Yasser Badrezzamane 3A EMSYS Chaymaa Rifar

Mini Projet

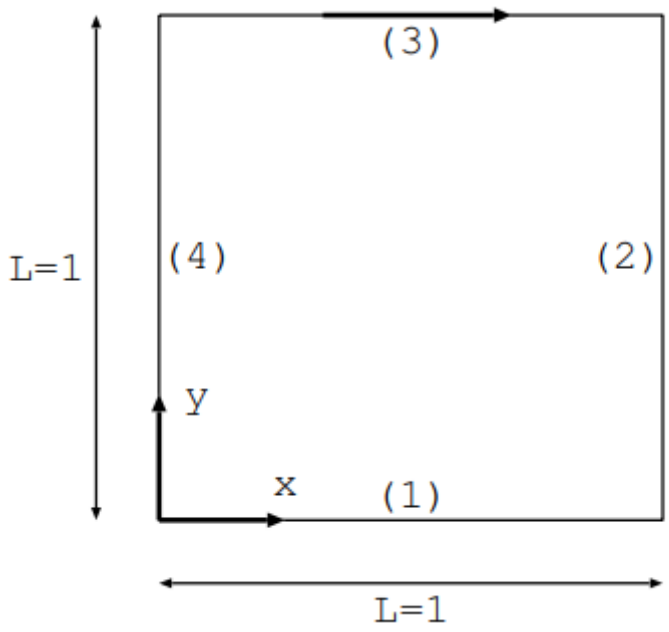
Ecoulement de fluide incompressible dans une cavité carrée

Méthodes numériques Navier Stockes

Professeur : Mathieu Jenny

Objectifs du mini projet:

Ecrire un code Matlab pour résoudre les équations numériques afin de calculer l'écoulement d'un fluide Newtonien incompressible contenu dans une cavité carrée en 2D, et entraîné par la paroi mobile (3), un tapis roulant qui entraîne le fluide à une vitesse: vitesse $\vec{u} = u_p \overrightarrow{e_x}$
on considère dans notre cas que $u_p = 1$ on aura donc dans la cavité une recirculation du fluide sous forme d'un tourbillon.



Mise en forme et résolution à l'ordre 1 en temps

1 Équations de Navier-Stokes

On écrit les conditions aux limites : le fluide est incompressible, les CL seront sur la vitesse uniquement, pas de contrainte sur la pression.

$\vec{u} = 0$ sur les parois (1),(2) et (4)
$\vec{u} = u_p \neq 0$ suivant $\overrightarrow{e_x}$ sur la paroi (3) avec u_p: $la\ vitesse$ à $la\ paroi$

On écrit les équations de Navier-Stokes et les équations de conservation de la masse dans le cas d'un fluide Newtonien incompressible

$$\begin{cases} \rho(\frac{\partial \vec{u}}{\partial t} + (\vec{u}\nabla)\overrightarrow{u)} = -\nabla\vec{p} + \frac{1}{Re}\Delta\vec{u} \\ \mathrm{div}(\overrightarrow{u}) = 0 \end{cases}$$

Quand on normalise : ρ=1 et μ=1/Re

Les équations deviennent :

$$\begin{cases} \frac{\partial \vec{u}}{\partial t} + (\vec{u}\nabla)\vec{u} = -\nabla\vec{p} + \frac{1}{Re}\Delta\vec{u} \\ \mathrm{div}(\overrightarrow{u}) = 0 \end{cases}$$

La vitesse s'écrit sous la forme $\vec{u} = u.\overrightarrow{ex} + v.\overrightarrow{ey}$
Les équations projetées :

$$\begin{cases} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial \hat{p}}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial \hat{p}}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \end{cases}$$

2 Discrétisation temporelle

Le temps t est discrétisé par un ensemble d'instants t_n séparés par un pas de temps Δt. Nous allons écrire les équations à l'instant t_{n+1} en considérant que les champs de vitesse et de pression sont connus aux instants précédents.
Afin de simplifier l'écriture, on pose $\vec{u}$ (tn+1) = $\vec{u}^{(n+1)}$

$$\frac{\partial \vec{u}^{(n+1)}}{\partial t} + (\vec{u}\nabla)\,\vec{u}^{(n+1)} = -\nabla\vec{p}^{(n+1)} + \frac{1}{Re}\Delta\,\vec{u}^{(n+1)}$$

On discrétise la dérivée temporelle :

$$\left.\frac{\partial u}{\partial t}\right|_{t_{n+1}} = \frac{\vec{u}^{n+1} - \vec{u}^{n}}{\Delta t} + O(\Delta t)$$

On peut donc montrer que :

$$(\vec{u}\vec{\nabla})\vec{u}|_{t_{n+1}} = (\vec{u}\vec{\nabla})\vec{u}|_{t_n} + O(\Delta t)$$

Pour résoudre une équation non linéaire on utilise la méthode de Newton, on linéarise le système par rapport à une vitesse initiale, puis on calcule la Jacobéenne et on itère n fois à chaque pas de temps jusqu'à convergence du système, le schéma devient moins stable.

Ainsi, en remplaçant à gauche le terme convectif par $(\vec{u}\vec{\nabla})\vec{u}|_{t_{n+1}}$ et à droite les champs de vitesse connus à l'instant t_n et le champ de pression à l'instant t_{n+1} , on aura :

$$\begin{cases} \frac{u^{n+1}}{\Delta t} - \frac{1}{Re}\left(\frac{\partial^2 u^{n+1}}{\partial x^2} + \frac{\partial^2 u^{n+1}}{\partial y^2}\right) = -\frac{\partial \hat{p}^{n+1}}{\partial x} + \frac{u^n}{\Delta t} - u^n \frac{\partial u^n}{\partial x} - v^n \frac{\partial u^n}{\partial y} \\ \frac{v^{n+1}}{\Delta t} - \frac{1}{Re}\left(\frac{\partial^2 v^{n+1}}{\partial x^2} + \frac{\partial^2 v^{n+1}}{\partial y^2}\right) = -\frac{\partial \hat{p}^{n+1}}{\partial x} + \frac{v^n}{\Delta t} - u^n \frac{\partial v^n}{\partial x} - v^n \frac{\partial v^n}{\partial y} \\ \frac{\partial u^n}{\partial x} + \frac{\partial v^n}{\partial y} = 0 \end{cases}$$

3 Discrétisation spatiale

Pour la discrétisation spatiale on utilisera la méthode des différences finies. La valeur des champs de pression et de vitesse seront discrétisés par rapport à ces points. Les vitesses sont prises aux points d'indice entier et la pression aux points d'indice demi-entier. Les pas d'espace dans la direction x et y sont respectivement Δx et Δy. Les inconnues du problème sont donc les valeurs de vitesse aux points (xi , yj) et de pression aux points (xi+1/2, yj+1/2).

Pour écrire les matrices des opérateurs de dérivation, nous avons intérêt à les dénir par blocs. Ces matrices ne s'appliquent pas aux limites.

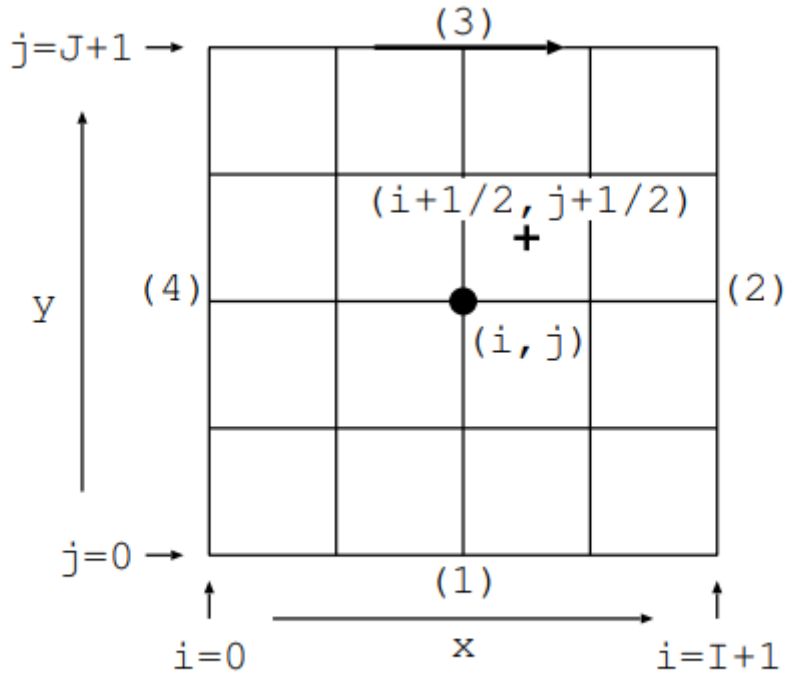


FIGURE 1.1 – Maillage avec positions d'indice demi-entier (+).

Opérations de dérivée pour les termes convectifs et le Laplacien

$$\frac{\partial u}{\partial x}(i,j) = \frac{u(i+1,j) - u(i-1,j)}{2\Delta x} + O(\Delta x^2)$$

$$\frac{\partial u}{\partial y}(i,j) = \frac{u(i,j+1) - u(i,j-1)}{2\Delta y} + O(\Delta y^2)$$

$$\frac{\partial^2 u}{\partial x^2}(i,j) = \frac{u(i+1,j) - 2u(i,j) + u(i-1,j)}{\Delta x^2}$$

$$\frac{\partial^2 u}{\partial y^2}(i,j) = \frac{u(i,j+1) - 2u(i,j) + u(i,j-1)}{\Delta y^2}$$

Pour les points de pression :

$$\frac{\partial \vec{P}}{\partial x} = \frac{P(i+1/2,j) - P(i-1/2,j)}{\Delta x}$$

$$\frac{\partial \vec{P}}{\partial x}(i,j) = \frac{P\left(i+\frac{1}{2},j+\frac{1}{2}\right) - P\left(i-\frac{1}{2},j+\frac{1}{2}\right)}{2\Delta x} + \frac{P\left(i+\frac{1}{2},j-1/2\right) - P(i-\frac{1}{2},j-1/2)}{2\Delta x}$$

Pour la divergence :

$$\frac{\partial u}{\partial x}(i-\tfrac{1}{2},j-1/2) = \frac{u(i,j)-u(i-1,j)}{2\Delta x} + \frac{u(i,j-1)-u(i-1,j-1)}{2\Delta x}$$

Nos équations seront sous la forme :

$$\left[u_{i-1,j}\left(\frac{1}{Re\Delta x^2}\right) + u_{i,j}\left(\frac{1}{\Delta t} + \frac{2}{Re\Delta x^2} + \frac{1}{Re\Delta y^2}\right) + u_{i+1,j}\left(-\frac{1}{Re\Delta x^2}\right) + u_{i,j-1}\left(-\frac{1}{Re\Delta y^2}\right) + u_{i,j+1}\left(-\frac{1}{Re\Delta y^2}\right)\right]^{n+1} = -\frac{\hat{p}^{n+1}{}_{i+1,j} - \hat{p}^{n+1}{}_{i-1,j}}{2\Delta x} - {u_{i,j}}^n\left(\frac{{u_{i+1,j}}^n - {u_{i-1,j}}^n}{2\Delta x} - \frac{1}{\Delta t}\right) - {v_{i,j}}^n\frac{{u_{i,j+1}}^n - {u_{i,j-1}}^n}{2\Delta y}$$

On va intégrer les conditions au limites avant de passer aux calcul matriciel :

Pour les conditions de Dirichlet : on impose la vitesse (uniquement, parce que le fluide est incompressible)
Pour les conditions de Newman : On impose a dérivée

$$Pour \begin{cases} i = 1 \\ 1 < j < J \end{cases}$$

pour i=1

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{1,j} = \frac{-2u_{1,j} + u_{1,j}}{\Delta x^2}$$

pour j=1

$$\left.\frac{\partial^2 u}{\partial y^2}\right|_{i,1} = \frac{-2u_{i,1}+u_{1,2}}{\Delta y^2}$$

pour j=J

$$\left.\frac{\partial^2 u}{\partial y^2}\right|_{i,J} = \frac{u_{i,J-1}-2u_{i,J}+u_p}{\Delta y^2}$$

4 Mise en forme matricielle des équations discrétisées

Nous allons donc écrire les opérateurs de dérivation sous forme matricielle, de la forme :

$$\begin{cases} AU = -G_x P + B_x \\ AV = -G_y P + B_y \\ Div_x U + Div_y V = 0 \end{cases}$$

Avec Gx la matrice relative au gradient pour u, Px la matrice relative à la pression pour u, Bx la matrice explicite en chaque point pour u. La matrice colonne U définie par bloc de toutes les vitesses suivant ex :

$$U = \begin{bmatrix} u(1,1) \\ \vdots \\ u(I,1) \\ \vdots \\ u(1,J) \\ \vdots \\ u(I,J) \end{bmatrix}$$

Pareil pour V

La matrice Dx, représentant la matrice dérivée par x sera construite par blocs de la façon suivante

$$D_x = \frac{1}{2\Delta x}\begin{bmatrix} \begin{matrix} 0 & 1 & \\ -1 & \ddots & 1 \\ & -1 & 0 \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} \\ \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} & \begin{matrix} 0 & 1 & \\ -1 & \ddots & 1 \\ & -1 & 0 \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} \\ \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} & \begin{matrix} 0 & 1 & \\ -1 & \ddots & 1 \\ & -1 & 0 \end{matrix} \end{bmatrix}$$

La matrice D2x, représentant la dérivée seconde par rapport à x va s'écrire de la façon suivante :

$$D2_x = \frac{1}{\Delta x^2}\begin{bmatrix} \begin{matrix} -2 & 1 & \\ 1 & \ddots & 1 \\ & 1 & -2 \end{matrix} & \begin{matrix} \\ 0 \\ \\ \end{matrix} & \begin{matrix} \\ 0 \\ \\ \end{matrix} \\ \begin{matrix} \\ 0 \\ \\ \end{matrix} & \begin{matrix} -2 & 1 & \\ 1 & \ddots & 1 \\ & 1 & -2 \end{matrix} & \begin{matrix} \\ 0 \\ \\ \end{matrix} \\ \begin{matrix} \\ 0 \\ \\ \end{matrix} & \begin{matrix} \\ 0 \\ \\ \end{matrix} & \begin{matrix} -2 & 1 & \\ 1 & \ddots & 1 \\ & 1 & -2 \end{matrix} \end{bmatrix}$$

$$D_y = \frac{1}{2\Delta y}\begin{bmatrix} \begin{matrix} 0 \\ \\ \end{matrix} & \begin{matrix} I_d \\ \\ \end{matrix} & \begin{matrix} 0 \\ \\ \end{matrix} \\ \begin{matrix} -I_d \\ \\ \end{matrix} & \begin{matrix} 0 \\ \\ \end{matrix} & \begin{matrix} I_d \\ \\ \end{matrix} \\ \begin{matrix} 0 \\ \\ \end{matrix} & \begin{matrix} -I_d \\ \\ \end{matrix} & \begin{matrix} 0 \\ \\ \end{matrix} \end{bmatrix}$$

$$D2_y = \frac{1}{\Delta y^2}\begin{bmatrix} \begin{matrix} -2I_d \\ \\ \end{matrix} & \begin{matrix} I_d \\ \\ \end{matrix} & \begin{matrix} 0 \\ \\ \end{matrix} \\ \begin{matrix} I_d \\ \\ \end{matrix} & \begin{matrix} -2I_d \\ \\ \end{matrix} & \begin{matrix} I_d \\ \\ \end{matrix} \\ \begin{matrix} 0 \\ \\ \end{matrix} & \begin{matrix} I_d \\ \\ \end{matrix} & \begin{matrix} -2I_d \\ \\ \end{matrix} \end{bmatrix}$$

Le système sera écrit sous la forme suivante sous Matlab :

$$\begin{bmatrix} A & 0 & G_x \\ 0 & A & G_y \\ D_x & D_y & 0 \end{bmatrix}\begin{bmatrix} U \\ V \\ P \end{bmatrix} = \begin{bmatrix} B_x + BCLx \\ B_y + BCLy \\ C \end{bmatrix}$$

Il se peut que la matrice $\begin{bmatrix} A & 0 & G_x \\ 0 & A & G_y \\ D_x & D_y & 0 \end{bmatrix}$ ne soit pas inversible. Pour régler ce problème, on pourra remplacer la matrice nulle en bas à droite par –ε.Id, avec un ε très petit pour pouvoir quand même inverser la matrice.

Avec $G_x = -Divx^T$ et $G_y = -Divy^T$. La matrice A définie, par la relation :

$$A = \frac{I_d}{\Delta t} - \frac{1}{Re}\left(D2_x + D2_y\right)$$

La matrice colonne Bx (même dimension de U) dans laquelle on définit la valeur de U Grad U à l'instant n est définie telle que :

$$B(i,j) = \frac{u^n(i,j)}{\Delta t} - u^n(i,j)\left(\frac{{u_{i+1,j}}^n - {u_{i-1,j}}^n}{2\Delta x}\right) - v^n(i,j)\left(\frac{{u_{i,j+1}}^n - {u_{i,j-1}}^n}{2\Delta y}\right)$$

$$B_x = \begin{bmatrix} b_x(1,1) \\ \vdots \\ b_x(I,1) \\ \vdots \\ b_x(1,J) \\ \vdots \\ b_x(I,J) \end{bmatrix}$$

En prenant en compte les conditions aux limites à la paroi (3), on a :

$$B_x = \begin{bmatrix} b_x(1,1) \\ \vdots \\ b_x(I,1) \\ \vdots \\ b_x(1,J) + \frac{1}{Re\Delta y^2}U_p \\ \vdots \\ b_x(I,J) + \frac{1}{Re\Delta y^2}U_p \end{bmatrix}$$

On procédera de même pour la matrice By.

On obtient vers la fin :

Avec :

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \quad ; \quad \tilde{B} = \begin{bmatrix} B_x \\ B_y \end{bmatrix} \quad ; \quad \widetilde{D} = \begin{bmatrix} D_x & D_y \end{bmatrix} \quad ; \quad \tilde{G} = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$$

$$\tilde{A}.\begin{bmatrix} U \\ V \end{bmatrix} + \tilde{G}.P = \tilde{B}$$

$$\widetilde{D}.\begin{bmatrix} U \\ V \end{bmatrix} = C$$

Du coup :

$$C = \tilde{D}.\tilde{A}.\tilde{B} - \tilde{D}.\tilde{A}^{-1}.\tilde{G}.P$$

Si

$$\tilde{D} = \tilde{G}^{t}$$

Alors on peut écrire comme quoi :

$$P = \left(\tilde{D}.\tilde{A}.\tilde{G}\right)^{-1}.(\tilde{D}.\tilde{A}^{-1}.\tilde{B} - C)$$

Nous pouvons au final obtenir nos paramètres de vitesses tel que :

$$\begin{bmatrix} U \\ V \end{bmatrix} = \tilde{A}^{-1}.(\tilde{B} - \tilde{G}.P)$$

5 Résolution avec Matlab

Nous avons tenté de simuler l'équation de Navier Stocks sur MATLAB grâce à la méthode des différences finies.

Le code utilisé est le suivant :

```
clear all
clc


L=input('Longueur: ');
H=input('Largeur: ');
T=input('Temps: ');
I=input('Valeur de I: ');
J=input('Valeur de J: ');
dt=input('Pas de temps: ');
Re=input('Reynolds: ');
Up=input('Vitesse paroie supérieur: ');


dx=L./(I+1);
dy=L./(J+1);


%% DD2x
v=ones(I-1,1);
Id=eye(I,I);
```

```
D2x=(-2.*Id+diag(v,1)+diag(v,-1));

DD2x=zeros(I*J,I*J);
for j=1:J
    Z=(j-1)*I+1:j*I;
    DD2x(Z,Z)=D2x;
end
DD2x=(1./dx.^2).*DD2x


%% DD2y

Idd=eye(J,J);
w=ones(I*J-J,1);
D2y=-2.*Idd;
DD2y=zeros(I*J,I*J);
for i=1:I
    ZZ=(i-1)*J+1:J*i;
    DD2y(ZZ,ZZ)=D2y;
end
DD2y=DD2y+diag(w,J)+diag(w,-J);
DD2y=(1./dy.^2).*DD2y


%%A

A=(I./dt)-(1./Re).*(DD2x+DD2y)


%%Gxx


Gx=zeros(I,I+1);
Gxx=zeros(I*J,(J+1)*(I+1));
for i=1:I
    Gx(i,i)=-1;
    Gx(i,i+1)=1;
end
Gx;

for j=1:J
    Z1=(j-1)*I+1:j*I;
    Z2=(j-1)*(I+1)+1:j*(I+1);
    Z3=j*(I+1)+1:(j+1)*(I+1);
    Gxx(Z1,Z2)=Gx;
    G2xx(Z1,Z3)=Gx;
end
Gxx=(1/(2*dx))*(Gxx+G2xx)
```

```
%%Gy


Gy=zeros(I,I+1);
Gyy=zeros(I*J,(J+1)*(I+1));
for i=1:I
    Gy(i,i)=-1;
    Gy(i,i+1)=1;
end
Gy;

for j=1:J
    Z1=(j-1)*I+1:j*I;
    Z2=(j-1)*(I+1)+1:j*(I+1);
    Z3=j*(I+1)+1:(j+1)*(I+1);
    Gyy(Z1,Z2)=Gx;
    G2yy(Z1,Z3)=Gx;
end
Gyy=(1/(2*dy))*(Gyy+G2yy)


%%Dx

for i=1:I
    Dx1(i,i)=1;
    Dx1(i+1,i)=-1;
end

for j=1:J
    Z1=(j-1)*(I+1);
    Z2=(j-1)*I;
    ZZ1=Z1+1:Z1+I+1;
    ZZ2=Z2+1:Z2+I;
    ZZ3=Z1+I+2:Z1+2*I+2;
    ZZ4=Z2+1:Z2+I;
    Dx(ZZ1,ZZ2)=Dx1;
    Dx(ZZ3,ZZ4)=Dx1;
end

Dx=Dx./(2*dx)

%Dy

for i=1:I
    Dy1(i,i)=1;
    Dy1(i+1,i)=1;
end
```

```
for j=1:J
    Z1=(j-1)*(I+1);
    Z2=(j-1)*I;
    ZZ1=Z1+1:Z1+I+1;
    ZZ2=Z2+1:Z2+I;
    ZZ3=Z1+I+2:Z1+2*I+2;
    ZZ4=Z2+1:Z2+I;
    Dy(ZZ1,ZZ2)=Dy1;
    Dy(ZZ3,ZZ4)=-Dy1;
end

Dy=Dy./(2*dy)


%Bxx

U=zeros(I,J);
V=zeros(I,J);
Bx=zeros(I,J);


for i=2:I-1
    j=1;
    Bx(i,j)=(U(i,j)./dt)-(U(i,j)*(U(i+1,j)-U(i-1,j)))./(2*dx)-
(U(i,j)*(U(i,j+1)))./(2*dy);
end

for i=2:I-1
    j=J;
    Bx(i,j)=(U(i,j)./dt)-(U(i,j)*(U(i+1,j)-U(i-1,j)))./(2*dx)-(U(i,j)*(Up-
U(i,j-1)))./(2*dy);
end

for j=2:J-1
    i=1;
    Bx(i,j)=(U(i,j)./dt)-(U(i,j)*(U(i+1,j)))./(2*dx)-(U(i,j)*(U(i,j+1)-
U(i,j-1)))./(2*dy);
end

for j=2:J-1
    i=I;
    Bx(i,j)=(U(i,j)./dt)-(U(i,j)*(-U(i-1,j)))./(2*dx)-(U(i,j)*(U(i,j+1)-
U(i,j-1)))./(2*dy);
end

for j=1;
    i=1;
```

```
    Bx(i,j)=(U(i,j)./dt)-(U(i,j)*(U(i+1,j))).*(2*dx)-
(V(i,j)*(U(i,j+1))).*(2*dy);
end

for j=J;
    i=I;
    Bx(i,j)=(U(i,j)./dt)-(U(i,j)*(-U(i-1,j))).*(2*dx)-(U(i,j)*(Up-U(i,j-
1))).*(2*dy);
end

for j=2:J-1
    for i=2:I-1
    Bx(i,j)=(U(i,j)./dt)-(U(i,j)*(U(i+1,j)-U(i-1,j))).*(2*dx)-
(U(i,j)*(U(i,j+1)-U(i,j-1))).*(2*dy);
    end
end

Bxx=zeros(I*J,1);

for j=1:J
    for i=1:I
        Bxx(i*i,1)=Bx(i,j)
    end
end
 Bx

%Byy

U=zeros(I,J);
V=zeros(I,J);
By=zeros(I,J);

for i=2:I-1
    j=1;
    By(i,j)=(V(i,j)./dt)-(V(i,j)*(V(i+1,j)-V(i-1,j))).*(2*dx)-
(V(i,j)*(V(i,j+1))).*(2*dy);
end

for i=2:I-1
    j=J;
    By(i,j)=(V(i,j)./dt)-(V(i,j)*(V(i+1,j)-V(i-1,j))).*(2*dx)-(V(i,j)*(-
V(i,j-1))).*(2*dy);
end

for j=2:J-1
    i=1;
    By(i,j)=(V(i,j)./dt)-(V(i,j)*(V(i+1,j))).*(2*dx)-(V(i,j)*(V(i,j+1)-
V(i,j-1))).*(2*dy);
end
```

```
for j=2:J-1
    i=I;
    By(i,j)=(V(i,j)./dt)-(V(i,j)*(-V(i-1,j))./(2*dx)-(V(i,j)*(V(i,j+1)-
V(i,j-1))./(2*dy);
end

for j=1;
    i=1;
    By(i,j)=(V(i,j)./dt)-(V(i,j)*(V(i+1,j)))./(2*dx)-
(V(i,j)*(V(i,j+1)))./(2*dy);
end

for j=J;
    i=I;
    By(i,j)=(V(i,j)./dt)-(V(i,j)*(-V(i-1,j))./(2*dx)-(V(i,j)*(-V(i,j-
1)))./(2*dy);
end

for j=2:J-1
    for i=2:I-1
    By(i,j)=(V(i,j)./dt)-(V(i,j)*(V(i+1,j)-V(i-1,j)))./(2*dx)-
(V(i,j)*(V(i,j+1)-V(i,j-1)))./(2*dy);
    end
end

Byy=zeros(I*J,1);

for j=1:J
    for i=1:I
        Byy(i*i,1)=By(i,j)
    end
end

Byy

%Matrice final

Matrice_final=[A,0*eye(I*J,I*J),Gxx;0*eye(I*J,I*J),A,Gyy;Dx,Dy,(-
0.0000001)*eye((J+1)*(I+1),(J+1)*(I+1))]


%C

 C=0;

%P

U=zeros((I+1)*(J+1),1);
```

```
P=(Dx.*inv(A).*Bxx+Dy.*inv(A).*Byy-C).*inv(Dx.*inv(A).*Gyy+Dy.*inv(A).*Gyy)

%U

U=zeros(I*J,1);

U=A^(-1).*(Bxx-Gxx.*P)

%V

V=zeros(I*J,1);

V=A^(-1).*(Byy-Gyy.*P)
```

Pour des valeurs initiales suivantes :

```
Longueur: 1
Largeur: 1
Temps: 1
Valeur de I: 3
Valeur de J: 3
Pas de temps: 3
Reynolds: 1
Vitesse paroie supérieur: 1
```

On obtient les matrices :

```
DD2x =

   -32    16     0     0     0     0     0     0     0
    16   -32    16     0     0     0     0     0     0
     0    16   -32     0     0     0     0     0     0
     0     0     0   -32    16     0     0     0     0
     0     0     0    16   -32    16     0     0     0
     0     0     0     0    16   -32     0     0     0
     0     0     0     0     0     0   -32    16     0
     0     0     0     0     0     0    16   -32    16
     0     0     0     0     0     0     0    16   -32

DD2y =

   -32     0     0    16     0     0     0     0     0
     0   -32     0     0    16     0     0     0     0
     0     0   -32     0     0    16     0     0     0
    16     0     0   -32     0     0    16     0     0
     0    16     0     0   -32     0     0    16     0
     0     0    16     0     0   -32     0     0    16
     0     0     0    16     0     0   -32     0     0
     0     0     0     0    16     0     0   -32     0
     0     0     0     0     0    16     0     0   -32
```

A =

65	-15	1	-15	1	1	1	1	1
-15	65	-15	1	-15	1	1	1	1
1	-15	65	1	1	-15	1	1	1
-15	1	1	65	-15	1	-15	1	1
1	-15	1	-15	65	-15	1	-15	1
1	1	-15	1	-15	65	1	1	-15
1	1	1	-15	1	1	65	-15	1
1	1	1	1	-15	1	-15	65	-15
1	1	1	1	1	-15	1	-15	65

Gxx =

-2	2	0	0	-2	2	0	0	0	0	0	0	0	0	0	0
0	-2	2	0	0	-2	2	0	0	0	0	0	0	0	0	0
0	0	-2	2	0	0	-2	2	0	0	0	0	0	0	0	0
0	0	0	0	-2	2	0	0	-2	2	0	0	0	0	0	0
0	0	0	0	0	-2	2	0	0	-2	2	0	0	0	0	0
0	0	0	0	0	0	-2	2	0	0	-2	2	0	0	0	0
0	0	0	0	0	0	0	0	-2	2	0	0	-2	2	0	0
0	0	0	0	0	0	0	0	0	-2	2	0	0	-2	2	0
0	0	0	0	0	0	0	0	0	0	-2	2	0	0	-2	2

Gyy =

-2	2	0	0	-2	2	0	0	0	0	0	0	0	0	0	0
0	-2	2	0	0	-2	2	0	0	0	0	0	0	0	0	0
0	0	-2	2	0	0	-2	2	0	0	0	0	0	0	0	0
0	0	0	0	-2	2	0	0	-2	2	0	0	0	0	0	0
0	0	0	0	0	-2	2	0	0	-2	2	0	0	0	0	0
0	0	0	0	0	0	-2	2	0	0	-2	2	0	0	0	0
0	0	0	0	0	0	0	0	-2	2	0	0	-2	2	0	0
0	0	0	0	0	0	0	0	0	-2	2	0	0	-2	2	0
0	0	0	0	0	0	0	0	0	0	-2	2	0	0	-2	2

Dx =

2	0	0	0	0	0	0	0	0
-2	2	0	0	0	0	0	0	0
0	-2	2	0	0	0	0	0	0
0	0	-2	0	0	0	0	0	0
2	0	0	2	0	0	0	0	0
-2	2	0	-2	2	0	0	0	0
0	-2	2	0	-2	2	0	0	0
0	0	-2	0	0	-2	0	0	0
0	0	0	2	0	0	2	0	0
0	0	0	-2	2	0	-2	2	0
0	0	0	0	-2	2	0	-2	2
0	0	0	0	0	-2	0	0	-2
0	0	0	0	0	0	2	0	0
0	0	0	0	0	0	-2	2	0
0	0	0	0	0	0	0	-2	2
0	0	0	0	0	0	0	0	-2

Dy =

2	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0
0	2	2	0	0	0	0	0	0
0	0	2	0	0	0	0	0	0
-2	0	0	2	0	0	0	0	0
-2	-2	0	2	2	0	0	0	0
0	-2	-2	0	2	2	0	0	0
0	0	-2	0	0	2	0	0	0
0	0	0	-2	0	0	2	0	0
0	0	0	-2	-2	0	2	2	0
0	0	0	0	-2	-2	0	2	2
0	0	0	0	0	-2	0	0	2
0	0	0	0	0	0	-2	0	0
0	0	0	0	0	0	-2	-2	0
0	0	0	0	0	0	0	-2	-2
0	0	0	0	0	0	0	0	-2

```
Bxx =

     0
     0
     0
     0
     0
     0
     0
     0
     0
```

```
Byy =

     0
     0
     0
     0
     0
     0
     0
     0
     0
```

```
Matrice_final =

  Columns 1 through 23

   65.0000  -15.0000    1.0000  -15.0000    1.0000    1.0000    1.0000    1.0000    1.0000         0         0         0         0         0         0         0         0         0   -2.0000    2.0000         0         0   -2.0000
  -15.0000   65.0000  -15.0000    1.0000  -15.0000    1.0000    1.0000    1.0000    1.0000         0         0         0         0         0         0         0         0         0         0   -2.0000    2.0000         0         0
    1.0000  -15.0000   65.0000    1.0000    1.0000  -15.0000    1.0000    1.0000    1.0000         0         0         0         0         0         0         0         0         0         0         0   -2.0000    2.0000         0
  -15.0000    1.0000    1.0000   65.0000  -15.0000    1.0000  -15.0000    1.0000    1.0000         0         0         0         0         0         0         0         0         0         0         0         0         0   -2.0000
    1.0000  -15.0000    1.0000  -15.0000   65.0000  -15.0000    1.0000  -15.0000    1.0000         0         0         0         0         0         0         0         0         0         0         0         0         0         0
    1.0000    1.0000  -15.0000    1.0000  -15.0000   65.0000    1.0000    1.0000  -15.0000         0         0         0         0         0         0         0         0         0         0         0         0         0         0
    1.0000    1.0000    1.0000  -15.0000    1.0000    1.0000   65.0000  -15.0000    1.0000         0         0         0         0         0         0         0         0         0         0         0         0         0         0
    1.0000    1.0000    1.0000    1.0000  -15.0000    1.0000  -15.0000   65.0000  -15.0000         0         0         0         0         0         0         0         0         0         0         0         0         0         0
    1.0000    1.0000    1.0000    1.0000    1.0000  -15.0000    1.0000  -15.0000   65.0000         0         0         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0   65.0000  -15.0000    1.0000  -15.0000    1.0000    1.0000    1.0000    1.0000    1.0000   -2.0000    2.0000         0         0   -2.0000
         0         0         0         0         0         0         0         0         0  -15.0000   65.0000  -15.0000    1.0000  -15.0000    1.0000    1.0000    1.0000    1.0000         0   -2.0000    2.0000         0         0
         0         0         0         0         0         0         0         0         0    1.0000  -15.0000   65.0000    1.0000    1.0000  -15.0000    1.0000    1.0000    1.0000         0         0   -2.0000    2.0000         0
         0         0         0         0         0         0         0         0         0  -15.0000    1.0000    1.0000   65.0000  -15.0000    1.0000  -15.0000    1.0000    1.0000         0         0         0         0   -2.0000
         0         0         0         0         0         0         0         0         0    1.0000  -15.0000    1.0000  -15.0000   65.0000  -15.0000    1.0000  -15.0000    1.0000         0         0         0         0         0
         0         0         0         0         0         0         0         0         0    1.0000    1.0000  -15.0000    1.0000  -15.0000   65.0000  -15.0000    1.0000  -15.0000         0         0         0         0         0
         0         0         0         0         0         0         0         0         0    1.0000    1.0000    1.0000  -15.0000    1.0000    1.0000   65.0000  -15.0000    1.0000         0         0         0         0         0
         0         0         0         0         0         0         0         0         0    1.0000    1.0000    1.0000    1.0000  -15.0000    1.0000  -15.0000   65.0000  -15.0000         0         0         0         0         0
         0         0         0         0         0         0         0         0         0    1.0000    1.0000    1.0000    1.0000    1.0000  -15.0000    1.0000  -15.0000   65.0000         0         0         0         0         0
    2.0000         0         0         0         0         0         0         0         0    2.0000         0         0         0         0         0         0         0         0   -0.0000         0         0         0         0
   -2.0000    2.0000         0         0         0         0         0         0         0    2.0000    2.0000         0         0         0         0         0         0         0         0   -0.0000         0         0         0
         0   -2.0000    2.0000         0         0         0         0         0         0         0    2.0000    2.0000         0         0         0         0         0         0         0         0   -0.0000         0         0
         0         0   -2.0000         0         0         0         0         0         0         0         0    2.0000         0         0         0         0         0         0         0         0         0   -0.0000         0
    2.0000         0         0    2.0000         0         0         0         0         0   -2.0000         0         0    2.0000         0         0         0         0         0         0         0         0         0   -0.0000
   -2.0000    2.0000         0   -2.0000    2.0000         0         0         0         0   -2.0000   -2.0000         0    2.0000    2.0000         0         0         0         0         0         0         0         0         0
         0   -2.0000    2.0000         0   -2.0000    2.0000         0         0         0         0   -2.0000   -2.0000         0    2.0000    2.0000         0         0         0         0         0         0         0         0
         0         0   -2.0000         0         0   -2.0000         0         0         0         0         0         0         0    2.0000         0         0         0         0         0         0         0         0         0
         0         0         0    2.0000         0         0    2.0000         0         0         0         0         0   -2.0000         0    2.0000         0         0         0         0         0         0         0         0
         0         0         0   -2.0000    2.0000         0   -2.0000    2.0000         0         0         0         0   -2.0000   -2.0000         0    2.0000    2.0000         0         0         0         0         0         0
         0         0         0         0   -2.0000    2.0000         0   -2.0000    2.0000         0         0         0         0   -2.0000   -2.0000         0    2.0000    2.0000         0         0         0         0         0
         0         0         0         0         0   -2.0000         0         0   -2.0000         0         0         0         0         0   -2.0000         0         0    2.0000         0         0         0         0         0
         0         0         0         0         0         0    2.0000         0         0         0         0         0         0         0         0   -2.0000         0         0         0         0         0         0         0
         0         0         0         0         0         0   -2.0000    2.0000         0         0         0         0         0         0         0   -2.0000   -2.0000         0         0         0         0         0         0
         0         0         0         0         0         0         0   -2.0000    2.0000         0         0         0         0         0         0         0   -2.0000   -2.0000   -2.0000         0         0         0         0
         0         0         0         0         0         0         0         0   -2.0000         0         0         0         0         0         0         0         0   -2.0000         0         0         0         0         0
```

Columns 24 through 34

2.0000	0	0	0	0	0	0	0	0	0	0
-2.0000	2.0000	0	0	0	0	0	0	0	0	0
0	-2.0000	2.0000	0	0	0	0	0	0	0	0
2.0000	0	0	-2.0000	2.0000	0	0	0	0	0	0
-2.0000	2.0000	0	0	-2.0000	2.0000	0	0	0	0	0
0	-2.0000	2.0000	0	0	-2.0000	2.0000	0	0	0	0
0	0	0	-2.0000	2.0000	0	0	-2.0000	2.0000	0	0
0	0	0	0	-2.0000	2.0000	0	0	-2.0000	2.0000	0
0	0	0	0	0	-2.0000	2.0000	0	0	-2.0000	2.0000
2.0000	0	0	0	0	0	0	0	0	0	0
-2.0000	2.0000	0	0	0	0	0	0	0	0	0
0	-2.0000	2.0000	0	0	0	0	0	0	0	0
2.0000	0	0	-2.0000	2.0000	0	0	0	0	0	0
-2.0000	2.0000	0	0	-2.0000	2.0000	0	0	0	0	0
0	-2.0000	2.0000	0	0	-2.0000	2.0000	0	0	0	0
0	0	0	-2.0000	2.0000	0	0	-2.0000	2.0000	0	0
0	0	0	0	-2.0000	2.0000	0	0	-2.0000	2.0000	0
0	0	0	0	0	-2.0000	2.0000	0	0	-2.0000	2.0000
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
-0.0000	0	0	0	0	0	0	0	0	0	0
0	-0.0000	0	0	0	0	0	0	0	0	0
0	0	-0.0000	0	0	0	0	0	0	0	0
0	0	0	-0.0000	0	0	0	0	0	0	0
0	0	0	0	-0.0000	0	0	0	0	0	0
0	0	0	0	0	-0.0000	0	0	0	0	0
0	0	0	0	0	0	-0.0000	0	0	0	0
0	0	0	0	0	0	0	-0.0000	0	0	0
0	0	0	0	0	0	0	0	-0.0000	0	0
0	0	0	0	0	0	0	0	0	-0.0000	0
0	0	0	0	0	0	0	0	0	0	-0.0000

Conclusion :

Même si nous ne sommes pas arrivés au bout du rapport, nous avons clairement pour calculer les vitesses, nous avons clairement pu voir lors de notre avancé sur le code et lors des séances de cours, comme quoi la méthode des différences finies est bien adaptée pour résoudre les équations de Naviers-stocks (simplification prises en compte).
La compréhension de cette méthode nous aidera à développer un esprit critique envers les résultats donnés par des logiciels de calculs tel que Fluent, et que si les résultats donné par celui-ci sont illogiques, on serait mieux adapté à cibler la source du problème, vu que l'on a conscience de ce qui se passe dans la partie numérique.