```
clc;
clear;
close all;
densityWater = 998;%kg/m^3
densityAir = 1.204;%kg/m^3 at room temp!!!
dynVisAir = 1.81*10^-5;%at room temp!
gravity = 9.81;%m/s^2
pipeDiam = 0.076; %in m
resolution = 100;
R = linspace (0 ,pipeDiam/2,resolution);
axialRange = 'B5:V19';
pitotRange = 'A5:K18';
labData =
 {readtable("EGME306B_Ex02_DataSheet_08",'Sheet',"Axial",'Range',axialRange),readtable("EG
%converting all manometer heights to meters
labData{1,1}{:,2:end} = labData{1,1}{:,2:end}/100;
labData{1,2}{:,2:end} = labData{1,2}{:,2:end}/100;
labData{1,2}{:,1} = labData{1,2}{:,1}/1000;

%Static Pressure Distribution (Static pressure == density * grav * )(p_ref -
 P_i))

%RPM set 1: [columns 1 to 5]
%RPM set 2: [columns 6 to 10]
%RPM set 3: [columns 11 to 15]
%RPM set 3: [columns 16 to 20]
axialPositions = labData{1,1}{1:end-1,1}; %since not using reference distance
axialPositions_mm = axialPositions * 1000;
manometerReadings = labData{1,1}{:,2:end};
referenceManometerReadings = manometerReadings(end,:);
manometerReadings = labData{1,1}{1:end-1,2:end};
hydroStaticPressures = (referenceManometerReadings - manometerReadings) *
 densityWater*gravity; %in Pa

%%Static Pressure Plots
staticPressurePlots = [];
plotNames = ["Static Pressure Distribution 1250RPM","Static Pressure
 Distribution 2000RPM","Static Pressure Distribution 2800RPM","Static Pressure
 Distribution 3600RPM"];
titles = ["$ Static \ Pressure \ Distribution \ Problem \ 2A \ 1250RPM \
 (Actual: \ 1248RPM) $","$ Static \ Pressure \ Distribution \ Problem \ 2A
 \ 2000RPM \ (Actual: \ 1995RPM) $","$ Static \ Pressure \ Distribution \
 Problem \ 2A \ 2800RPM \ (Actual: \ 2804RPM) $","$ Static \ Pressure \
 Distribution \ Problem \ 2A \ 36000RPM \ (Actual: \ 3632PM) $"];
legendStuff = {'$ 10 \% \ Open $','$ 30 \% \ Open $','$ 50 \% \ Open $','$ 70
 \% \ Open $', '$ 100 \% \ Open $'};
for i = 1:4 %(since 4 different RPMs)
    staticPressurePlots(i) = figure; %this way I don't have to manually
 delcare a new figure.
    for ii = 1:5 %since 5 different percentage openings [10,30,50,70,100]
        plot(axialPositions,hydroStaticPressures( : , ii+ 5*(i-1)) , "o-")
        % for example, if i = 2, and ii = 1, then 1+5(2-1) = 6, which is
```

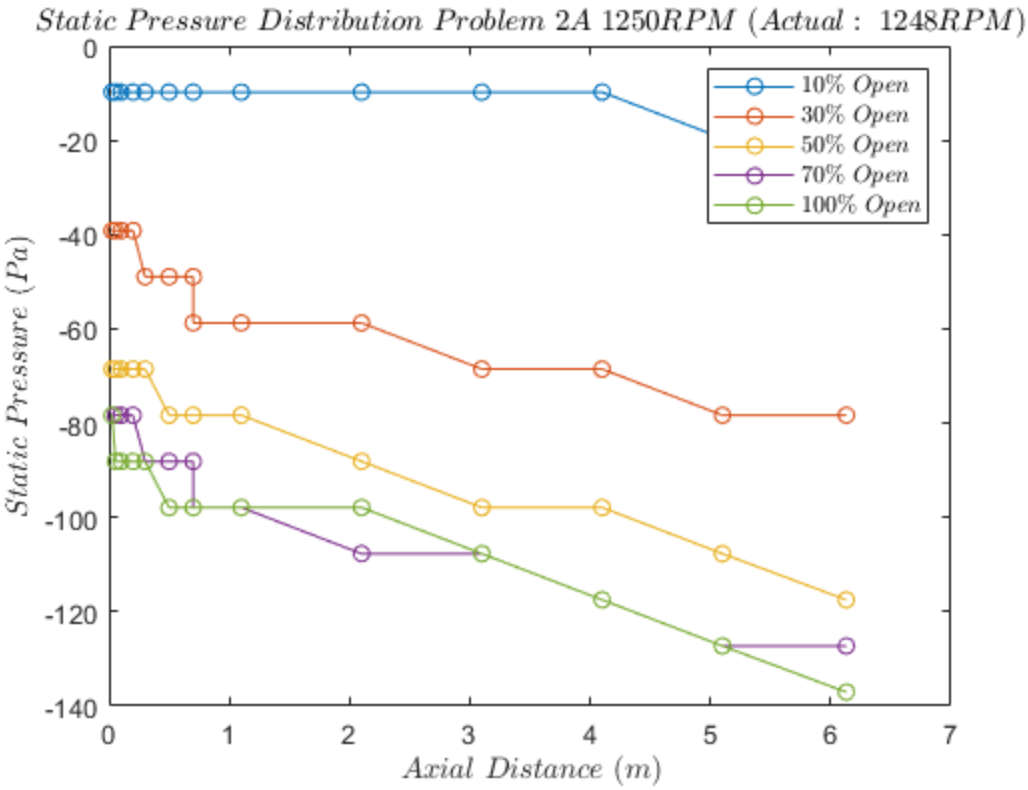
```
        % where RPM #2 starts
        hold on;
    end
    title(titles(i),'Interpreter','latex')
    lgnd= legend(legendStuff);
    set(lgnd, 'Interpreter','latex')
    xlabel("$ Axial \ Distance \ (m) $",'Interpreter','latex')
    ylabel("$Static \ Pressure \ (Pa)$",'Interpreter','latex')
    hold off;
    plot1 = plotNames(i);
    print('-r600','-dpng',plot1);
end

%%Entrance Lengths

%afterwards, entrance lengths are obtained by visually looking at the
%graph, and seeing where the pressure starts to linearly decrease.

%[ each row is RPM, and each column in percent opening]
entranceLengths = [ 4100 , 4100, 4100 , 2100, 3100 ; 3100, 700, 1100, 1100,
 1100; 1100, 700, 700, 700, 700; 1100, 700, 700, 700, 700 ]/1000; %in m
normalizedEntranceLengths = entranceLengths/pipeDiam;

RPMs = [1250,2000,2800,3600];
fanPercents = [10,30,50,70,100];

figure;
% Each line is specific to a percent opening, and each point is per RPM
for i = 1:5
    plot(RPMs,entranceLengths(:,i),"o-")
    hold on;
end
title("$ Entrance \ Length \ V.S \ Fan \ Speed $",'Interpreter','latex')
lgnd= legend(legendStuff);
set(lgnd, 'Interpreter','latex')
xlabel("$ Fan \ Speed \ (RPM) $",'Interpreter','latex')
ylabel("$Entrance \ Length \ (m)$",'Interpreter','latex')
hold off;
plot1 = "Entrance Length vs RPM";
print('-r600','-dpng',plot1);

%%Plotting Velocity Profiles With Parabolic Curve
normalizedPitotHeights = []; %each column is per percent opening, each row is
 the normalized height
for i = 1:5 % for the 5 different percent openings at 1250rpm
    %each static pressure is tap 14, and each of their positions is 2+2*(i-1)
    %each stagnation pressure is tap 19, and each of their positions is
 2+2&(i-1) + 1 == 3+2*(i-1)

    %is h_static - h_stagnation
    normalizedPitotHeights(:,i) = labData{1,2}{:,2+2*(i-1)} - labData{1,2}
{:,3+2*(i-1)};
end
```

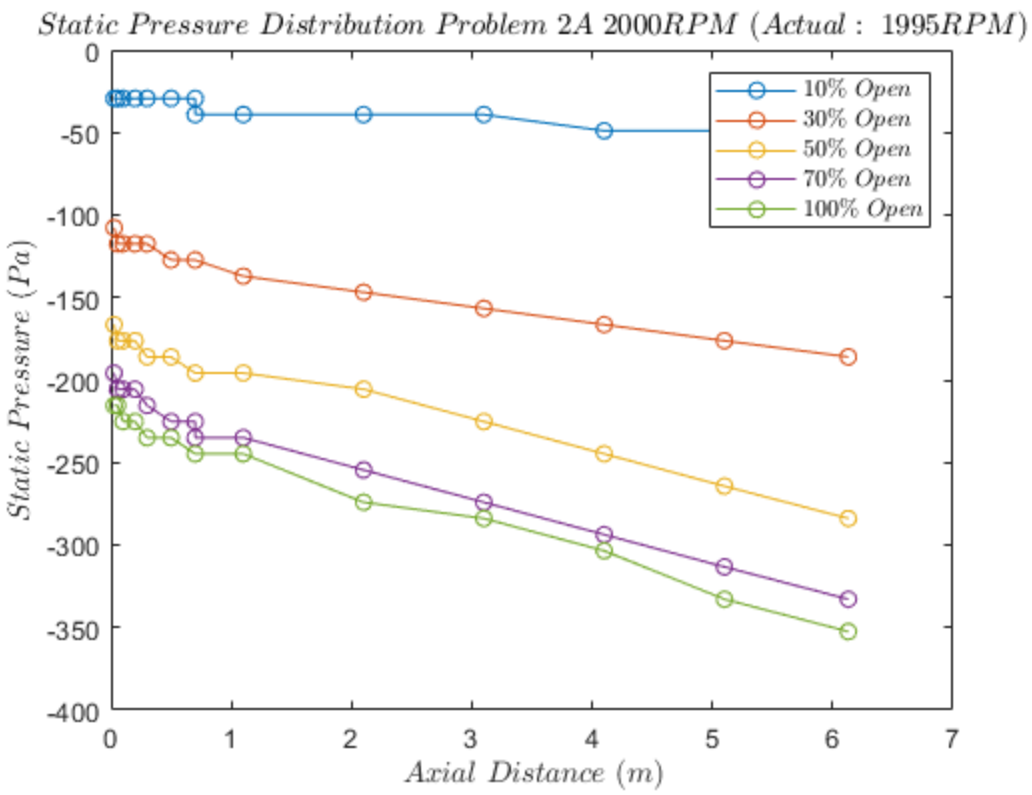
```
localVelocities = sqrt(2*densityWater*gravity*abs(normalizedPitotHeights)/
densityAir); %in m/s
legendStuff = {'$ 10 \% \ Open $','$ 30 \% \ Open $','$ 50 \% \ Open $','$ 70
 \% \ Open $', '$ 100 \% \ Open $'};

figure;
for i = 1:5
    pointPlot = plot(labData{1,2}{:,1}*1000,localVelocities(:,i), "o");
    hold on;
    eqn = polyfit(labData{1,2}{:,1}*1000,localVelocities(:,i) , 2);
    plot(labData{1,2}{:,1}*1000,polyval(eqn,labData{1,2}
{:,1}*1000),'color',get(pointPlot,'color'),'HandleVisibility','off')
    hold on;
end
title("$ Velocity \ Profile \ Parabolic \ Approximation \ Problem \ 2A
 $",'Interpreter','latex')
lgnd= legend(legendStuff);
set(lgnd, 'Interpreter','latex')
xlabel("$ Radial \ Distance \ (mm) $",'Interpreter','latex')
ylabel("$Local \ Velocity \ (m/s)$",'Interpreter','latex')
hold off;
plot1 = "Parabolic Velocity Profile Approximation";
print('-r600','-dpng',plot1);

%%Plotting Velocity Profiles With Power Curve
V2 = [];
figure;
for i = 1:5 %for each percent opening again
    initGuesses = [10.0,7.0];
    [c,~] =
 lsqcurvefit(@Nathan_Delos_Santos_power_func ,initGuesses ,labData{1,2}
{:,1},localVelocities(:,i));
    V2(:,i) = c(1)*(1- R/(pipeDiam/2)) .^(1/c(2));
    pointPlot = plot(labData{1,2}{:,1}*1000,localVelocities(:,i), "o");
    hold on;

 plot(R*1000,V2(:,i),'color',get(pointPlot,'color'),'HandleVisibility','off')
    hold on;
end
title("$ Velocity \ Profile \ Power \ Curve \ Approximation \ Problem \ 2A
 $",'Interpreter','latex')
lgnd= legend(legendStuff);
set(lgnd, 'Interpreter','latex')
xlabel("$ Radial \ Distance \ (mm) $",'Interpreter','latex')
ylabel("$Local \ Velocity \ (m/s)$",'Interpreter','latex')
hold off;
plot1 = "Power Curve Velocity Profile Approximation";
print('-r600','-dpng',plot1);

%%Calculating Volumetric Flowrate

%when numerically integrating, do relative from "edge", so relative from
%last data point
Q = zeros(1,5);
```

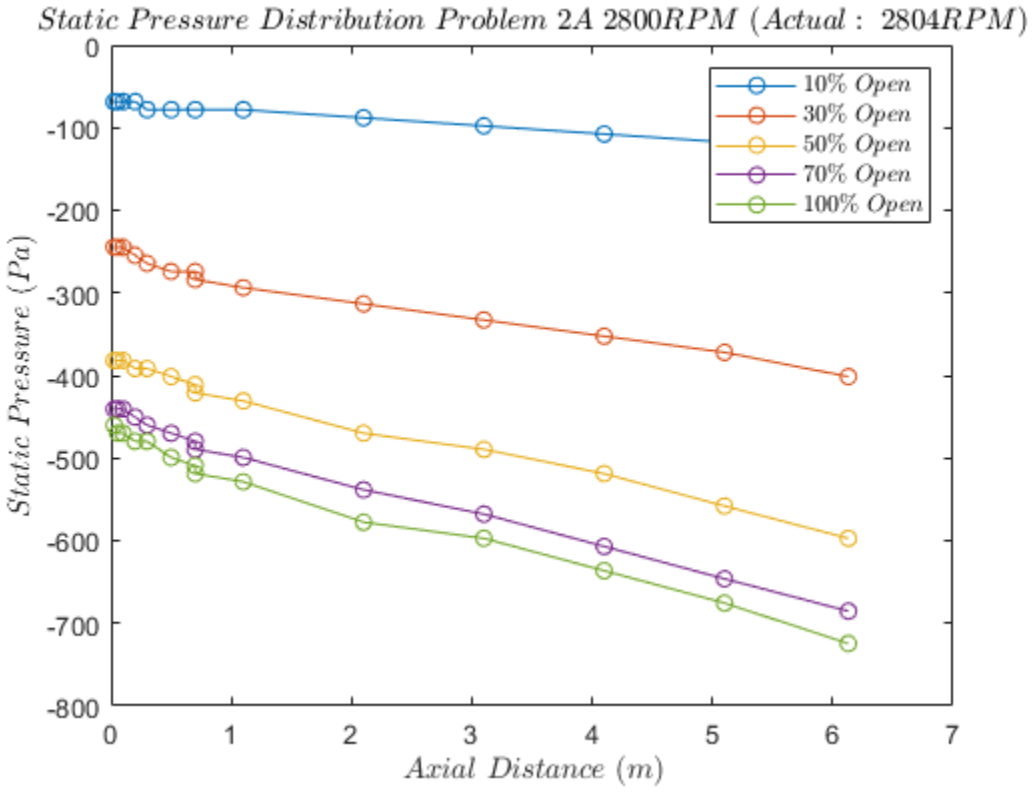
```
Re = zeros(1,5);
for i = 1:5
    for ii = 1:13
       Q(i) = Q(i) + 0.5*pi*(localVelocities(ii+1,i) +
 localVelocities(ii,i))...
           *(labData{1,2}{ii+1,1} + labData{1,2}{ii,1})*(labData{1,2}{ii+1,1}
 - labData{1,2}{ii,1});
    end
    Re(i) = 4.*densityAir*Q(i)/(pi*dynVisAir*pipeDiam);
end

figure;
% Entrance Length VS Re for 1250RPM
plot(Re,entranceLengths(1,:)/pipeDiam,"o-")
title("$ (Normalized) \ Entrance \ Length \ V.S \ \Re \ Problem \ 2A
 $",'Interpreter','latex')
xlabel("$ Reynolds \ Number \ , \ \Re $",'Interpreter','latex')
ylabel("$ (Normalized) Entrance \ Length \ (m)$",'Interpreter','latex')
hold off;
plot1 = "Normalized Entrance Length vs Re for 1250RPM";
print('-r600','-dpng',plot1);
```

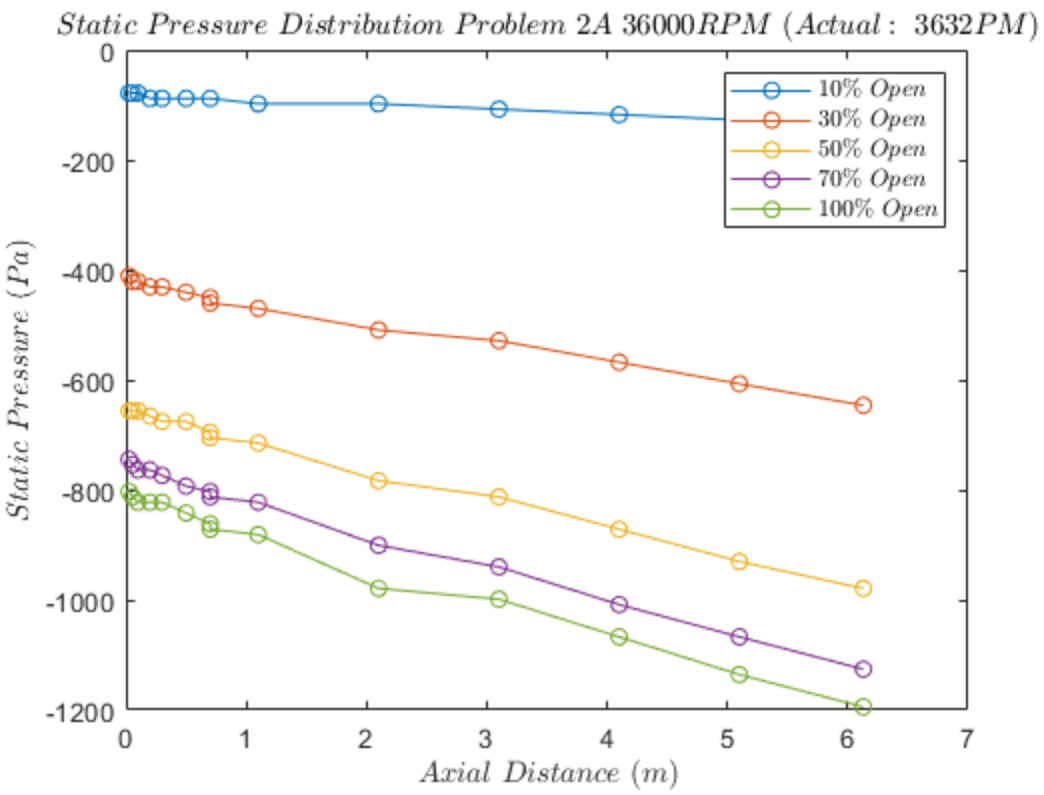
```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.


Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative
 to
its initial value is less than the value of the function tolerance.


Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative
 to
its initial value is less than the value of the function tolerance.


Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative
 to
its initial value is less than the value of the function tolerance.


Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative
 to
```

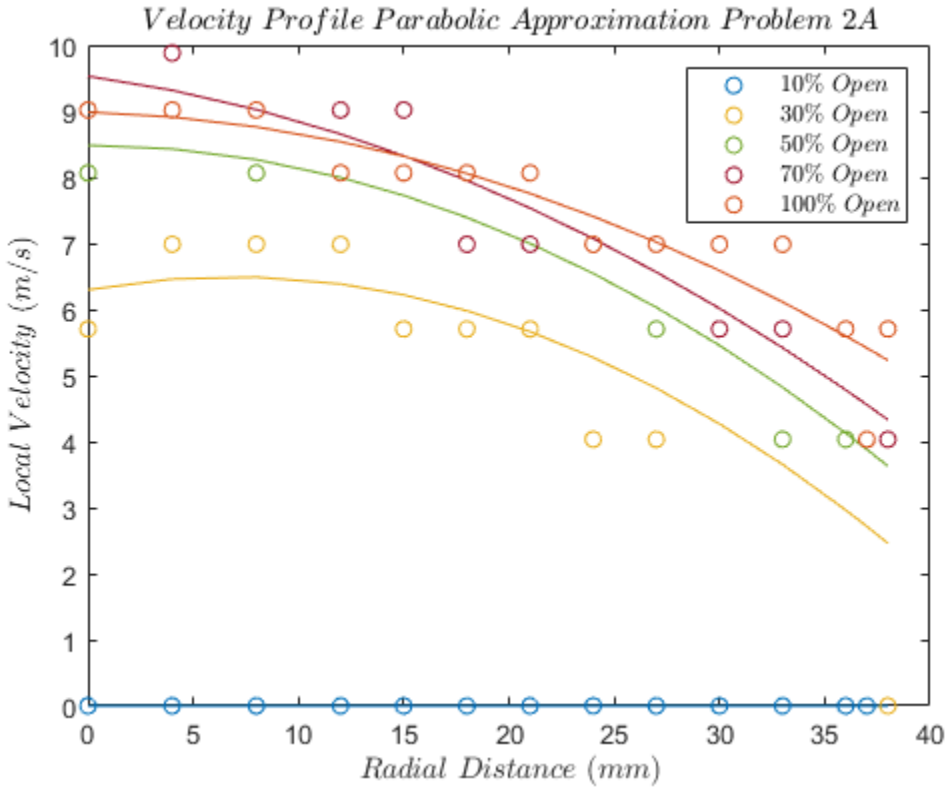
its initial value is less than the value of the function tolerance.

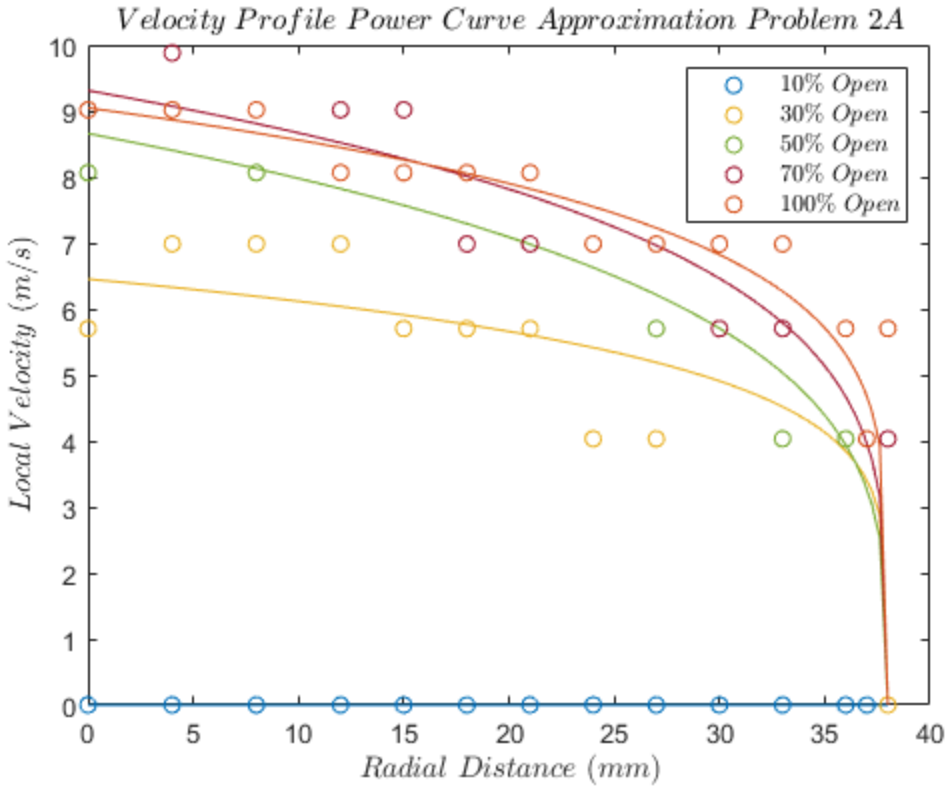


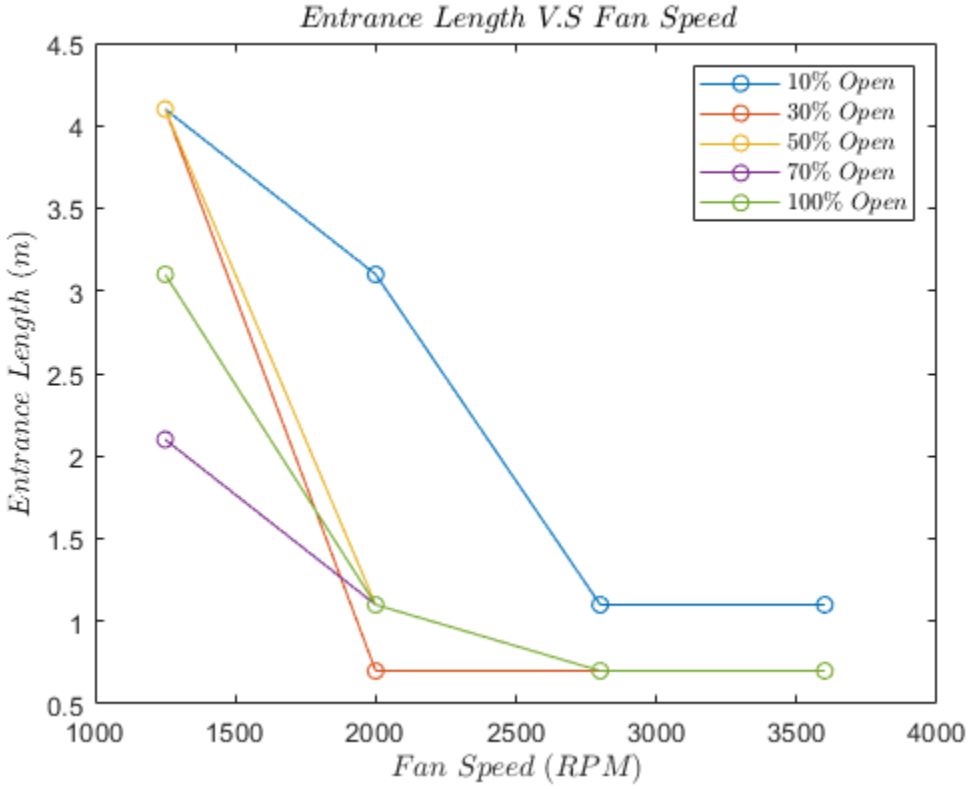

Static Pressure Distribution Problem 2A 2000RPM (Actual : 1995RPM)
10% Open
30% Open
50% Open
70% Open
100% Open
Static Pressure (Pa)
Axial Distance (m)
0
-50
-100
-150
-200
-250
-300
-350
-400
0
1
2
3
4
5
6
7

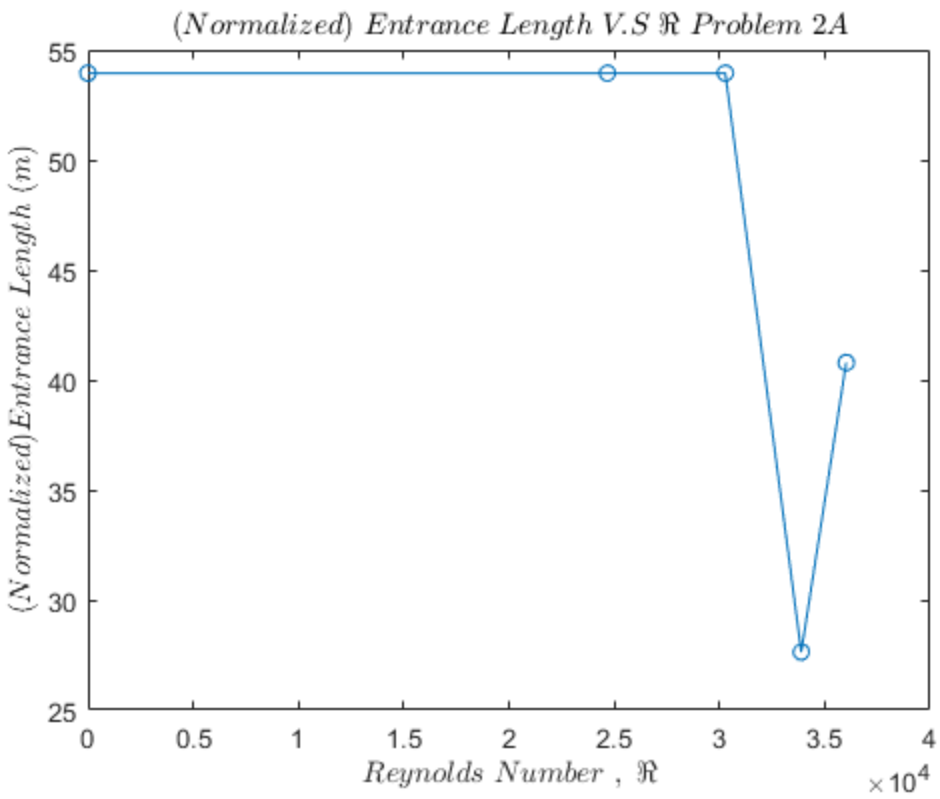

Static Pressure Distribution Problem 2A 2800RPM (Actual : 2804RPM)
10% Open
30% Open
50% Open
70% Open
100% Open
Static Pressure (Pa)
Axial Distance (m)
0
-100
-200
-300
-400
-500
-600
-700
-800
0
1
2
3
4
5
6
7


Static Pressure Distribution Problem 2A 36000RPM (Actual: 3632PM)
Static Pressure (Pa)
Axial Distance (m)
10% Open
30% Open
50% Open
70% Open
100% Open
0
-200
-400
-600
-800
-1000
-1200
0
1
2
3
4
5
6
7


Velocity Profile Parabolic Approximation Problem 2A
Local Velocity (m/s)
Radial Distance (mm)
10% Open
30% Open
50% Open
70% Open
100% Open
0
1
2
3
4
5
6
7
8
9
10
0
5
10
15
20
25
30
35
40


Velocity Profile Power Curve Approximation Problem 2A
10% Open
30% Open
50% Open
70% Open
100% Open
Local Velocity (m/s)
Radial Distance (mm)
0
1
2
3
4
5
6
7
8
9
10
0
5
10
15
20
25
30
35
40


Entrance Length V.S Fan Speed
10% Open
30% Open
50% Open
70% Open
100% Open
Entrance Length (m)
Fan Speed (RPM)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
1000
1500
2000
2500
3000
3500
4000



Published with MATLAB® R2022b