

#extrait du code Python TriSimple – Mars 2020

**for iter in range(1,MaxIter+1):**

#solution des deux equations du momentum

**up,AS=m6p.TriConvectionDiffusion(xx,yy,tri,are,bcdv,rho,Un,gama,sch,Sx)**

**vp,AS=m6p.TriConvectionDiffusion(xx,yy,tri,are,bcdv,rho,Un,gama,sch,Sy)**

#Moyenne des vitesses de Rhie-Chow

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

#sous relaxation de Rhie-Chow

**UnRC = Un + AlphaRC\*(UnRC-Un)**

#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)**

#divergence pour impression et/ou contrôle

**Div\_avant = m6p.TriDivergence(xx,yy,tri,are,UnRC)**

#Div\_apres = m6p.TriDivergence(xx,yy,tri,are,UF)

#correction du champ de pression sur les triangles

**pre += AlphaP \* PPrime**

#Gradient de pression et vitesse normale pour la prochaine itération

**bcdp=(['Libre',0],['Neumann',0],['Dirichlet',0],['Neumann',0])**

**grpre = m6p.TriGradientLS(xx,yy,tri,are,bcdp,pre)**

**Sx = -grpre[:,0]**

**Sy = -grpre[:,1]**

**Un = UF**

#impression de la divergence sur le domaine de calcul

**DivTri[iter-1] = np.sum(np.abs(Div\_avant))**

**print('iter, divergence avant',iter, DivTri[iter-1])**