ROLES AND RESPONSIBILITIES

- Developed code in matlab and interfaced it with openfoam to perform meshing task in openfoam
- Simulated fluid flow through pipe with icoFoam solver in open foam and compared its accuracy with Hagen Poiseuilles equation
- Plotted velocity profiles of fluid at different lengths of pipe and post processed results using paraview

CODE SNIPPETS AND RESULTS

```
D=0.02;%Diameter of the pipe
R=D/2; %Radius of the pipe
Re=2100; %Given Reynolds number
L_E=0.06*Re*D;%Entrance length
L=0.3+L_E; %Total length of the pipe
nu=0.0000010023; %Kinematic viscosity of water at 20 deg. Cel.
u=(nu*Re)/(D);
prompt = 'Wedge angle(the value should be less than 5degree)=';
theta=input(prompt);
prompt= 'Mesh Grading Factor=';
g_s=input(prompt);
v0=[0 0 0];
v1=[0 R*cosd(theta/2) -R*sind(theta/2)];
v2=[0 R*cosd(theta/2) R*sind(theta/2)];
v3=[L 0 0];
v4=[L R*cosd(theta/2) -R*sind(theta/2)];
v5=[L R*cosd(theta/2) R*sind(theta/2)];

f1=fopen('blockMeshDict.txt','w');
fprintf(f1,'%s\n','/*--------------------------------*- C++ -*----------------------------------*
\\');
fprintf(f1,'%s\n','  =========                 |');
fprintf(f1,'%s\n','  \\\\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox');
fprintf(f1,'%s\n','   \\\\    /   O peration     | Website:  https://openfoam.org');
fprintf(f1,'%s\n','    \\\\  /    A nd           | Version:  6');
fprintf(f1,'%s\n','     \\\\/     M anipulation  |');
```

```
/*--------------------------------*- C++ -*----------------------------------*\\
  =========                 |
  \\\\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\\\    /   O peration     | Website:  https://openfoam.org
    \\\\  /    A nd           | Version:  6
     \\\\/     M anipulation  |
\\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

convertToMeters 1;

vertices
(
        (0 0 0)
        (0 9.996573e-03 -2.617695e-04)
        (0 9.996573e-03 2.617695e-04)
        (2.820000e+00 0 0)
        (2.820000e+00 9.996573e-03 -2.617695e-04)
```