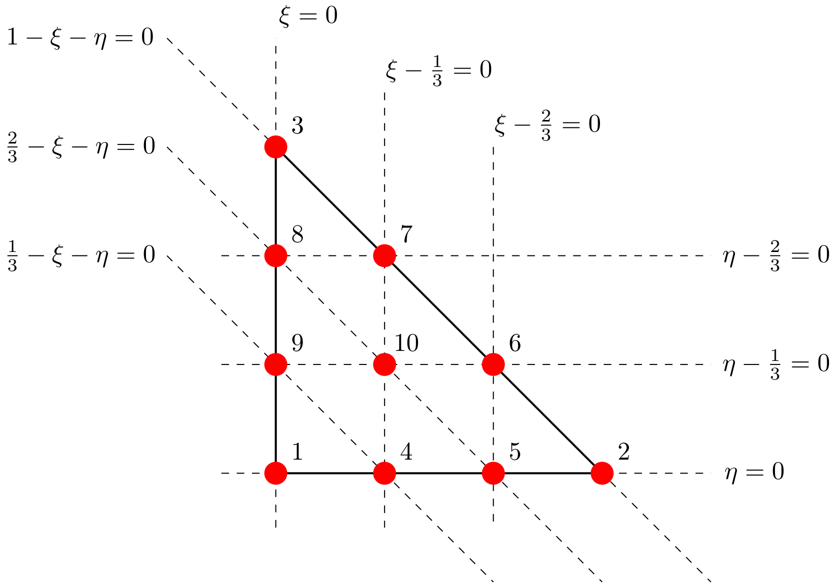
On écrit ainsi :

$$\phi_{10}(\xi,\eta) \quad = \quad \eta\ \xi\ \Big(1-\xi-\eta\Big)\quad 27$$

Car évaluer les 3 droites en $(\frac{1}{3},\frac{1}{3})$
donne $\frac{1}{3}\times\frac{1}{3}\times\Big(1-\frac{2}{3}\Big)=\frac{1}{27}$



Pour les autres fonctions, il faut tracer toutes les droites et effectuer les bonnes combinaisons !

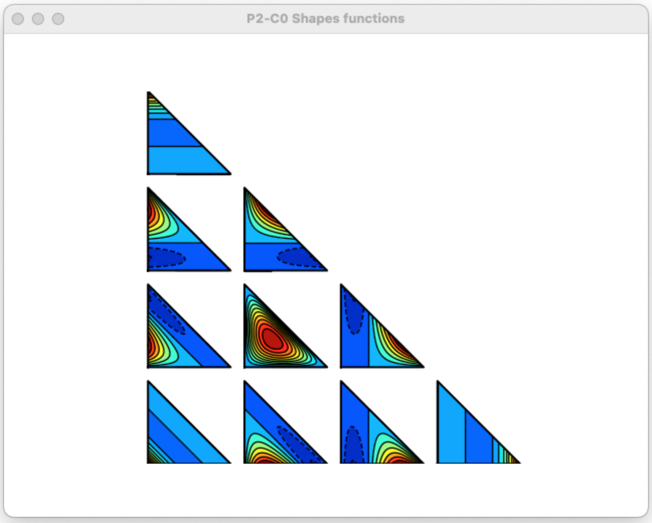


On conclut donc :

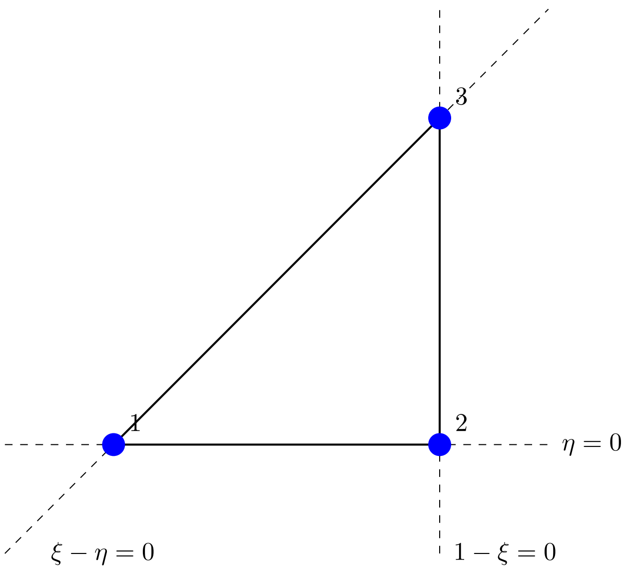
$$\boxed{\begin{array}{rclrrrr}
\phi_1(\xi,\eta) & = & \left(\frac{1}{3}-\xi-\eta\right) & \left(\frac{2}{3}-\xi-\eta\right) & \left(1-\xi-\eta\right) & \frac{9}{2} \\
\phi_2(\xi,\eta) & = & \xi & \left(\xi-\frac{1}{3}\right) & \left(\xi-\frac{2}{3}\right) & \frac{9}{2} \\
\phi_3(\xi,\eta) & = & \eta & \left(\eta-\frac{1}{3}\right) & \left(\eta-\frac{2}{3}\right) & \frac{9}{2} \\
\phi_4(\xi,\eta) & = & \xi & \left(\frac{2}{3}-\xi-\eta\right) & \left(1-\xi-\eta\right) & \frac{27}{2} \\
\phi_5(\xi,\eta) & = & \xi & \left(\xi-\frac{1}{3}\right) & \left(1-\xi-\eta\right) & \frac{27}{2} \\
\phi_6(\xi,\eta) & = & \xi & \eta & \left(\xi-\frac{1}{3}\right) & \frac{27}{2} \\
\phi_7(\xi,\eta) & = & \xi & \eta & \left(\eta-\frac{1}{3}\right) & \frac{27}{2} \\
\phi_8(\xi,\eta) & = & \eta & \left(\eta-\frac{1}{3}\right) & \left(1-\xi-\eta\right) & \frac{27}{2} \\
\phi_9(\xi,\eta) & = & \eta & \left(1-\xi-\eta\right) & \left(\frac{2}{3}-\xi-\eta\right) & \frac{27}{2} \\
\phi_{10}(\xi,\eta) & = & \eta & \xi & \left(1-\xi-\eta\right) & 27
\end{array}}$$

- Une implémentation possible est fournie par :

```
from numpy import *
from numpy.linalg import solve
import matplotlib
import matplotlib.pyplot as plt
myColorMap = matplotlib.cm.jet

# ==========================================================================

def shapeComputeCoefficient(nodes):
  n = len(nodes)
  B = diag(ones(n))
  A = zeros([n,n])
  for i in range(0,n):
    xi  = nodes[i,0]
    eta = nodes[i,1]
    A[i,:] = [1, xi, eta, xi*xi, xi*eta, eta*eta, xi**3, xi*xi*eta, xi*eta*eta, eta**3 ]
  return transpose(solve(A,B))

# ==========================================================================

def shapeCompute(phi,xi,eta):
  return (phi[0] + phi[1]*xi + phi[2]*eta
                 + phi[3]*xi**2 + phi[4]*xi*eta + phi[5]*eta**2
                 + phi[6]*xi**3 + phi[7]*eta*xi**2 + phi[8]*xi*eta**2
                 + phi[9]*eta**3);

# ==========================================================================

# -1- Computing coefficients

nodes = array([[0,0],[1,0],[0,1],
              [1/3,0],[2/3,0],[2/3,1/3],
              [1/3,2/3],[0,2/3],[0,1/3],[1/3,1/3]])
n = len(nodes)
coeff = shapeComputeCoefficient(nodes)

# -2- Checks if the sum is equat to one
#     for a random point (0.2;0.4)

sumPhi = 0;
for i in range(0,n):
  X = 0.2; Y = 0.4
  sumPhi += shapeCompute(coeff[i,:],X,Y)
print(" ==== Check of the sum of shape fuctions : %14.7e ===== " % sumPhi)

# -3- Nice plot

plt.figure("P2-C0 Shapes functions")
for i in range(0,n):
  Xg,Yg = meshgrid(linspace(0,1,100),linspace(0,1,100))
  Xt = [0,0,1,0];   Yt = [0,1,0,0];
  U = shapeCompute(coeff[i,:],Xg,Yg)
  U[Xg+Yg>1] = nan
  Xg = Xg + 3.5*nodes[i,0];
  Yg = Yg + 3.5*nodes[i,1];
  Xt = Xt + 3.5*nodes[i,0];
  Yt = Yt + 3.5*nodes[i,1];
  plt.contourf(Xg,Yg,U,10,cmap=myColorMap,vmin=-0.3,vmax=1)
  plt.contour(Xg,Yg,U,10,colors='k',linewidths=1)
  plt.plot(Xt,Yt,'-k');
plt.axis("equal"); plt.axis("off")
plt.show()
```



5 Il suffit de faire le même procédé que pour le triangle parent usuel :-)

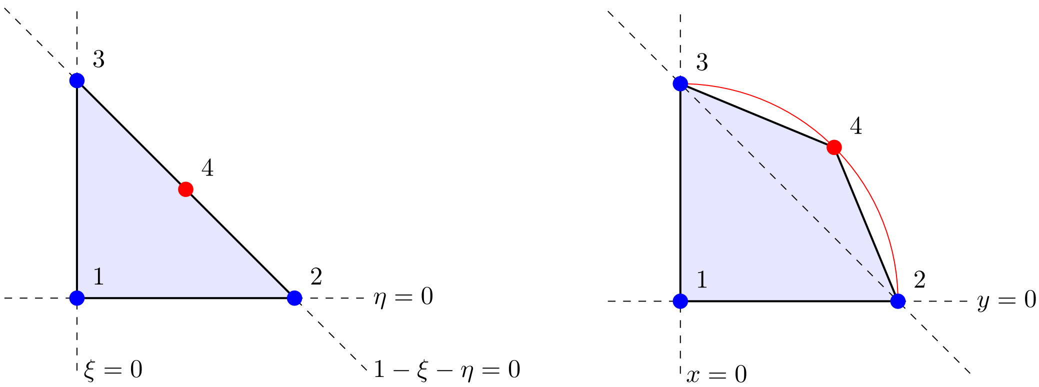


Et on écrit immédiatement : $$\boxed{\begin{array}{rcl} \phi_1(\xi,\eta) & = & 1-\xi \\ \phi_2(\xi,\eta) & = & \xi-\eta \\ \phi_3(\xi,\eta) & = & \eta \end{array}}$$

On peut aussi observer immédiatement que la somme des trois fonctions vaut toujours un :-)

6 Pour ceux qui se posent la question, ceci pourrait être une très jolie question d'examen !

1. Même si cela semble élémentaire, on observer en pratique que pas mal d'étudiants éprouvent des difficultés pour tracer l'élément parent $\widehat{\Omega}$ et l'élément Ω :-)



   On voit donc qu'on approche aussi le quart de cercle par un quadrilatère et qu'on va donc estimer la surface du quart de cercle par celle de ce quadrilatère.

2. Pour obtenir les quatre fonctions de forme, on part des trois fonctions de forme linéaire. Ensuite, on obtient la fonction de forme associée au quatrième noeud qui s'annule aux 3 autres noeuds et vaut l'unité pour ce quatrième noeud. Ensuite, on calcule le terme $\xi\eta$ à ajouter aux trois autres fonctions de forme pour qu'elles s'annulent au quatrième noeud.

   Et on peut facilement en déduire :
$$\boxed{\begin{array}{rcl} \phi_1(\xi,\eta) & = & 1-\xi-\eta \\ \phi_2(\xi,\eta) & = & \xi-2\xi\eta \\ \phi_3(\xi,\eta) & = & \eta-2\xi\eta \\ \phi_4(\xi,\eta) & = & 4\xi\eta \end{array}}$$

   Ici, il est vachement utile de vérifier que $\sum_{i=1}^{4} \phi_i(\xi,\eta) = 1$.

3. Pour obtenir les dérivées, il suffit tout simplement de dériver :

   On écrit ensuite :
$$\boxed{\begin{array}{rcl|rcl} \phi_{1,\xi}(\xi,\eta) & = & -1 & \phi_{1,\eta}(\xi,\eta) & = & -1 \\ \phi_{2,\xi}(\xi,\eta) & = & 1-2\eta & \phi_{2,\eta}(\xi,\eta) & = & -2\xi \\ \phi_{3,\xi}(\xi,\eta) & = & -2\eta & \phi_{3,\eta}(\xi,\eta) & = & 1-2\xi \\ \phi_{4,\xi}(\xi,\eta) & = & 4\eta & \phi_{4,\eta}(\xi,\eta) & = & 4\xi \end{array}}$$

   A nouveau, il est vachement utile de vérifier que $\sum_{i=1}^{4} \phi_{\xi,i}(\xi,\eta) = \sum_{i=1}^{4} \phi_{\eta,i}(\xi,\eta) = 0$.

4. Pour le jacobien de la transformation, il faut juste écrire :

$$
\begin{aligned}
J(\xi,\eta) &= \det \begin{bmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \end{bmatrix} \\
&= \det \begin{bmatrix} 1-2\eta+2\sqrt{2}\eta & -2\xi+2\sqrt{2}\xi \\ -2\eta+2\sqrt{2}\eta & 1-2\xi+2\sqrt{2}\xi \end{bmatrix} \\
&= \Big(1-2\eta+2\sqrt{2}\eta\Big)\Big(1-2\xi+2\sqrt{2}\xi\Big) - \Big(-2\eta+2\sqrt{2}\eta\Big)\Big(-2\xi+2\sqrt{2}\xi\Big) \\
&= 1-2\eta+2\sqrt{2}\eta-2\xi+2\sqrt{2}\xi
\end{aligned}
$$

5. Comme le jacobien est linéaire, une règle d'intégration avec une seul point permettra d'intégrer exactement la surface !
   On prend le centre de gravité du triangle situé en $(\xi_1, \eta_1) = (\frac{1}{3}, \frac{1}{3})$.
   Le poids associé est $w_1 = \frac{1}{2}$.

6. Il suffit de calculer l'intégrale :-)

$$
\begin{aligned}
\frac{\pi}{4} \approx \int_\Omega dx\, dy &= \int_{\widehat{\Omega}} J(\xi,\eta)\, d\xi\, d\eta \\
&= \frac{1}{2}\, J\left(\frac{1}{3},\frac{1}{3}\right) \\
&= \frac{1}{2}\, \left(1-\frac{4}{3}+\frac{4\sqrt{2}}{3}\right) \\
&= \frac{1}{2}\, \left(\frac{4\sqrt{2}}{3}-\frac{1}{3}\right)
\end{aligned}
$$

On conclut donc : $\boxed{\pi \approx \frac{2}{3}\Big(4\sqrt{2}-1\Big) = 3.1046}$

C'est une approximation qui est assez acceptable finalement :-)