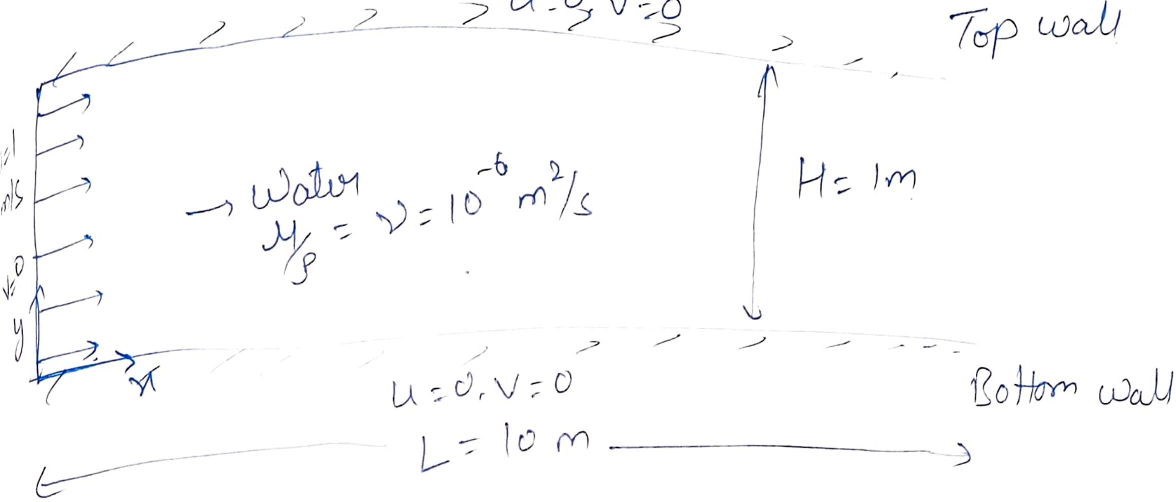
Computational Fluid dynamics

Assignment - 3 Mehul Verma - CH20BTECH11022

No-Slip, No-penetration

$\Rightarrow u=0, v=0$



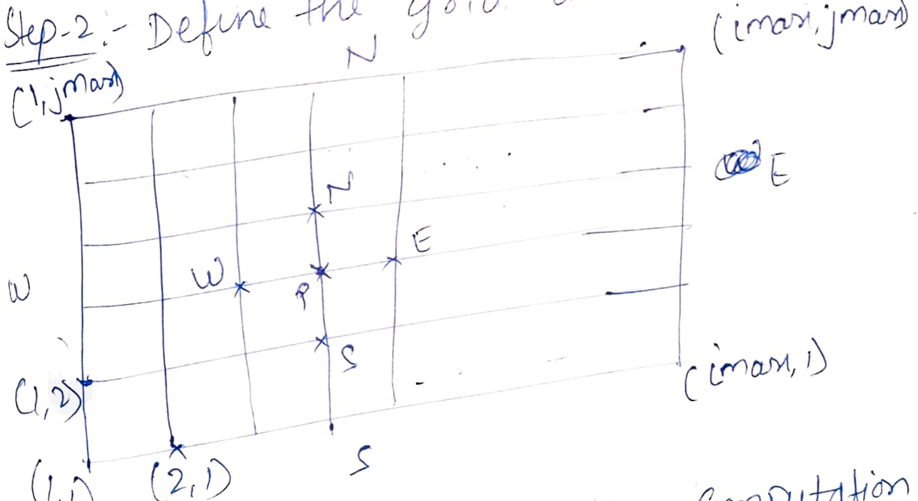
Step-1:- Governing equations :-

$$\rho\left[\frac{\delta \omega}{\delta t} + u\frac{\delta \omega}{\delta x} + v\frac{\delta \omega}{\delta y}\right] = \mu\left[\frac{\delta^2 \omega}{\delta x^2} + \frac{\delta^2 \omega}{\delta y^2}\right] \quad -(1)$$

$$\omega = -\left(\frac{\delta^2 \psi}{\delta x^2} + \frac{\delta^2 \psi}{\delta y^2}\right) \quad -(2) \qquad \omega = \text{vorticity}$$

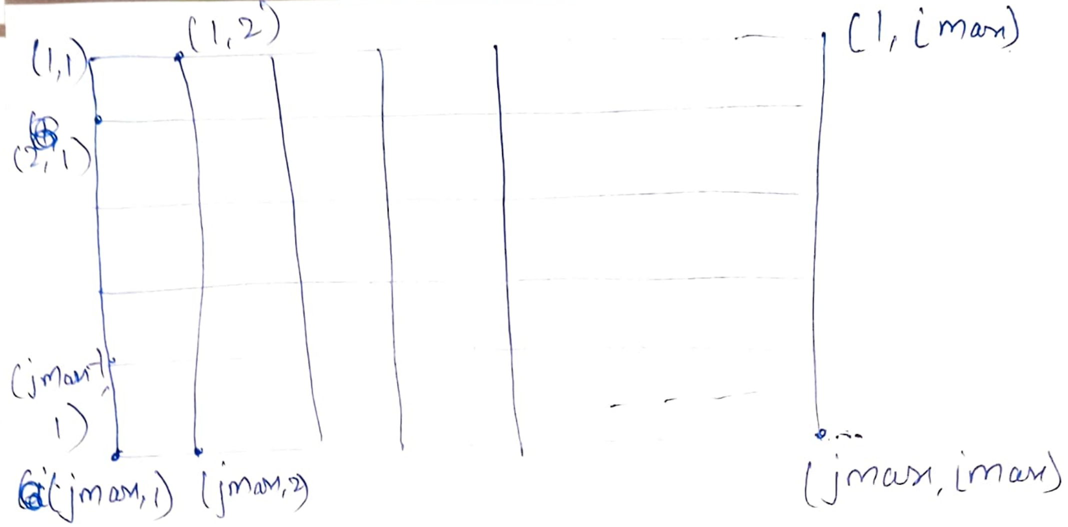
$$u = \frac{\delta \psi}{\delta y}, \quad v = \frac{-\delta \psi}{\delta x} \quad -(3) \qquad \psi = \text{Streamfunction}$$

Step-2:- Define the grid and discretisation schemes



Computational domain

$\rightarrow$ But in a matlab matrix would be like this.



Step-3 The discretised methods

$$\frac{\omega_{i,j}^{n+1}-\omega_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\left(\frac{\omega_{i+1,j}^{n}-\omega_{i-1,j}^{n}}{2\Delta x}\right)+v_{i,j}^{n}\left(\frac{\omega_{i,j+1}^{n}-\omega_{i,j-1}^{n}}{2\Delta y}\right)$$

$$=\nu\left[\frac{\omega_{i+1,j}^{n}-2\omega_{i,j}^{n}+\omega_{i-1,j}^{n}}{\Delta x^2}+\frac{\omega_{i,j+1}^{n}-2\omega_{i,j}^{n}+\omega_{i,j-1}^{n}}{\Delta y^2}\right] \quad -(1a)$$

$$\omega_{i,j}^{n+1}=-\left[\frac{\psi_{i+1,j}^{n}-2\psi_{i,j}^{n+1}+\psi_{i-1,j}^{n}}{\Delta x^2}+\frac{\psi_{i,j+1}^{n}-2\psi_{i,j}^{n+1}+\psi_{i,j-1}^{n}}{\Delta y^2}\right] \quad -(2a)$$

$$u_{i,j}^{n+1}=\frac{\psi_{i,j+1}^{n+1}-\psi_{i,j-1}^{n+1}}{2\Delta y}, \quad v=-\left(\frac{\psi_{i+1,j}^{n+1}-\psi_{i-1,j}^{n+1}}{2\Delta x}\right) \quad -(3a)$$

Step-4 :- Solution method

Initial Conditions :- $u=0;\ v=0,\ \psi=0,\ \omega=0$

Boundary Conditions :- Left wall :-

$u(1,j)=1,$

$v(1,j)=0$

$\omega(1,j)=0$

$\psi(1,j)=y=(j-1)\Delta y$

Right wall :-

$\frac{\delta u}{\delta x}=0, \ \frac{\delta v}{\delta x}=0, \ \frac{\delta w}{\delta x}=0, \ \frac{\delta \psi}{\delta x}=0$ (Neumann BC)

it means. $u(i_{max}, j) = u(i_{max}-1, j)$
$v(i_{max}, j) = v(i_{max}-1, j)$
$w(i_{max}, j) = w(i_{max}-1, j)$
$\psi(i_{max}, j) = \psi(i_{max}-1, j)$

North wall: $u=0, \ v=0$ [No slip, No penetration]
$\psi = (J_{max}-1)\Delta y$. $\ w = -\frac{\delta^2 \psi}{\delta y^2}$

South wall: $u=0, \ v=0, \ w=0, \ \psi=0$

Method :- Calculate $w_{i,j}^{n+1}$ at the new time step at all interior grid points (eq 1a)
Now, eq (2a), Calculate $\psi_{i,j}^{n+1}$ using information from previous time step.

$$\psi_{i,j}^{n+1} = \left(\frac{\Delta x^2 \Delta y^2}{2(\Delta x^2+\Delta y^2)}\right)\left(\frac{\psi_{i,j+1}^n + \psi_{i,j-1}^n}{\Delta x^2} + \frac{\psi_{i-1,j}^n + \psi_{i+1,j}^n}{\Delta y^2} + w_{i,j}^n\right) \quad \text{--- (4)}$$

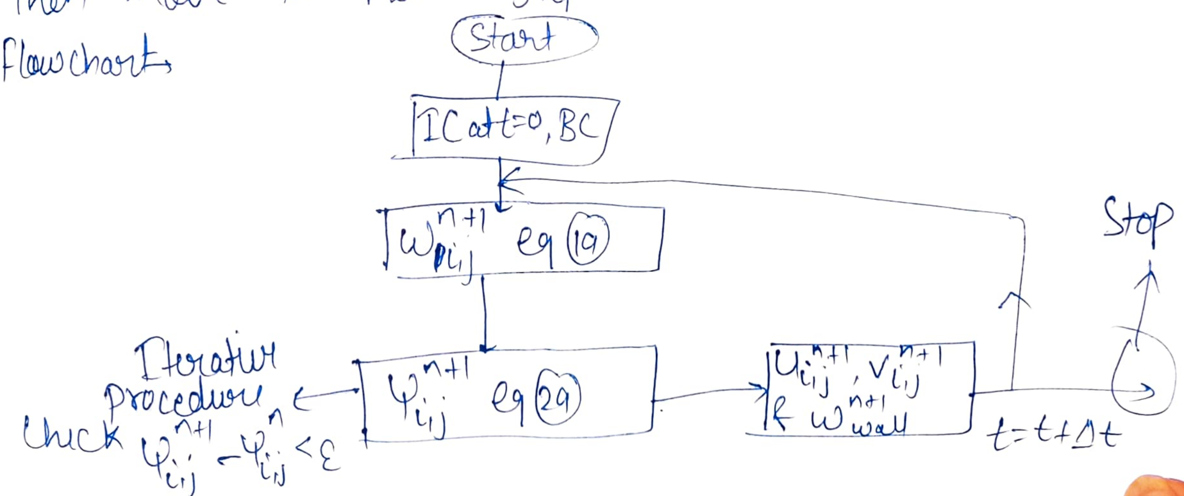
Loop eq (4) until residue $\psi_{i,j}^{n+1} - \psi_{i,j}^{n} \le \varepsilon \rightarrow 1\times10^{-6}$

If $\psi_{i,j}^{n+1} - \psi_{i,j}^{n} > \varepsilon$, then repeat [Gauss-Seidal Method]

Calculate $u_{i,j}^{n+1}, \ v_{i,j}^{n+1}, \ w_{wall}^{n+1} \rightarrow$ (3a)

Then move to next step.

Flowchart



Given Information

```
clc;clear;close all;

imax =32;
jmax = 32;
u_init =0.002;
re = 1000;
% re = u_init * H / nu
% u_init and H is adjusted so that Reynold's number is within laminar flow

L=1;
H=0.5;
nu = 1e-6;
e = 1e-6;% e to be used in Gauss seidal method
```

Initial Conditions

```
u0 = zeros(jmax,imax);
v0=zeros(jmax,imax);
u = zeros(jmax,imax);
v=zeros(jmax,imax);
w=zeros(jmax,imax);
w0 = zeros(jmax,imax);
psi0=zeros(jmax,imax);
psi=zeros(jmax,imax);

% u,v,w,psi correspond to (n+1)th time (New time points)
% u0,v0,w0,psi0 correspond to (n)th time (Old time points)

delx = L/(imax-1);
dely = H/(jmax-1);

y_axis = 0:dely:H;
tmax =10;
delt = 0.005;
nt = tmax/delt;
```

Boundary Conditions

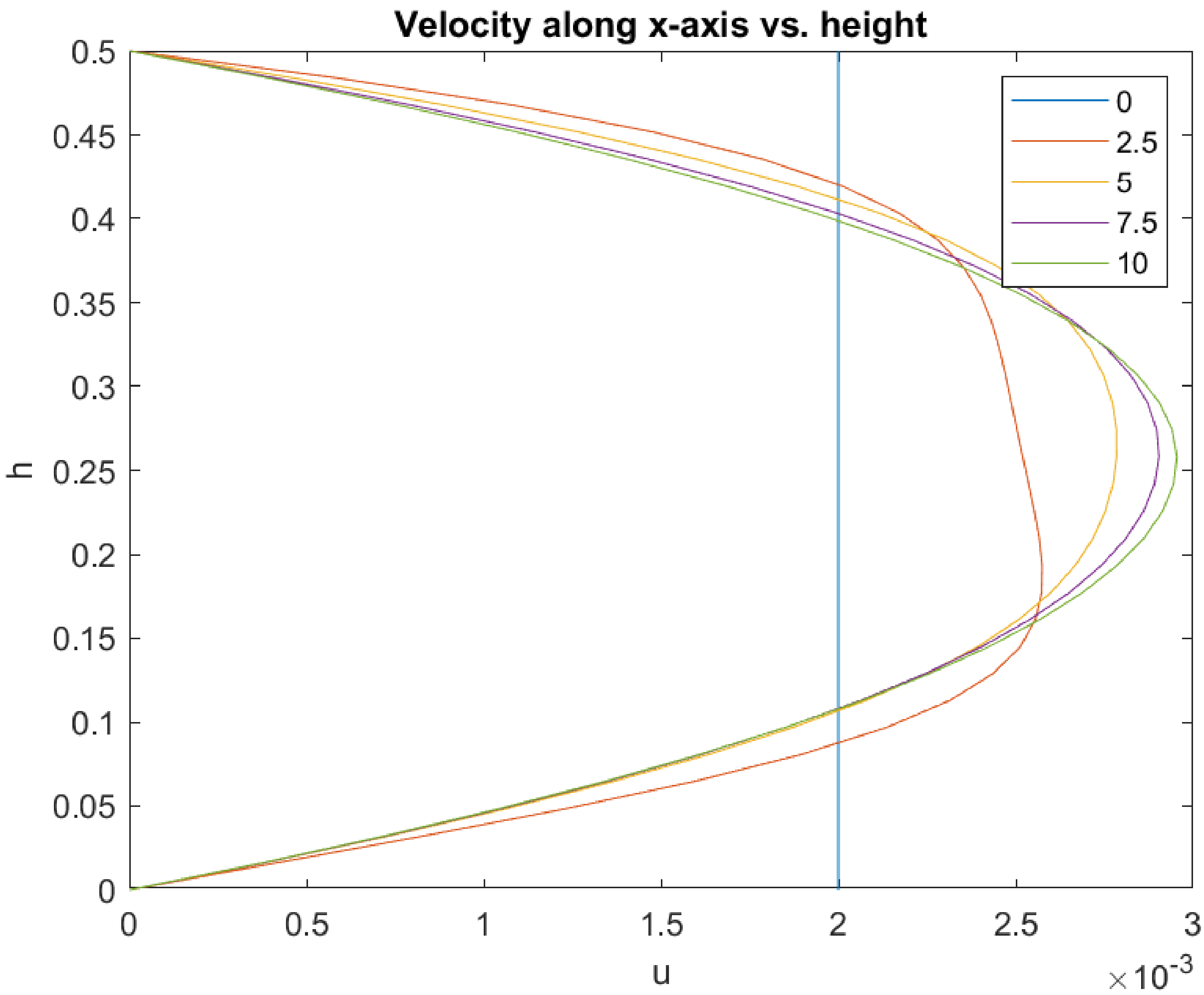
```
%top wall
u0(1,:)=0;
v0(1,:)=0;
psi0(1,:) = u_init*H;

%bottom wall
u0(jmax,:) = 0;
v0(jmax,:)=0;
psi0(jmax,:) = 0;

%left wall
u0(:,1) =u_init;
v0(:,1)=0;
psi0(:,1) = u0(:,1).* (H:-dely:0)';

%updating vorticity for top wall
```

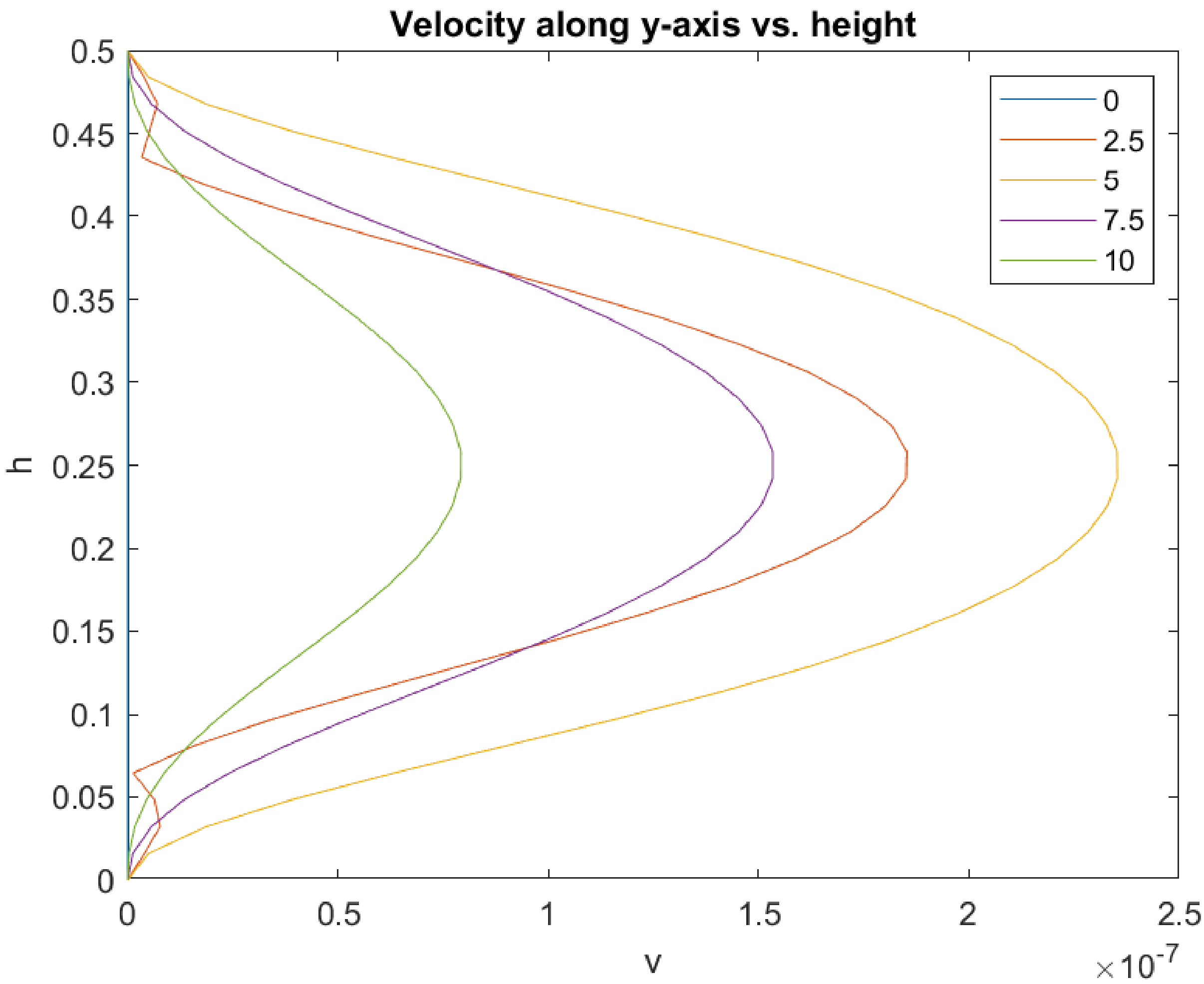
```
for i=1:imax
    w0(1,i) = -1* ((u0(1,i)-u0(2,i))/dely);
end

w=w0;
u=u0;
v=v0;
psi=psi0;
```

Main Iteration Loop

```
t=0;
plotTimings = [0 nt/4 nt/2 3*(nt/4) nt];
if (any(plotTimings(:) == t))
    figure(1);
    plot(u(:, 1), y_axis);
    hold on;
    figure(2);
    plot(abs(v(:, 1)), y_axis);
    hold on;
end
while(t < nt)
    %% calculating w at new time step
    for i= 2:imax-1
        w(1, i) = 2 * (psi0(1, i) - psi0(2, i)) / dely^2;
        w(jmax, i) = 2 * (psi0(jmax, i) - psi0(jmax - 1, i)) / dely^2;
    end

    for i=2:imax-1
        for j=2:jmax-1
            temp = (1/re)*( (w0(j,i-1) -2*w0(j,i) + w0(j,i+1))/delx^2 + (w0(j-1,i)
-2*w0(j,i) + w0(j+1,i))/dely^2 );
            w(j,i) = delt*( temp - u0(j,i)*( (w0(j,i+1) - w0(j,i-1)) / (2*delx) )
- v0(j,i)*( (w0(j-1,i) - w0(j+1,i)) / (2*dely) )) + w0(j,i);
        end
    end
    w(:,imax) = w(:,imax-1);


    %% psi at new time step and also implementing gauss seidal method
    satisfied = false;
    while (satisfied == false)
        for i=2:imax-1
            for j=2:jmax-1
                psi(j,i) = 0.5 * ((delx^2 * dely^2)/(delx^2 + dely^2)) * (w(j,i) +
(psi0(j,i+1) + psi0(j,i-1))/delx^2 + (psi0(j+1,i) + psi0(j-1,i))/dely^2);
            end
        end
        psi(:,imax) = psi(:,imax-1);
        curr = true;
        for i=2:imax-1
            for j=2:jmax-1
                if(abs(psi(j,i)-psi0(j,i)) > e)
                    curr = false;
                end
            end
```

```
        end
        if(curr==true)
            satisfied = true;
        end
        psi0 = psi;
    end


    %% calculating u, v at new time step
    for i=2:imax-1
        for j=2:jmax-1
            u(j,i) = (psi(j-1,i) - psi(j+1,i))/(2*dely);
            v(j,i) = -1 * (psi(j,i+1) - psi(j,i-1))/(2*delx);
        end
    end
    u(:,imax) = u(:,imax-1);
    v(:,imax) = v(:,imax-1);

    %% Updating old time step info with the new time step's
    w0=w;
    u0=u;
    v0=v;
    t=t+1;

    %% Plotting u, v on y-axis
    if (any(plotTimings(:) == t))
        figure(1);
        plot(u(:, imax), y_axis);
        hold on;
        figure(2);
        plot(abs(v(:, jmax)), y_axis);
        hold on;
    end
end
figure(1);
title('Velocity along x-axis vs. height');
xlabel('u');
ylabel('h');
legend('0',num2str(nt * delt / 4),num2str(nt * delt / 2), num2str(3 * nt * delt /
4), num2str(nt * delt));
hold off;
figure(2);
title('Velocity along y-axis vs. height');
xlabel('v');
ylabel('h');
legend('0',num2str(nt * delt / 4),num2str(nt * delt / 2), num2str(3 * nt * delt /
4), num2str(nt * delt));
hold off;
```


Velocity along x-axis vs. height
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
h
0
0.5
1
1.5
2
2.5
3
u
×10^-3
0
2.5
5
7.5
10



Velocity along y-axis vs. height
h
v
$\times 10^{-7}$
0
2.5
5
7.5
10
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.5
1
1.5
2
2.5