220311/205001 – Computational engineering

Assignment 2 Generic convection-diffusion transport equation

Boulogne Quentin

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

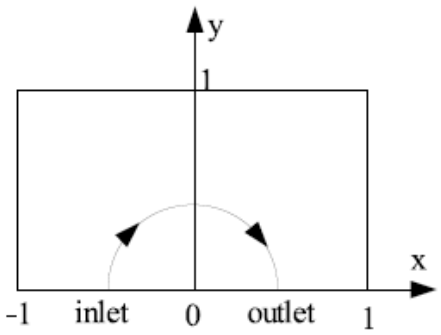
SOMMAIRE

1 INTRODUCTION

The Smith Hutton is a convection diffusion problem, so we will see how to solve this kinds of problems. We will use multiple numerical schemes to solve this problem and use different meshes to see the influence on our results compared to the results given in the exercise pdf.

2 SMITH HUTTON PROBLEM

We can see a scheme of the Smith Hutton problem :



2.1 Input data

The different data are :

- L=[-1 , 1] m ; H=1 m
- α = 10
  - ➢ Velocity components:
- $u = 2y(1 - x^2)$
- $v = -2x(1 - y^2)$
  - ➢ Bondary conditions :
- Inlet : $\phi = 1 + \tanh(\alpha(2x + 1))\ for\ x\ \epsilon\ [-1; 0]$
- Outlet : $\frac{\partial \phi}{\partial y}(x, 0) = 0\ \ for\ x\ \epsilon\ [0; 1]$
- Left wall : $\phi(-1, y) = 1 + \tanh(\alpha)$
- Right wall : $\phi(1, y) = 1 + \tanh(\alpha)$
- Upper wall : $\phi(x, 1) = 1 + \tanh(\alpha)$

2.2 Methodology

Solving the Smith Hutton problem is solving the convection diffusion equation that means solving :

$$\frac{\partial \rho\phi}{\partial t} + \nabla \cdot (\rho\vec{v}\phi) = \nabla \cdot (\Gamma \nabla\phi) + S$$

The discretization for the implicit scheme :

$$\frac{(\rho\phi)_P^{n+1} - (\rho\phi)_P^n}{\Delta t} \cdot \Delta x\Delta y + [(\rho u\phi)_e^{n+1} - (\rho u\phi)_w^{n+1}] \cdot \Delta y + [(\rho v\phi)_n^{n+1} - (\rho v\phi)_s^{n+1}] \cdot \Delta x$$
$$= \left[\left(\Gamma\frac{\partial\phi}{\partial x}\right)_e^{n+1} - \left(\Gamma\frac{\partial\phi}{\partial x}\right)_w^{n+1}\right] \cdot \Delta y + \left[\left(\Gamma\frac{\partial\phi}{\partial x}\right)_n^{n+1} - \left(\Gamma\frac{\partial\phi}{\partial x}\right)_s^{n+1}\right] \cdot \Delta x + S_P^{n+1} \cdot \Delta x\Delta y$$

After discretization we obtain the

$$a_P \cdot \phi_P^{n+1} = a_N \cdot \phi_N^{n+1} + a_S \cdot \phi_S^{n+1} + a_E \cdot \phi_E^{n+1} + a_W \cdot \phi_W^{n+1} + b + b_{pe}$$

$a_E = D_e.A(|Pe_e|) + \max(-F_e, 0)$ $\quad D_e = \frac{\Gamma_e \Delta y}{(\delta x)_e}$ $\quad F_e = (\varphi u)_e \Delta y$

$a_W = D_w.A(|Pe_w|) + \max(F_w, 0)$ $\quad D_w = \frac{\Gamma_w \Delta y}{(\delta x)_w}$ $\quad F_w = (\varphi u)_w \Delta y$

$a_N = D_n.A(|Pe_n|) + \max(-F_n, 0)$ $\quad D_n = \frac{\Gamma_n \Delta x}{(\delta y)_n}$ $\quad F_n = (\varphi v)_n \Delta x$

$a_S = D_s.A(|Pe_s|) + \max(F_s, 0)$ $\quad D_s = \frac{\Gamma_s \Delta x}{(\delta y)_s}$ $\quad F_s = (\varphi v)_s \Delta x$

$$a_P = a_E + a_W + a_N + a_S + \varphi_P^n \frac{\Delta x\Delta y}{\Delta t}$$

$$b = \varphi_P^n \frac{\Delta x\Delta y}{\Delta t}\phi_P^n + S_P^{n+1}\Delta x\Delta y$$

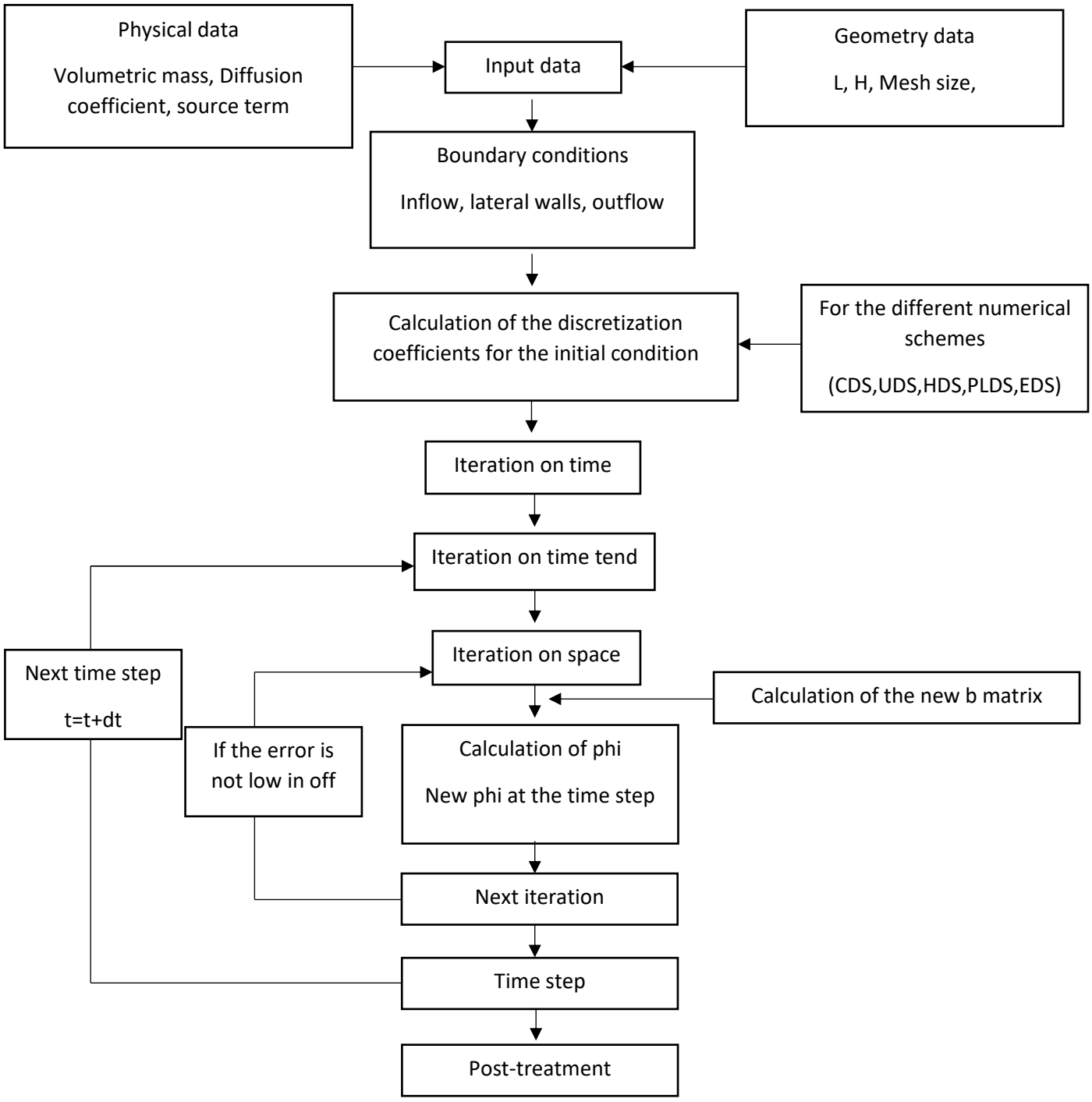
Peclet number at the face (f) : $P_f = \frac{F_f}{D_f}$

Numerical schemes	A(|P|)
UDS	1
CDS	1-0.5(|P|)
HDS	max(0,(1-0.5(|P|)))
EDS	|P|/(exp(|P|)-1)
PLDS	max(0,(1-0.5(|P|))^5)

Those schemes are all first order numerical schemes. To see if an high order scheme is better than the first order we implement a second order numerical scheme.

In our case we suppose a incompressible flow so we can calculate every coefficients before the calculation exept for b witch is changing with phi .

3 CODE STRUCTURE



4 CODE VERIFICATION

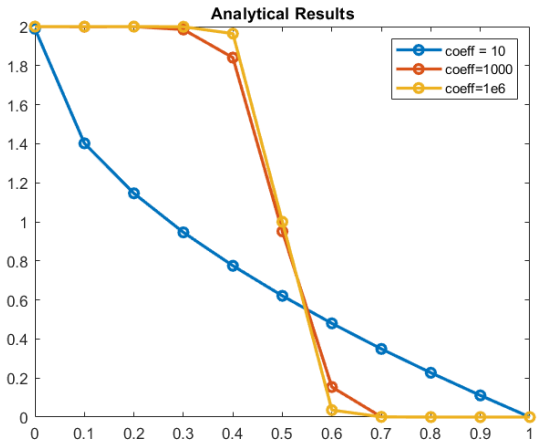
As we can see in the exercise pdf we have some results given:

Position x	$\rho/\Gamma = 10$	$\rho/\Gamma = 1000$	$\rho/\Gamma = 1000000$
0.0	1.989	2.0000	2.000
0.1	1.402	1.9990	2.000
0.2	1.146	1.9997	2.000
0.3	0.946	1.9850	1.999
0.4	0.775	1.8410	1.964
0.5	0.621	0.9510	1.000
0.6	0.480	0 1540	0.036
0.7	0.349	0.0010	0.001
0.8	0.227	0.0000	0.000
0.9	0.111	0.0000	0.000
1.0	0.000	0.0000	0.000

Table 1: Numerical results at the outlet for different ρ/Γ numbers

In order to verify my result I compares those results to mine with an maximum error.

We will see that discontinuities are increasing the error even if the mesh is better the discontinuities are showing higher errors. To see these issue I deed a plot of the numerical results at the outlet for the different values of rho on the diffusive coefficient.



Those numerical solution can be compared to my results. It's what I did in the results part.

5 RESULTS DISCUSSION

5.1 Improving the mesh

We can see that the mesh is playing a important part on the accuracy of the results, this is why we need to take a mesh with in off control volumes to have an accurate result. To tes the influence in a first part I took a mesh of 19 by 19 control volumes.

Finally, I took a mesh of 199 by 199 control volumes witch gives good results.

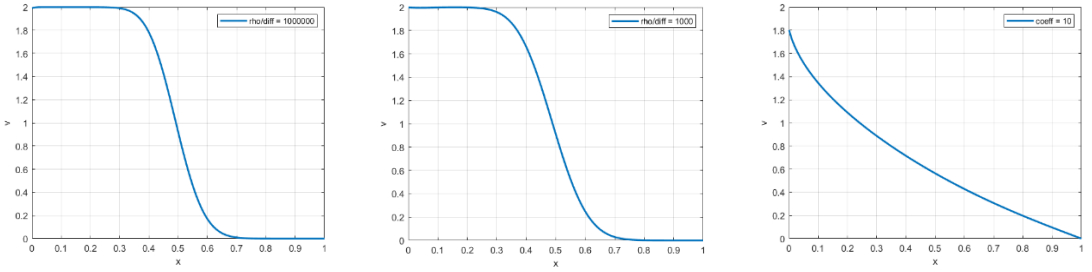
5.2 Changing numerical schemes

For every scheme we are using a mesh of 199 by 199 control volumes.

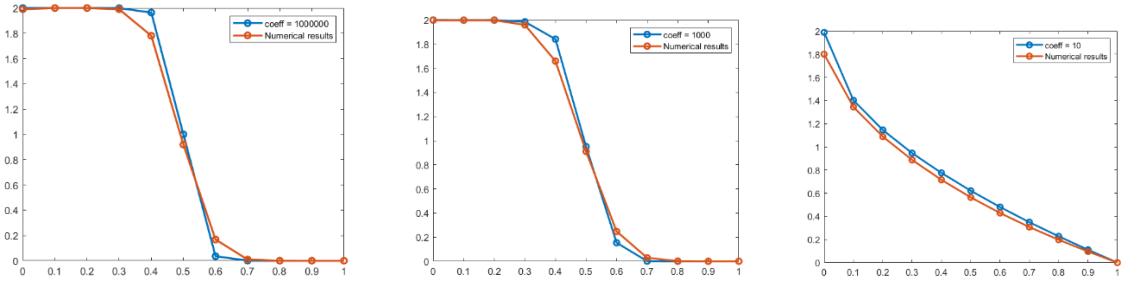
We can compare the different first order numerical schemes CDS, UDS , HDS , EDS , PLDS. In the results we are presenting the UDS and HDS scheme but all those schemes are available in my code.

5.3 Upwind Differencing Scheme

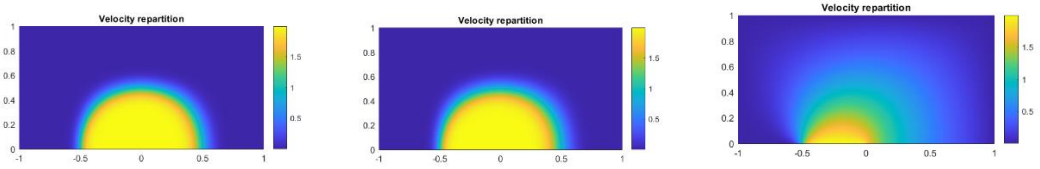
We are looking at the results obtained by the UDS of first order .



For my code with a fine mesh depending on rho/diff



Analytical solution compared to numerical solution UDS



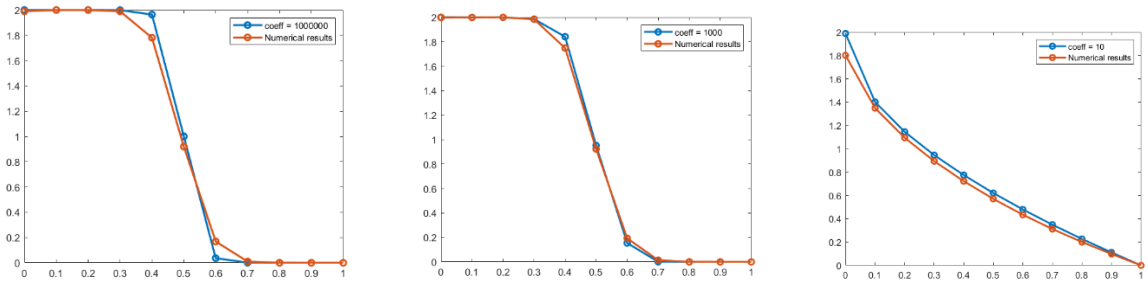
Velocity repartition UDS

$\frac{\varphi}{\Gamma} = 1.0e6$	$\frac{\varphi}{\Gamma} = 1000$	$\frac{\varphi}{\Gamma} = 10$
2.3160 %	2.2175 %	7.4236 %

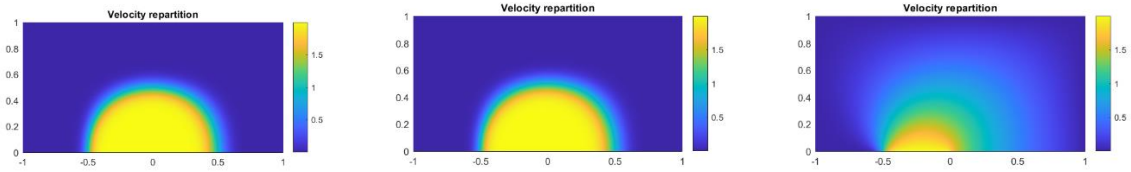
Error values

We have a mesh of 199 elements by 199 elements the choice is to have the results calculated at the points where we have an analytical solution. As we can see for $\frac{\varphi}{\Gamma} = 10$ the error on the solution is higher than in over cases this is due to discontinuities at x=0.

5.1 High Differencing Scheme



Analytical solution compared to numerical solution HDS



Velocity repartition HDS

$\frac{\varphi}{\Gamma} = 1.0e6$	$\frac{\varphi}{\Gamma} = 1000$	$\frac{\varphi}{\Gamma} = 10$
2.3157 %	1.0444 %	6.9966 %

Error values

The maximum error is also for $\frac{\varphi}{\Gamma} = 10$ this is due to the discontinuity at x=0.

5.2 Conclusion on results

Results are showing that in this case the UDS and HDS scheme are giving pretty much the same results. We can see the issues of the discontinuity for $\frac{\varphi}{\Gamma} = 10$ at x=0. To see the difference with a second order scheme would be good to see the actual benefit of the high orders schemes.

6 CONCLUSION

With more time the goal would be to implement the high order schemes (second order or higher) to study the influence of an high order scheme. The other thing to take into account is that my code is able to calculate the different solution with different first order schemes HDS, CDS,UDS,EDS,PLDS. The different schemes are to use with caution because the value of the Peclet number can lead to a diverging solution.

7 ANNEXE

7.1 Code for Smith Hutton resolution

```
%Ex2 Smith-hutton problem
%Auteur Boulogne Quentin

clear all;
close all ;
clc;
%% INPUT DATA
clc;
clear;
%Domain lengths
Lx=2.0;% L length along positive x axis [m]
Ly=1.0;% H hight along positive y axis [m]
domainLengths=[Lx Ly];% domain size [m]

%Mesh sizes
Nx=199; %Number of counter volumes on x axis
Ny=199;%Number of counter volumes on y axis
meshSizes=[Nx Ny];% Number of counter volumes on each axis

scheme=1; %UDS = 1 CDS = 2 HDS = 3 EDS=4 PLDS=5

dx=Lx/(Nx+1);
dy=Ly/(Ny+1);

Npx=Nx+2;
Npy=Ny+2;
```

```
% Mesh definition
x=linspace(-1,1,Npx);
y=linspace(Ly,0,Npy);
[X,Y] = meshgrid(x,y);

an=zeros(Npy,Npx);
aw=zeros(Npy,Npx);
ae=zeros(Npy,Npx);
as=zeros(Npy,Npx);
ap=ones(Npy,Npx);
bp=zeros(Npy,Npx);

F_n=zeros(Npy,Npx);
F_s=zeros(Npy,Npx);
F_w=zeros(Npy,Npx);
F_e=zeros(Npy,Npx);

P_n=zeros(Npy,Npx);
P_s=zeros(Npy,Npx);
P_e=zeros(Npy,Npx);
P_w=zeros(Npy,Npx);

phi=zeros(Npy,Npx);
phinplus=zeros(Npy,Npx);
dt=0.001;
alpha=10;
S=0; % Source term
Coeff=1000000; % rho / diff
rho=1;%  density [kg/m^3]
Diff_coeff=rho/Coeff;


Sx=dx;
Sy=dy;
Se=Sy;
Sw=Sy;
Ss=Sx;
Sn=Sx;


Vx=zeros(Npy,Npx);
Vy=zeros(Npy,Npx);

max_divergence=1.0e10;
max_time=10;
max_iteration=1000000;
```

```
tolerance_solver=1.0e-5;
error = 1;


   for i=1:Npy
      for j=1:Npx

         De(i,j)=Diff_coeff*dy/dx;
         Dw(i,j)=Diff_coeff*dy/dx;
         Ds(i,j)=Diff_coeff*dx/dy;
         Dn(i,j)=Diff_coeff*dx/dy;

         Vx(i,j)=2*Y(i,j)*(1-(X(i,j))^2);
         Vy(i,j)=-2*X(i,j)*(1-(Y(i,j))^2);

         F_n(i,j)=rho*Vy(i,j)*Sn;
         F_s(i,j)=rho*Vy(i,j)*Ss;
         F_e(i,j)=rho*Vx(i,j)*Se;
         F_w(i,j)=rho*Vx(i,j)*Sw;

        P_n(i,j)=F_n(i,j)/Dn(i,j);
        P_s(i,j)=F_s(i,j)/Ds(i,j);
        P_e(i,j)=F_e(i,j)/De(i,j);
        P_w(i,j)=F_w(i,j)/Dw(i,j);

        if i > 1 && i < Npy && j > 1 && j < Npx && scheme==1 %UDS
           ae(i,j)=De(i,j)*1+max(-F_e(i,j),0);
           aw(i,j)=Dw(i,j)*1+max(F_w(i,j),0);
           an(i,j)=Dn(i,j)*1+max(-F_n(i,j),0);
           as(i,j)=Ds(i,j)*1+max(F_s(i,j),0);
           ap(i,j)=ae(i,j)+aw(i,j)+an(i,j)+as(i,j)+rho*dx*dy/dt;
        end

        if i > 1 && i < Npy && j > 1 && j < Npx && scheme==2 %CDS
         ae(i,j)=De(i,j)*(1-0.5*abs(P_e(i,j)))+max(-F_e(i,j),0);
         aw(i,j)=Dw(i,j)*(1-0.5*abs(P_w(i,j)))+max(F_w(i,j),0);
         an(i,j)=Dn(i,j)*(1-0.5*abs(P_n(i,j)))+max(-F_n(i,j),0);
         as(i,j)=Ds(i,j)*(1-0.5*abs(P_s(i,j)))+max(F_s(i,j),0);
         ap(i,j)=ae(i,j)+aw(i,j)+an(i,j)+as(i,j)+rho*dx*dy/dt;
        end


         if i > 1 && i < Npy && j > 1 && j < Npx && scheme==3 %HDS
         ae(i,j)=De(i,j)*max(0,(1-0.5*abs(P_e(i,j))))+max(-F_e(i,j),0);
         aw(i,j)=Dw(i,j)*max(0,(1-0.5*abs(P_w(i,j))))+max(F_w(i,j),0);
         an(i,j)=Dn(i,j)*max(0,(1-0.5*abs(P_n(i,j))))+max(-F_n(i,j),0);
```

```
        as(i,j)=Ds(i,j)*max(0,(1-0.5*abs(P_s(i,j))))+max(F_s(i,j),0);
        ap(i,j)=ae(i,j)+aw(i,j)+an(i,j)+as(i,j)+rho*dx*dy/dt;
        end

        if i > 1 && i < Npy && j > 1 && j < Npx && scheme==4 %EDS
        ae(i,j)=De(i,j)*(abs(P_e(i,j))/(exp(abs(P_e(i,j)))-1))+max(-F_e(i,j),0);
        aw(i,j)=Dw(i,j)*(abs(P_w(i,j))/(exp(abs(P_w(i,j)))-1))+max(F_w(i,j),0);
        an(i,j)=Dn(i,j)*(abs(P_n(i,j))/(exp(abs(P_n(i,j)))-1))+max(-F_n(i,j),0);
        as(i,j)=Ds(i,j)*(abs(P_s(i,j))/(exp(abs(P_s(i,j)))-1))+max(F_s(i,j),0);
        ap(i,j)=ae(i,j)+aw(i,j)+an(i,j)+as(i,j)+rho*dx*dy/dt;
        end
        if i > 1 && i < Npy && j > 1 && j < Npx && scheme==5 %PLDS
        ae(i,j)=De(i,j)*max(0,(1-0.5*abs(P_e(i,j)))^5)+max(-F_e(i,j),0);
        aw(i,j)=Dw(i,j)*max(0,(1-0.5*abs(P_w(i,j)))^5)+max(F_w(i,j),0);
        an(i,j)=Dn(i,j)*max(0,(1-0.5*abs(P_n(i,j)))^5)+max(-F_n(i,j),0);
        as(i,j)=Ds(i,j)*max(0,(1-0.5*abs(P_s(i,j)))^5)+max(F_s(i,j),0);
        ap(i,j)=ae(i,j)+aw(i,j)+an(i,j)+as(i,j)+rho*dx*dy/dt;
        end
       %max( P_n, P_s, P_e, P_w);
        phi_old(i,j)=1-tanh(alpha);
       % Inlet
       if i==Npy && j<Npx/2+2
           phi_old(i,j)=1+tanh(alpha*(2*X(i,j)+1));
       end
       %outlet
       if i==Npy && j>Npx/2
          phi_old(i,j)=phi_old(i-1,j);
       end

     end
  end
phi=phi_old;
Nt=100000;
ite=0;
tend=0.01;
t=0;
while t<tend

        for i=1:Npy
          for j=1:Npx
             bp(i,j)= S*dx*dy+rho*dx*dy*phi(i,j)/dt;
          end
        end

while error>tolerance_solver && ite<max_iteration
```

```
        for i=1:Npy
            for j=1:Npx
                bp(i,j)= S*dx*dy+rho*dx*dy*phi(i,j)/dt;
              if i==Npy && j<Npx/2+1
                 phinplus(i,j)=1+tanh(alpha*(2*X(i,j)+1));
              end
              if i==1
                 phinplus(i,j)=1-tanh(alpha);
              end
              if j==1 || j==Npx
                 phinplus(i,j)=1-tanh(alpha);
              end
              if i > 1 && i < Npy && j > 1 && j < Npx
                  phinplus(i,j)=(1/ap(i,j))*(ae(i,j)*phi(i,j+1)+aw(i,j)*phi(i,j-1)+an(i,j)*phi(i-
1,j)+as(i,j)*phi(i+1,j)+bp(i,j));
              end
               if i==Npy && j> Npx/2
                 phinplus(i,j)=phinplus(i-1,j);
              end
            end
        end
    error=max(max(abs(phinplus-phi)));
    phi=phinplus;
    ite=ite+1;
 end
 t=t+dt;
 phi=phinplus;
 end

Stream=zeros(Npy,Npx);
for i=1:Npy
   for j=1:Npx
      Stream(i,j)=-(1-X(i,j)^2)*(1-Y(i,j)^2);
   end
end
figure('Name','Streamfunction','NumberTitle','off');
contour(X,Y,Stream,10,'LineWidth',2);
title("Stream function");
Results=zeros(1,11);

Results(1)=phinplus(Npy,101);
Results(2)=phinplus(Npy,111);
Results(3)=phinplus(Npy,121);
Results(4)=phinplus(Npy,131);
Results(5)=phinplus(Npy,141);
Results(6)=phinplus(Npy,151);
```

```
Results(7)=phinplus(Npy,161);
Results(8)=phinplus(Npy,171);
Results(9)=phinplus(Npy,181);
Results(10)=phinplus(Npy,191);
Results(11)=phinplus(Npy,201);
Results;

% Analytical solution
x_ana=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0];
Coeff_10=[1.989 1.402 1.146 0.946 0.775 0.621 0.480 0.349 0.227 0.111 0.0];
Coeff_1000=[2.0 1.9990 1.9997 1.9850 1.8410 0.9510 0.1540 0.0010 0.0 0.0 0.0];
Coeff_1000000=[2.0 2.0 2.0 1.999 1.964 1.000 0.036 0.001 0.0 0.0 0.0];
Coeff_anal=[Coeff_10 Coeff_1000 Coeff_1000000];

% plot(x_ana,phinplus(Npy,1:11));
figure('Name','Analatical results results ','NumberTitle','off');
plot(x_ana,Coeff_10,'-o',x_ana,Coeff_1000,'-o',x_ana,Coeff_1000000,'-
o','LineWidth',2)
legend('coeff = 10','coeff=1000','coeff=1e6')
title('Analytical Results')

figure('Name','Result 2D vizualisation','NumberTitle','off');
pcolor(X,Y,phinplus);
colorbar
shading interp;
axis equal
title('Velocity repartition')
axis tight
grid on

totalresult=phinplus(Npy,101:end);

if Coeff==10
    error_result=max(abs(Results-Coeff_10)*100/Coeff_10);
    figure('Name','Results Comparaison ','NumberTitle','off');
    plot(x_ana,Coeff_10,'-o',x_ana,Results,'-o','LineWidth',2)
    legend('coeff = 10','Numerical results')
    figure('Name','Results for the mesh used ','NumberTitle','off');
    plot(X(Npy,101:end),totalresult,'LineWidth',2)
    legend('coeff = 10')
    xlabel('x');
    ylabel('v');
    grid
end
if Coeff==1000
    error_result=max(abs(Results-Coeff_1000)*100/Coeff_1000);
```

```
    figure('Name','Results Comparaison ','NumberTitle','off');
    plot(x_ana,Coeff_1000,'-o',x_ana,Results,'-o','LineWidth',2)
    legend('coeff = 1000','Numerical results')
    figure('Name','Results for the mesh used ','NumberTitle','off');
    plot(X(Npy,101:end),totalresult,'LineWidth',2)
    legend('rho/diff = 1000')
    xlabel('x');
    ylabel('v');
    grid
end
if Coeff==1000000
    error_result=max(abs(Results-Coeff_1000000)*100/Coeff_1000000);
    figure('Name','Results Comparaison ','NumberTitle','off');
    plot(x_ana,Coeff_1000000,'-o',x_ana,Results,'-o','LineWidth',2)
    legend('coeff = 1000000','Numerical results')
     figure('Name','Results for the mesh used ','NumberTitle','off');
    plot(X(Npy,101:end),totalresult,'LineWidth',2)
    xlabel('x');
    ylabel('v');
    legend('rho/diff = 1000000')

    grid
end

Peclet=max(max(abs(P_n)));
```