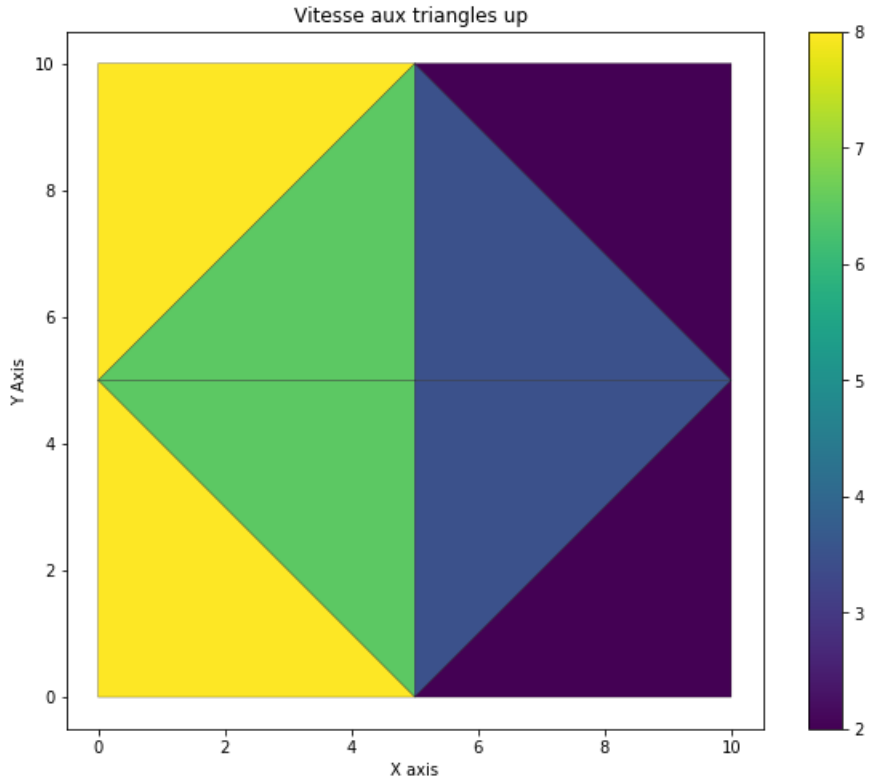
```python
# -*- coding: utf-8 -*-
"""
MEC6616 - Hiver 2020 - mars 2020

@authors: Jean-Yves Trépanier

Programme test pour Correction de la pression
LAPP6-2020

"""
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.tri as mtri
import mec6616
import mec6616_private as m6p

#-----------------------------------------------------------------------
#test de correction de pression (LAPP6)
#-----------------------------------------------------------------------
#domaine rectangle
xmin=0
xmax=10
ymin=0
ymax=10
Nx=2
Ny=2
#Test de RectMesh
xx,yy,tri,are = mec6616.RectMesh(xmin,xmax,ymin,ymax,Nx,Ny)

xc,yc = m6p.TriCenter(xx,yy,tri)
#metriques des arêtes
DAI,DXI,PNXI,PXIET,NAx,NAy,DXAP,DYAP = m6p.AreMetric(xx,yy,tri,are)

#triangulation matplotlib.tri pour graphes
MTri2 = mtri.Triangulation(xx, yy,tri)   #objet Triangulation pour figures

#initialisation pour les vitesses et autres parametres
x,y= sp.symbols('x y')
u = 10.0 - x
v = 0. + 0.*y
rho=1.
gama = 0.1

#conversion d'une expression en une fonction appelable et évaluable
fu = sp.lambdify([x,y], u, 'numpy')
fv = sp.lambdify([x,y], v, 'numpy')

#centre des aretes
xca = 0.5*(xx[are[:,0]]+xx[are[:,1]])
yca = 0.5*(yy[are[:,0]]+yy[are[:,1]])

#vitesse débitante nulle pour l'appel de convection-diffusion
Un=np.zeros(are.shape[0])
sch='Upwind'
Sx = 0.0       #terme source de pression nul
```

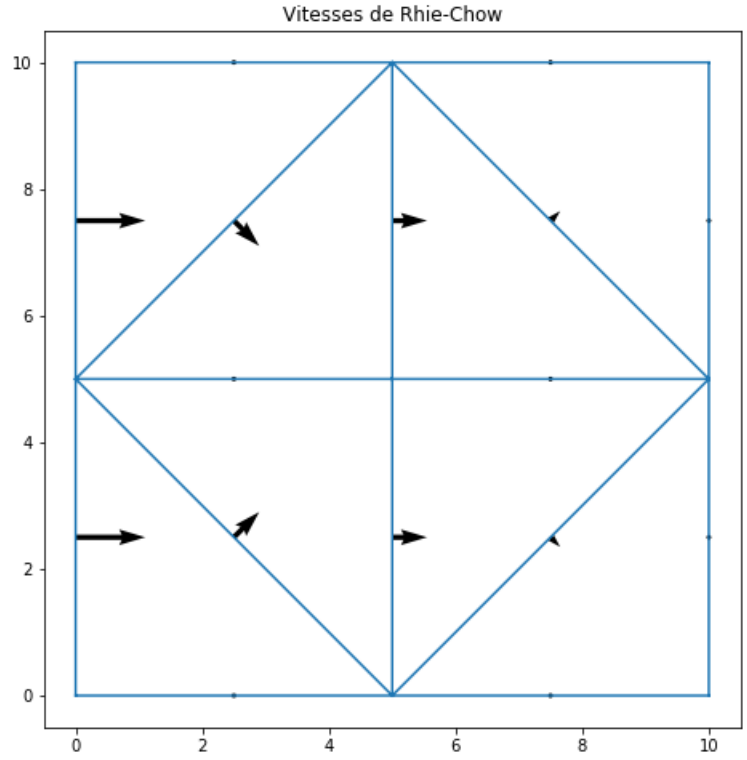
```
#codage des conditions limites - tuple de liste
bcdu=(['Dirichlet',fu],['Dirichlet',fu],['Dirichlet',fu],['Dirichlet',fu])
#appel de ConvectionDiffusion pour obtenir la matrice AS et les vitesses up
up,AS=m6p.TriConvectionDiffusion(xx,yy,tri,are,bcdu,rho,Un,gama,sch,Sx)

#pas besoin de vitesse v pour ce test
vp = np.zeros(tri.shape[0])
bcdv=(['Dirichlet',0],['Dirichlet',0],['Dirichlet',0],['Dirichlet',0])

#plot triangles with one color per triangle for solution up
fig9, ax9 = plt.subplots()
tcf = ax9.tripcolor(MTri2, facecolors=up, edgecolors='k')
fig9.colorbar(tcf)
ax9.set_title('Vitesse aux triangles up ')
ax9.set_xlabel('X axis')
ax9.set_ylabel('Y Axis')
ax9.axes.set_aspect('equal')
plt.show()

print('Sortie de convection diffusion - Matrice AS')
print('AS ',AS.todense())

#vitesses de Rhie-Chow
pre = np.zeros(tri.shape[0])
bcdp=(['Libre',0],['Neumann',0],['Dirichlet',0],['Neumann',0])
grpre = m6p.TriGradientLS(xx,yy,tri,are,bcdp,pre)
#pre et grpre sont nuls pour ce test

UnRC = m6p.TriRhieChow(xx,yy,tri,are,up,vp,bcdu,bcdv,pre,grpre,AS)

#visualisation des vitesses de Rhie-Chow
#composantes de la vitesse débitante
unx = UnRC * NAx
uny = UnRC * NAy
#tracage d'un champ de vecteur au milieu des aretes
plt.figure()
plt.triplot(MTri2)
plt.axes().set_aspect('equal')
plt.axes().set_title('Vitesses de Rhie-Chow')
plt.quiver(xca,yca,unx,uny,scale=100 )
plt.show

#solution de la correction de la pression
bcdpc=(['Entree',0],['Paroi',0],['Sortie',0],['Paroi',0])
PPrime,UF=m6p.TriCorrectionPression(xx,yy,tri,are,bcdpc,AS,UnRC)

#plot triangles with one color per triangle for solution Solx
fig9, ax9 = plt.subplots()
tcf = ax9.tripcolor(MTri2, facecolors=PPrime, edgecolors='k')
fig9.colorbar(tcf)
ax9.set_title('Correction de pression ')
ax9.set_xlabel('X axis')
ax9.set_ylabel('Y Axis')
ax9.axes.set_aspect('equal')
plt.show()
```

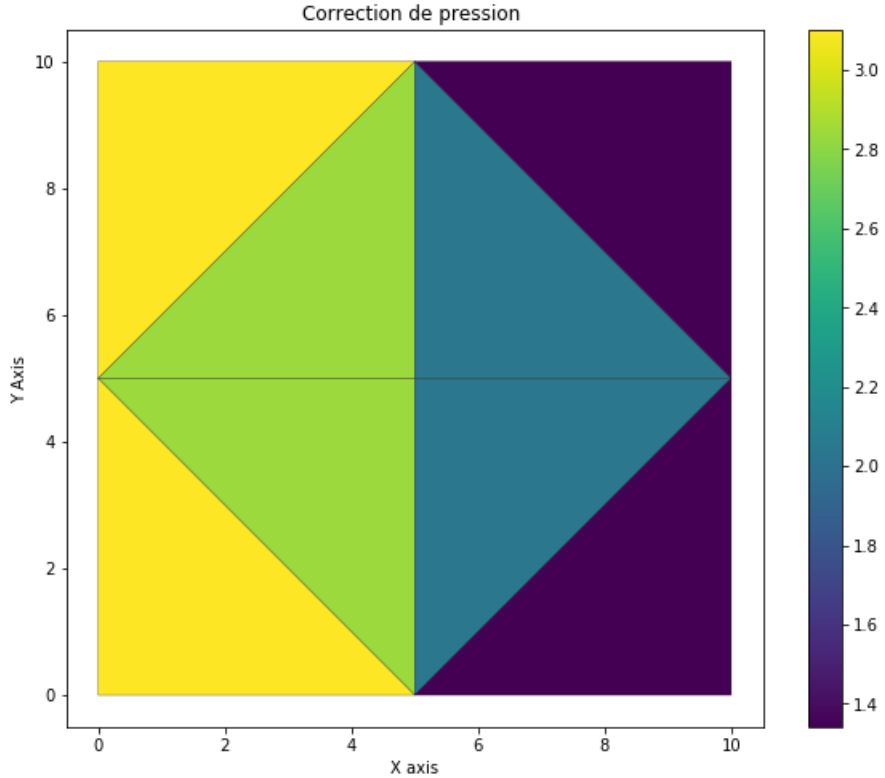
```
#composantes de la vitesse débitante après la correction
ufx = UF * NAx
ufy = UF * NAy

#tracage d'un champ de vecteur au milieu des aretes
plt.figure()
plt.triplot(MTri2)
plt.axes().set_aspect('equal')
plt.axes().set_title('Vitesses après la correction')
plt.quiver(xca,yca,ufx,ufy,scale=100 )
plt.show

Div_avant = m6p.TriDivergence(xx,yy,tri,are,UnRC)
Div_apres = m6p.TriDivergence(xx,yy,tri,are,UF)

print('divergence avant',Div_avant)
print('divergence apres',Div_apres)
```

```
In [66]: runfile('C:/Users/Jean-Yves Trépanier/OneDrive/CoursPoly/MEC6616/Python/
TestCorrectionPression-LAPP6.py', wdir='C:/Users/Jean-Yves Trépanier/OneDrive/CoursPoly/
MEC6616/Python')
Reloaded modules: mec6616, mec6616_private
```

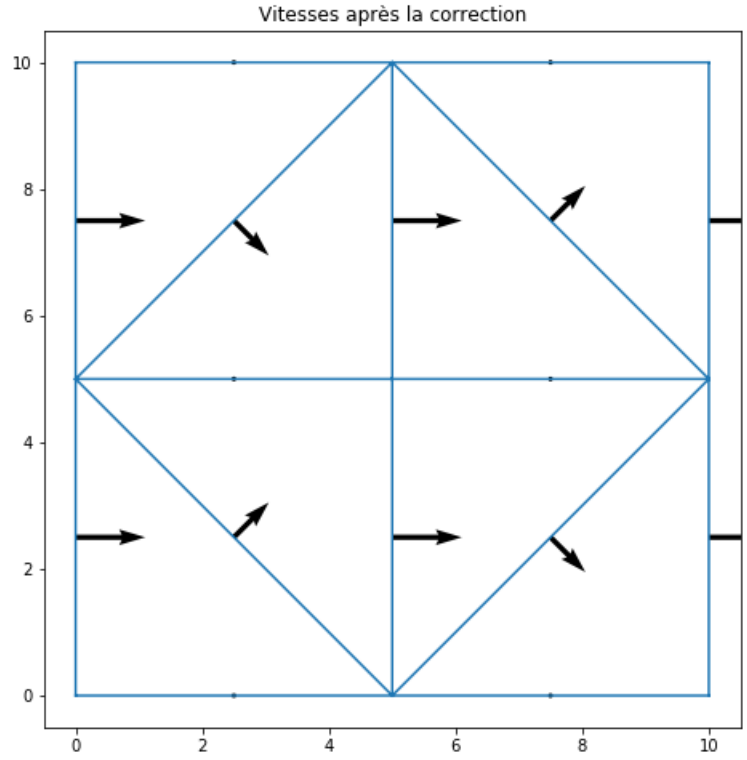



```
Sortie de convection diffusion - Matrice AS
AS  [[ 0.6  -0.3  -0.15  0.   -0.15  0.    0.    0.  ]
 [-0.3   0.9   0.    0.    0.    0.    0.    0.  ]
 [-0.15  0.    0.6  -0.3   0.    0.   -0.15  0.  ]
 [ 0.    0.   -0.3   0.9   0.    0.    0.    0.  ]
 [-0.15  0.    0.    0.    0.6  -0.3  -0.15  0.  ]
 [ 0.    0.    0.    0.   -0.3   0.9   0.    0.  ]
 [ 0.    0.   -0.15  0.   -0.15  0.    0.6  -0.3 ]
 [ 0.    0.    0.    0.    0.    0.   -0.3   0.9 ]]
```






```
divergence avant [11.25 13.75 11.25 13.75 11.25 13.75 11.25 13.75]
divergence apres [ 2.87547763e-14 -7.10542736e-15  2.13162821e-14  1.42108547e-14
  4.97379915e-14  2.13162821e-14  1.42108547e-14  0.00000000e+00]
```



In [67]: