220311/205001 – Computational engineering

Assignment 1 Non – viscous potential flows

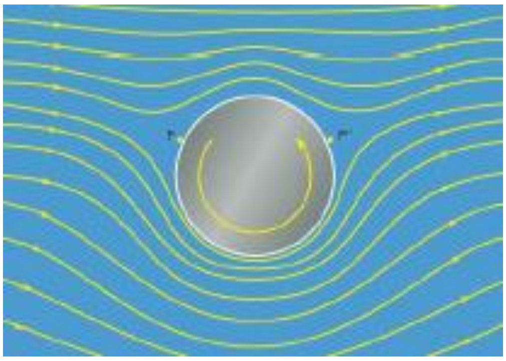
Boulogne Quentin

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

SOMMAIRE

1 INTRODUCTION

2 PROBLEM DEFINITION



2.1 Input data

The different data are :

- L=1 m ; H=0.5 m
- α = 10
  - Inflow conditions:
- $v_{in} = 20\ m/s$
- $P_{in} = 1e5\ Pa$
- $T_{in} = 300\ K$
- Initial volumic mass according to perfect gaz law
  - Bondary conditions :
- Inflow : Dirichlet condition $\psi(0, y) = v_{in}. y\ \ for\ y\ \epsilon\ [0; 0.5]$
- lateral : Dirichlet condition $\psi(x, 0) = v_{in}. 0\ and\ \ \psi(x, 0.5) = v_{in}. 0.5$
- Outflow : $Neuman\ condition$ $\frac{\partial\,\psi}{\partial y}(1, y) = 0\ \ for\ y\ \epsilon\ [0; 0.5]$ means that the $\psi(1 - dx, y) = \psi(1, y)$

2.2 Methodology

To solve this problem we make the assumption on the mass conservation equation witch is :

$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0$ for a steady, incompressible, 2 D and irrotational flow ~~flow~~ we have

$\rho \nabla \cdot (\vec{v}) = 0$.

The stream function in steady 2D flows are defined by :

$$v_x = \frac{\varphi_{ref}}{\varphi}\left(\frac{\partial \psi}{\partial y}\right) \ ; \ v_y = -\frac{\varphi_{ref}}{\varphi}\left(\frac{\partial \psi}{\partial x}\right)$$

Then by replacing v_x and v_y in $\nabla \cdot (\vec{v}) = 0$ we obtain the equation of the problem according to the assumption we made:

$$\frac{\partial}{\partial x}\left(\frac{\varphi_{ref}}{\varphi}\frac{\partial \psi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\varphi_{ref}}{\varphi}\frac{\partial \psi}{\partial y}\right) = 0$$

Stoke's theorem is egal to the circulation 0 in irrotational flow :

$$\int_S (\nabla \times \vec{v}) dS = \oint_C \vec{v} \cdot d\vec{\ell}$$

$$\Gamma \approx v_{ye}\Delta y_P - v_{xn}\Delta x_P - v_{yw}\Delta y_P + v_{xs}\Delta x_P = 0$$

$$-\frac{\varphi_{ref}}{\varphi_e}\left(\frac{\partial \psi}{\partial x}\right)_e \cdot \Delta y_P - \frac{\varphi_{ref}}{\varphi_n}\left(\frac{\partial \psi}{\partial y}\right)_n \cdot \Delta x_P + \frac{\varphi_{ref}}{\varphi_w}\left(\frac{\partial \psi}{\partial x}\right)_w \cdot \Delta y_P + \frac{\varphi_{ref}}{\varphi_s}\left(\frac{\partial \psi}{\partial y}\right)_s \cdot \Delta x_P = 0$$

The stream function discretization equation gives the equation we need to solve the internal control volumes :

$$-\frac{\varphi_{ref}}{\varphi_e}\frac{\psi_E - \psi_P}{d_{PE}}\Delta y_P - \frac{\varphi_{ref}}{\varphi_n}\frac{\psi_N - \psi_P}{d_{PN}}\Delta x_P + \frac{\varphi_{ref}}{\varphi_w}\frac{\psi_P - \psi_W}{d_{PW}}\Delta y_P + \frac{\varphi_{ref}}{\varphi_s}\frac{\psi_P - \psi_S}{d_{PS}}\Delta x_P = 0$$

$$a_p\psi_p = a_E\psi_E + a_W\psi_W + a_N\psi_N + a_S\psi_S + b_p$$

$$a_E = \frac{\varphi_{ref}}{\varphi_e}\frac{\Delta y_P}{d_{PE}};\ a_W = \frac{\varphi_{ref}}{\varphi_w}\frac{\Delta y_P}{d_{PW}};\ a_N = \frac{\varphi_{ref}}{\varphi_n}\frac{\Delta x_P}{d_{PN}};\ a_S = \frac{\varphi_{ref}}{\varphi_s}\frac{\Delta x_P}{d_{PS}}$$

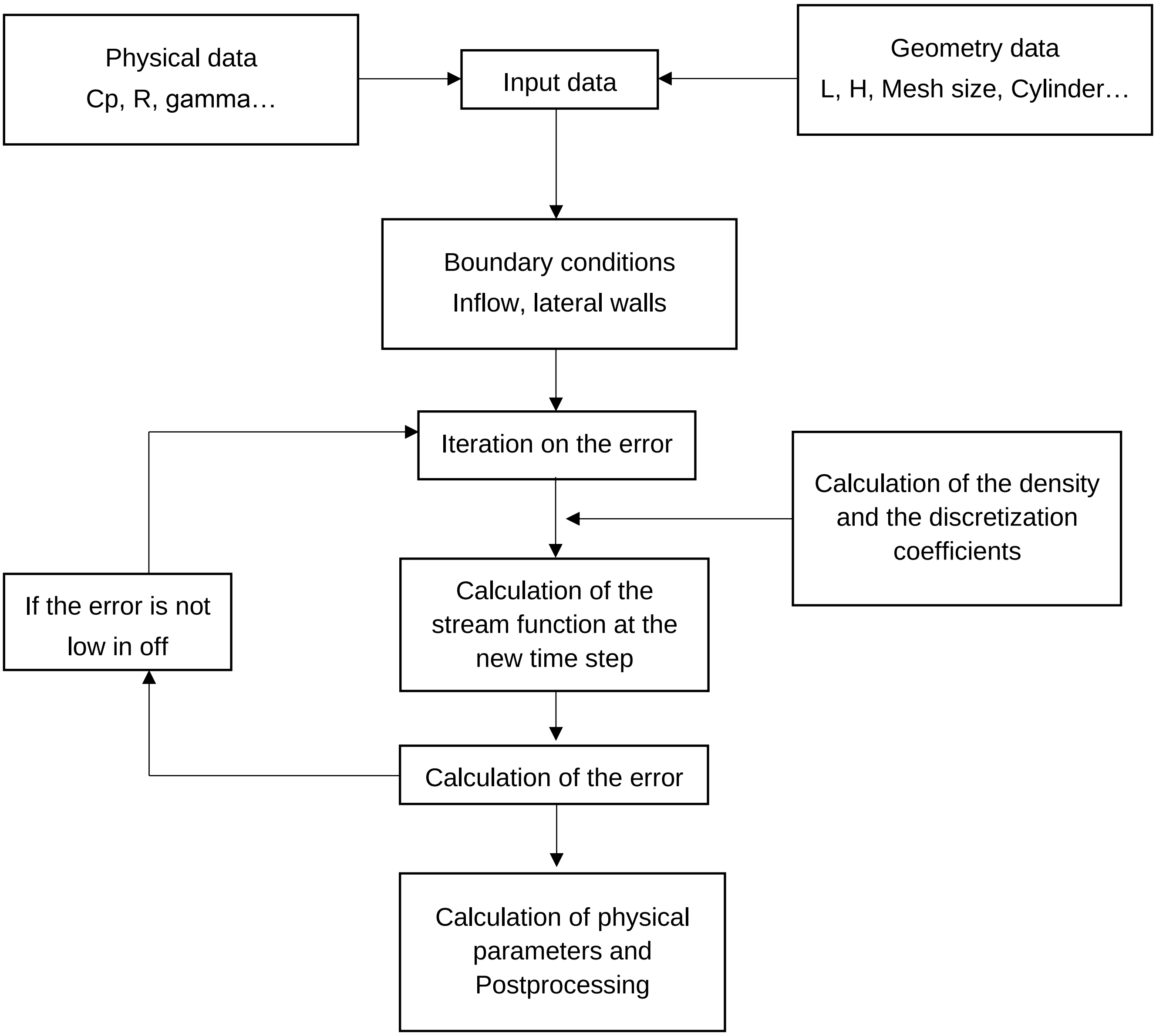
$$a_P = a_E + a_W + a_N + a_S\ ; \quad b_P = 0$$

Then we know how to solve the coefficient and the stream function for every control volume (interior).

$$a_p\psi_p = (a_E\psi_E + a_W\psi_W + a_N\psi_N + a_S\psi_S + b_p)(\frac{1}{a_p})$$

3 CODE STRUCTURE

The code structure is the following :



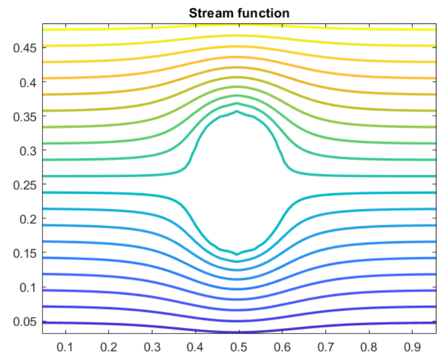
4 CODE VERIFICATION

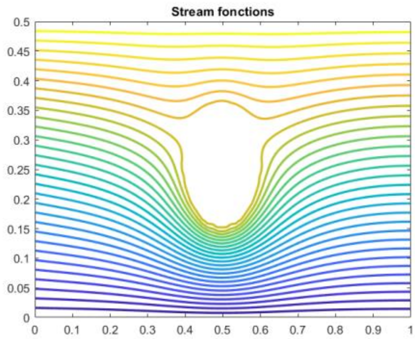
We are comparing the results to the exact results in the case of the potential flow without cylinder. When I confirmed this part I have used the same code for the cylinder (improving the code and so on for the rotation.).

We also can look at the circulation and other physical constant. For example, we can look at the drag and the lift for the cylinder without rotation =0.

5 RESULTS

5.1 Stream fonctions

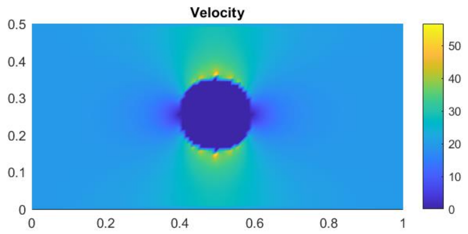


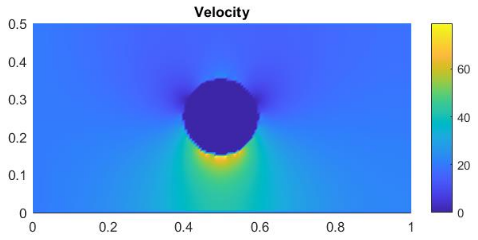


On the left side we can see the stream functions for the flow channel without rotation in that case we can see a good result. We can also see the stream function as small discontinuities on top and bottom of the cylinder. This is linked to the mesh we use to solve this issue we can consider more control volumes to have smaller control volume.

On the right side we can see the result for the stream function with the cylinder in rotation.

5.2 Velocity

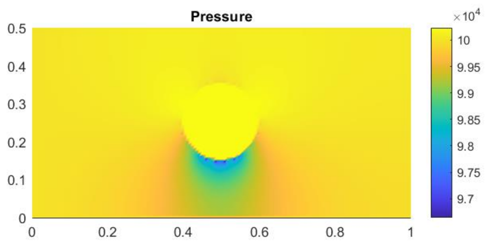


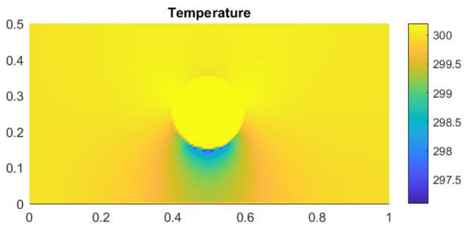


On the left the velocity is higher on top and bottom of the cylinder as it as no rotation.

The right side show a high velocity on the bottom of the cylinder witch is linked to the rotation.

5.3 Pressure and temperature with rotation





We can se a similar distribution for the pressure and the temperature for the same rotation with different values. To have more interesting properties we should study a compressible flow.

5.4 Physics coefficient

The Lift and drag coefficient should be equal to zero if the cylinder as no rotation.

In case of rotation we will have a value not equal to zero for the lift and the drag.

The calculation wasn't made for time consuming reasons. Those calculations are good to confirm the code without the rotation but are really interesting for rotational study. Then we can do the study to change the geometry to study the lift and the drag of airfoils for example.

6 CONCLUSION

To conclude, I managed to obtain the requested results for the velocity pression and stream functions. In order to work on all the projects and to have some results I wasn't able to compute the drag, the lift and the circulation. I would like to do it to have a really complete code. An over point would be to use a compressible model for higher Mach number. For simplicity reasons I only did a uniform mesh so to improve the performance of the code It would be great to have a finer mesh near the cylinder.

I think that with my limited capacities in coding when I arrived , I managed to do a report with the requested results. I found those projects interesting because we really apply the theory into the codes witch is something I am not used to.

7 ANNEXE

7.1 Code rotation

```
% Ex 1 flow in a channel with rotation
% Auteur : Boulogne Quentin

clear all;
close all
clc;
%% 1-Input Data
tic;
%Domain lengths
Lx=1;% L length along positive x axis [m]
Ly=0.5;% H hight along positive y axis [m]
domainLengths=[Lx Ly];% domain size [m]

%Mesh sizes
Nx=160; %Number of counter volumes on x axis
Ny=80;%Number of counter volumes on y axis
meshSizes=[Nx Ny];% Number of counter volumes on each axis

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

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

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

% Cylinder definition
solid = zeros(Npy,Npx);
Cylx=Npx/2;
Cyly=Npy/2;
rayon=0.1; %[m]
for i=1:Npy
    for j=1:Npx
        d=sqrt((dx*(j-Cylx))^2+(dy*(i-Cyly))^2);
        if d<=rayon
            solid(i,j)=1; % definition of the  solid
        end
    end
end
```

```
% Fluid properties
Cp=1005;
R=287.058;
gamma=1.4;

% Initial flow conditions
Tin=300; % Temperature inflow [K]
Pin=1e5; % Pressure inflow [Pa]
Vin=20; % Initial Speed [m/s]
rhoin=Pin/R*Tin; % initial density [kg/m3]

% Computation data
error=1e-6; % error condition
max_number_iter=1e6;

%% 2-Initialisation of matrices
% Physics properties
Streamf=ones(Npy,Npx);
Temp=Tin*ones(Npy,Npx);
Pres=Pin*ones(Npy,Npx);
rho=rhoin*ones(Npy,Npx);
Vx=Vin*ones(Npy,Npx);
Vy=zeros(Npy,Npx);
V=Vx+Vy;

%% 3-Boundary conditions

Streamf(:,1)=Vin*y;
Streamf(1,:)=Vin*y(1);
Streamf(Npy,:)=0;
Streamfold=Streamf;

%% In the solid
for i=1:Npy
   for j=1:Npx
      if solid(i,j)==1
         Streamf(i,j)=Vin*y(1)*3/4;% rotation condition 1/2 no rotation
         Temp(i,j)=0;
         Pres(i,j)=0;
         rho(i,j)=0;
         Vx(i,j)=0;
      end
   end
end

%% 4-Coefficients
```

```
rhoN = zeros(Npy,Npx);
rhoW = zeros(Npy,Npx);
rhoS = zeros(Npy,Npx);
rhoE = zeros(Npy,Npx);

aN=zeros(Npy,Npx);
aW=zeros(Npy,Npx);
aE=zeros(Npy,Npx);
aS=zeros(Npy,Npx);
aP=zeros(Npy,Npx);
bP=zeros(Npy,Npx);

errorini=1;
iteration=0;

   while (errorini>error)
      for i=2:Npy-1
         for j=2:Npx-1
            if solid(i,j)==0
               rhoN(i,j)=dy/((dy/2)/(rhoin/rho(i,j))+(dy/2)/(rhoin/rho(i-1,j))); %harmonic
mean
               rhoS(i,j)=dy/((dy/2)/(rhoin/rho(i,j))+(dy/2)/(rhoin/rho(i+1,j)));
               rhoE(i,j)=dx/((dx/2)/(rhoin/rho(i,j))+(dx/2)/(rhoin/rho(i,j+1))); %DENSITY
               rhoW(i,j)=dx/((dx/2)/(rhoin/rho(i,j))+(dx/2)/(rhoin/rho(i,j-1)));

               aN(i,j)=rhoN(i,j)*dx/dy;% discretization coefficient
               aS(i,j)=rhoS(i,j)*dx/dy;
               aE(i,j)=rhoE(i,j)*dy/dx;
               aW(i,j)=rhoW(i,j)*dy/dx;
               bP(i,j)=0; %in this case bp is equal to 0
               aP(i,j)=aN(i,j)+aS(i,j)+aE(i,j)+aW(i,j);

               Streamf(i,j) = (aE(i,j)*Streamf(i,j+1)+aW(i,j)*Streamf(i,j-1)+... % Stream
fonction calculation
                     aN(i,j)*Streamf(i-1,j)+aS(i,j)*Streamf(i+1,j)+...
                     bP(i,j))/aP(i,j);
            end
         end

      end


   Streamf(:,Npx)=Streamf(:,Npx-1);% Outflow
   errorini=max(max(abs(Streamf-Streamfold))); % convergence criteria
   Streamfold=Streamf;% new stream function for next iteration
   iteration=iteration+1;%Calculation of the number of iteration
   end
```

```
%% 5-Computation of the other parameters

for i=2:Npy-1
   for j=2:Npx-1
      VxN=rhoN(i,j)*((Streamf(i-1,j)-Streamf(i,j))/dy);
      VxS=rhoS(i,j)*((Streamf(i,j)-Streamf(i+1,j))/dy);
      VyE=-rhoE(i,j)*((Streamf(i,j+1)-Streamf(i,j))/dx);
      VyW=-rhoW(i,j)*((Streamf(i,j)-Streamf(i,j-1))/dx);
      Vx(i,j)=(VxN+VxS)/2;
      Vy(i,j)=(VyE+VyW)/2;
      V(i,j)=sqrt(Vx(i,j)^2+Vy(i,j)^2); % Velocity [m/s]
      Temp(i,j)=Tin+(Vin^2-V(i,j)^2)/(2*Cp); % Temperature [K]
      Pres(i,j)=Pin*(Temp(i,j)/Tin)^(gamma/(gamma-1)); %Pressure calculation [Pa]
      rho(i,j)=Pres(i,j)/(R*Temp(i,j)); % Density calculation [kg/m^3]

   end
end
time_computation=toc;
%% 6-Post processing
C_p=zeros(Npy,Npx);
for i=1:Npy
   for j=1:Npx
      C_p(i,j) = 1-(V(i,j)/Vin)^2;
   end
end
C_pmax=max(max(C_p));
figure(1)
contour(X,Y,C_p)

Mach=max(V)/345;

figure('Name','Streamfonctions','NumberTitle','off');
contour(X,Y,Streamf,30,'LineWidth',2);
title("Stream fonctions");

figure('Name','Flow velocity','NumberTitle','off');
quiver(X,Y,Vx,Vy);
title("Flow velocity");

figure;
pcolor(X,Y,V);
colorbar
shading interp;
axis equal
title('Velocity')
axis tight
```

```
grid on

figure;
pcolor(X,Y,rho);
colorbar
shading interp;
axis equal
title('Density')
axis tight
grid on

figure;
pcolor(X,Y,Pres);
colorbar
shading interp;
axis equal
title('Pressure')
axis tight
grid on

figure;
pcolor(X,Y,Temp);
colorbar
shading interp;
axis equal
title('Temperature')
axis tight
grid on
```