**Step 1: Problem Identification and Statement**

Determining the flow through pipes by calculating the Friction factor using an analytical method and a numeric method. This is to analyze the graphical relationship between conduit diameter, pipe roughness, fluid density, dynamic viscosity with the friction factor.

**Step 2: Gathering Information**

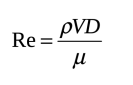
* Equation to calculate Reynolds number

Where:

r = the fluid’s density [kg/m3 ]

V = its velocity [m/s]

µ = dynamic viscosity [N s/m2 ]



* Re must be greater than 4000 to be valid for use in the coming equations.
* Analytical method to calculate friction factor

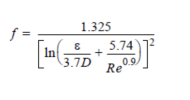
Where:

ƒ = friction factor

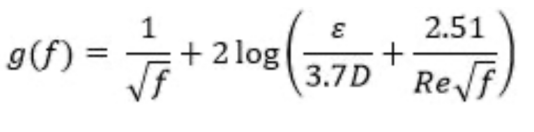
e = the roughness (m)

D = conduit diameter (m)

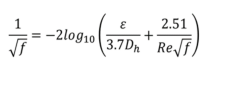
Re = the Reynolds number, defined by the following equation



* Numeric method to calculate friction factor using bisection method to calculate roots:



* The bisection method: it is done by using an upper and lower value for f, where a root lies between them. Then finding the midpoint of the f values and finding if the root is between the midpoint and the upper f or the midpoint and the lower f to narrow down the upper and lower bounds until they are smaller than the tolerance. Then the friction factor will be the final midpoint.
* *This equation is derived from Colebrook’s equation that calculates the friction factor*



Input/ Output

Analytically calculated Friction factor

Numerically calculated friction factor

Graphs of conduit diameter vs analytically and numerically calculated friction factors

Graphs of pipe roughness vs analytically and numerically calculated friction factors

Graphs of fluid density vs analytically and numerically calculated friction factors

Graphs of dynamic viscosity vs analytically and numerically calculated friction factors

Fluid Density

Velocity

Dynamic Viscosity

Conduit Diameter

Pipe Roughness

Upper and lower bounds of diameter

Upper and lower

bounds of roughness

Upper and lower

bounds of density

Upper and lower

bounds of viscosity

**Step 3: Test cases**

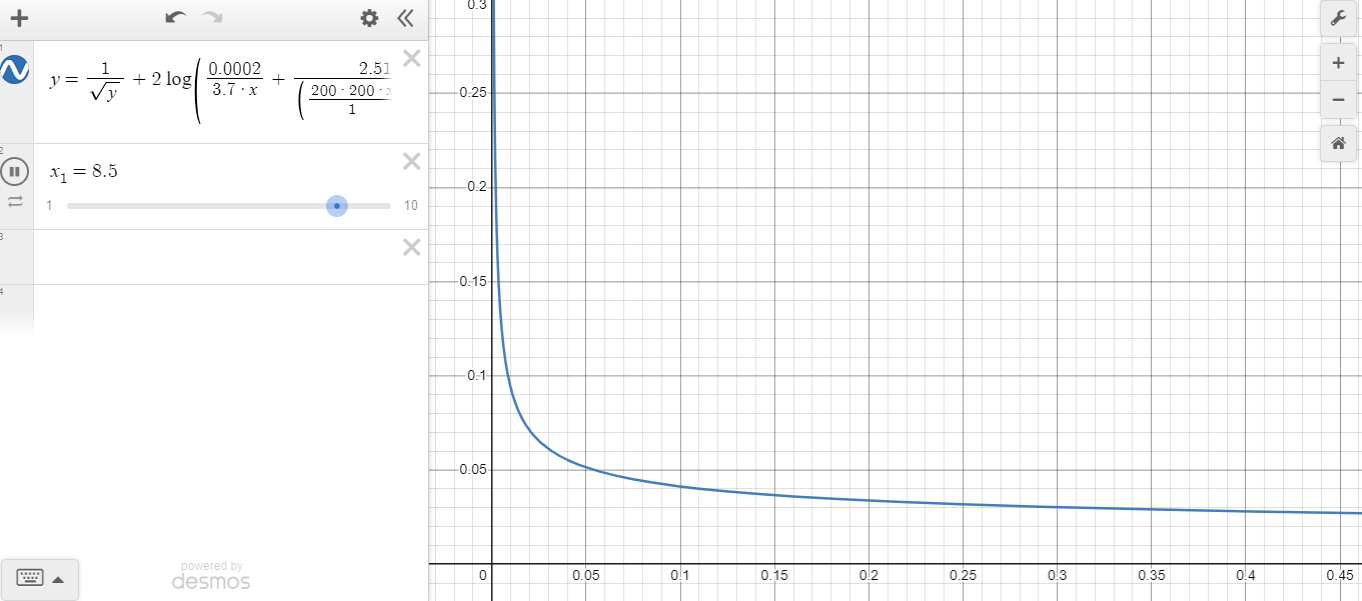
Test1: testing menu item 1 to check if it prints the numerical and analytical values for the friction factor correctly

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| density Kg/m^3 | Velocity/ m/s | Viscosity/ N s/m^2 | Diameter/m | Roughness/mm | Friction factor analytical | Friction factor numerical |
| 200 | 200 | 1 | 10 | 0.0002 | 0.0139006 | 0.0139 |

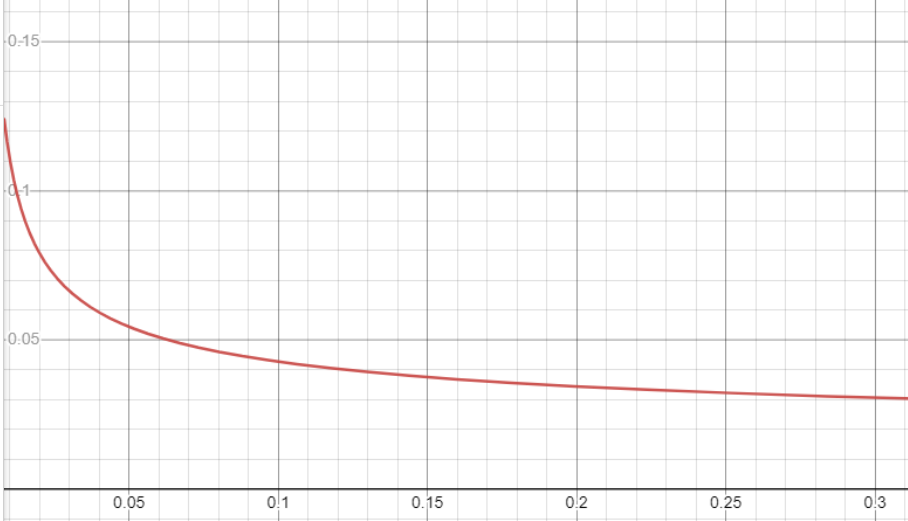
Test2: testing the plotting of the friction factor vs the conduit diameter analytically and numerically

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Density/Kg/m^3 | Velocity/m/s | Viscosity/N s/m^2 | diameter/m  start | diameter/m  end | roughness/mm |
| 200 | 200 | 1 | 1 | 10 | 0.0002 |

Graph of Friction factor vs conduit diameter numerically



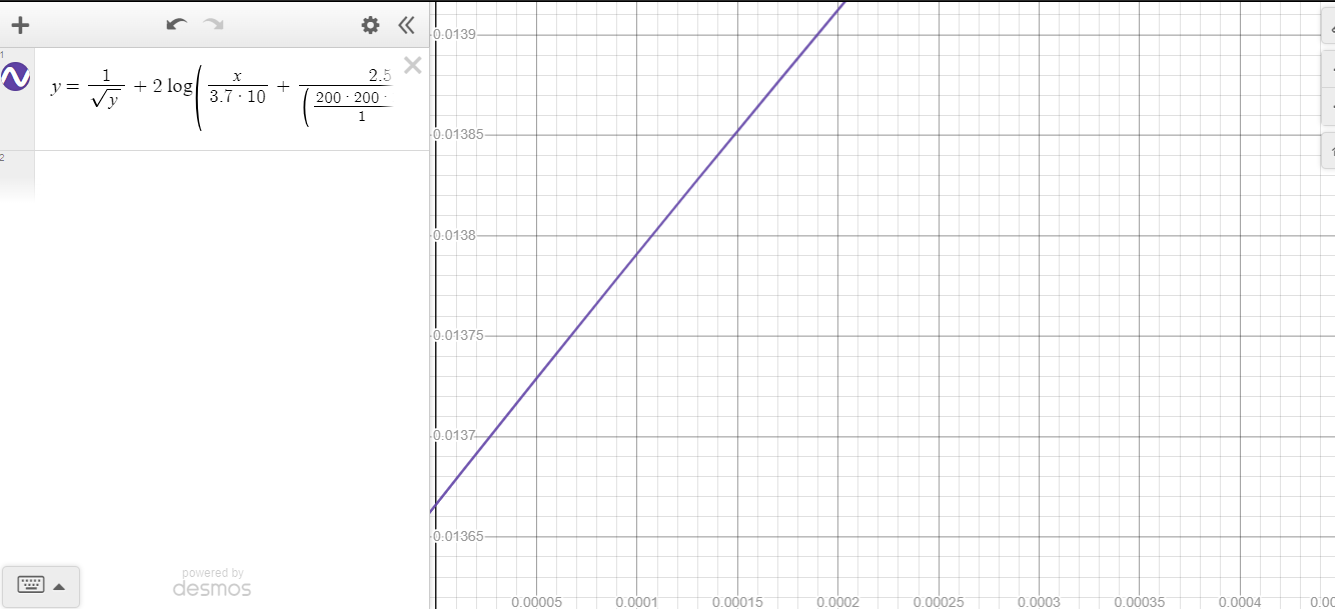
Graph of Friction factor vs conduit diameter analytically



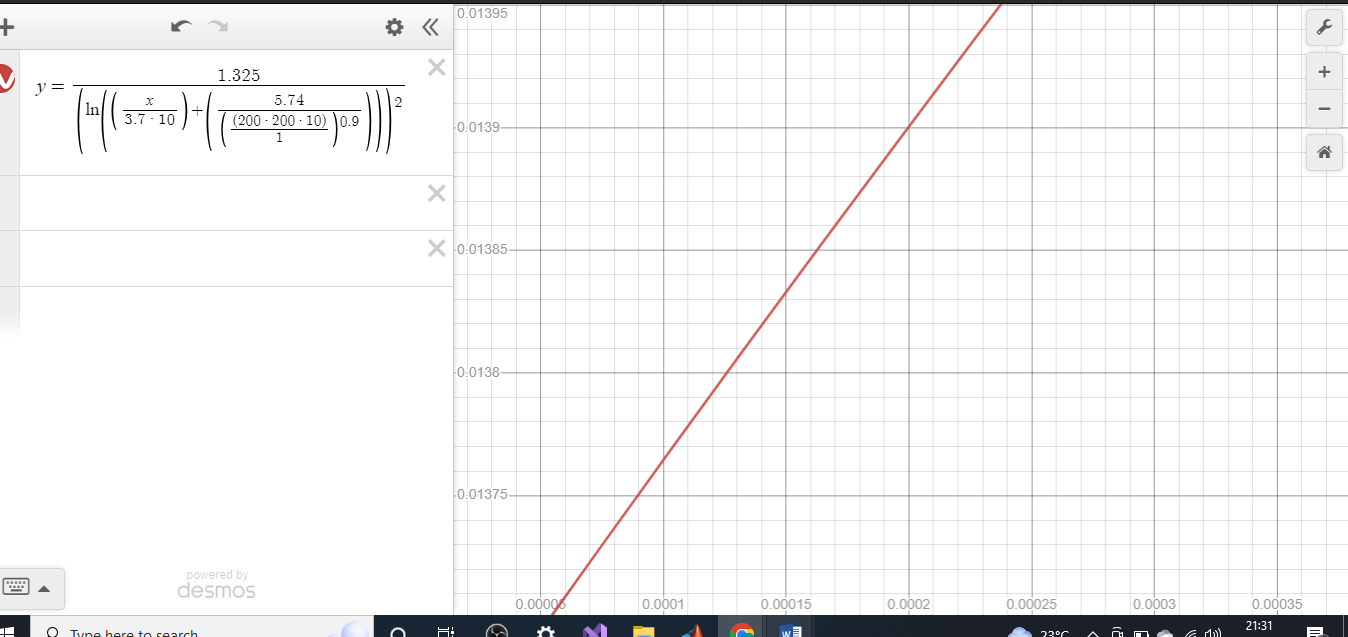
Test3: testing friction factor vs pipe roughness

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Density/Kg/m^3 | Velocity/m/s | Viscosity/N s/m^2 | Roughness/mm  start | Roughness/mm  end | Diameter/m |
| 200 | 200 | 1 | 1 | 2 | 10 |

Graph of friction factor vs pipe roughness numerically



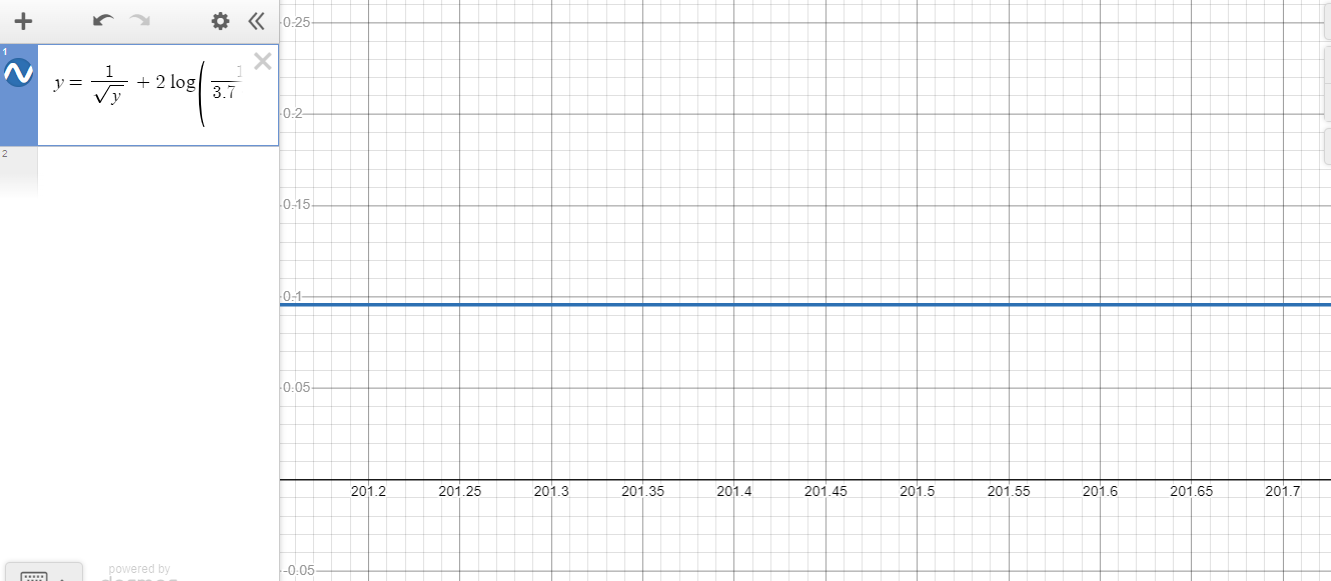
Graph of friction factor vs pipe roughness analytically



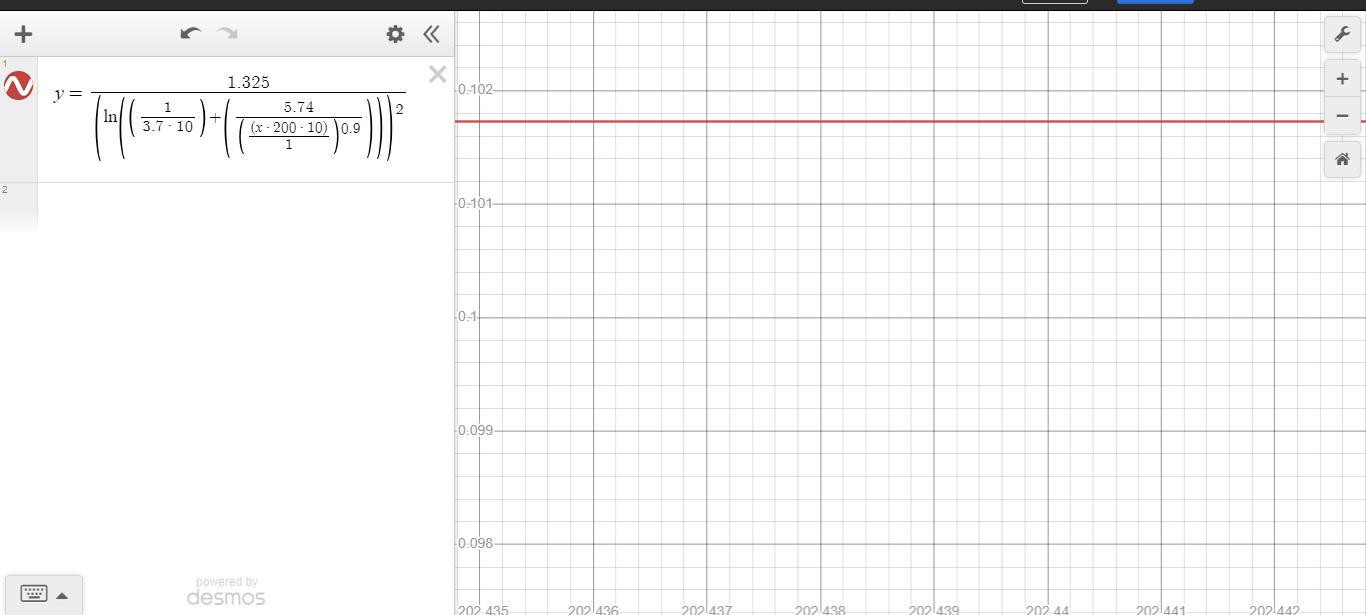
Test4: testing graph of fluid density vs friction factor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Roughness/mm | Velocity/m/s | Viscosity/N s/m^2 | Density/Kg/m^3  start | Density/Kg/m^3  end | Diameter/m |
| 1 | 200 | 1 | 200 | 300 | 10 |

Graph of friction factor vs fluid density calculated numerically



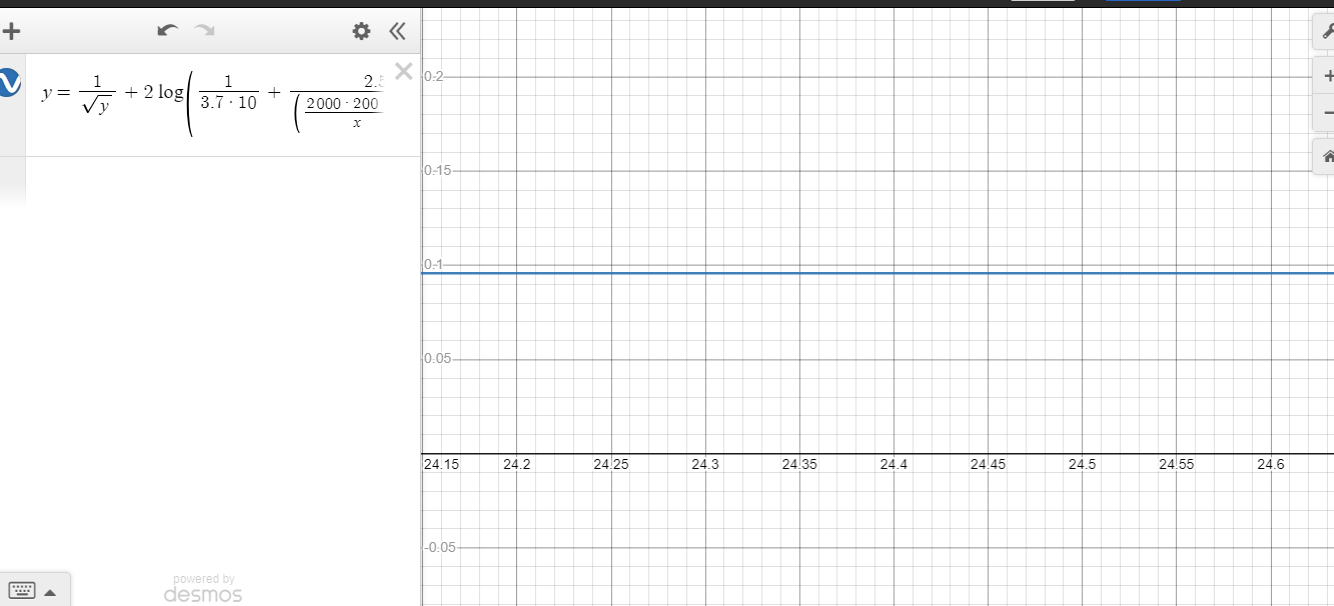
Graph of friction factor vs fluid density analytically



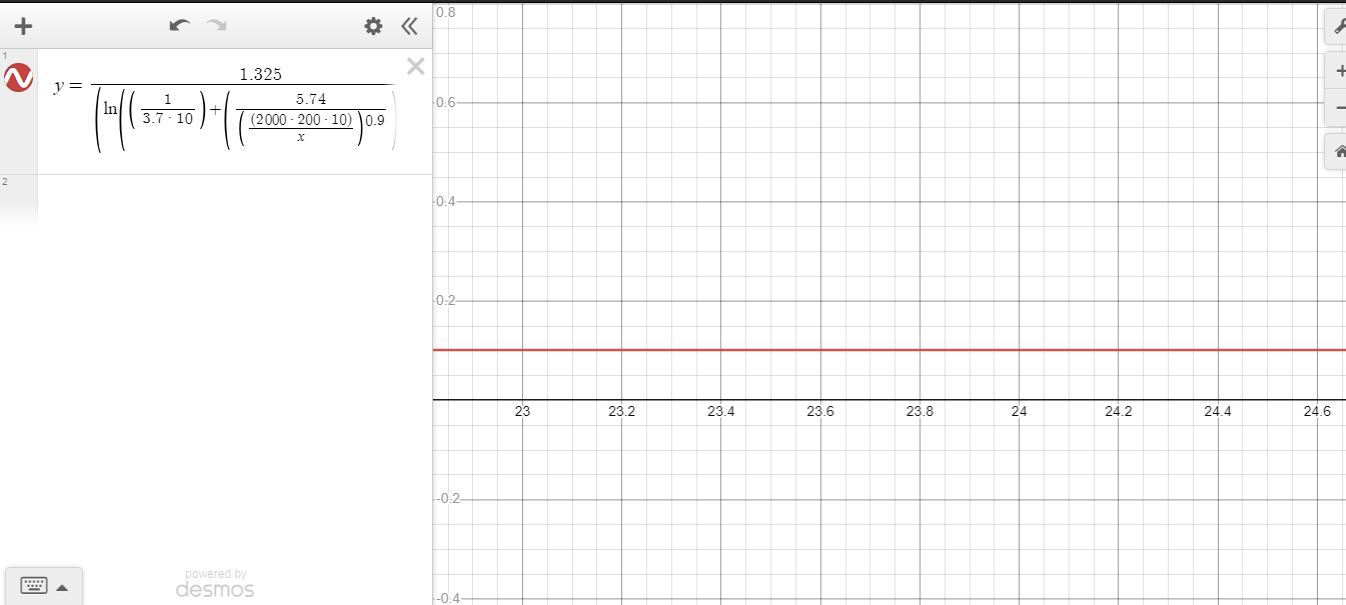
Test5: Testing graph of friction factor vs dynamic viscosity analytically and numerically

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Roughness/mm | Velocity/m/s | Density/Kg/m^3 | viscosity /N s/m^2  start | Viscosity/ /N s/m^2  end | Diameter/m |
| 1 | 2000 | 200 | 10 | 20 | 10 |

Graph of friction factor vs dynamic viscosity numerically



Graph of friction factor vs dynamic viscosity analytically



* Test6: Testing case 6 to see that menu exits when any number other than 1, 2, 3,4, 5 is pressed
* Test7: Testing the validation of Re to see that message appears when it is < 4000

**Step 3: Algorithm**

Print “1- calculate friction factor”, new line, “2- graph friction factor vs conduit diameter”, new line ,“3- graph friction factor vs pipe roughness”, new line ,“4- graph friction factor vs fluid density”, new line, “5- graph friction factor vs dynamic viscosity”, new line ,“press other number to exit menu”

Assign true to x

While x is equal true

Print “enter choice 1, 2, 3, 4 or 5”

Input choice

If choice equals 1

Print “enter fluid density (Kg/m^3)”

Input dens

While dens<0.5 or dens>2000

Print “Incorrect value, enter fluid density (Kg/m^3)”

Input dens

Print “enter conduit diameter (m)”

Input diam

Print “enter roughness (mm)”

Input rough

While rough<0.0001 or rough>3

Print “Incorrect value, enter fluid density (Kg/m^3)”

Input rough

Call function Re with arguments dens, vel, diam, visc and assign the return to Ren

If (Ren< 4000)

Print “Inputs are incorrect, calculated Reynolds number is< 4000”

Call function f with arguments diam, Ren, rough and assign the return to fana

Call function fnum with arguments diam, Ren, rough and assign the return to fnumeric

Print “friction factor calculated analytically”, fana

Print “friction factor calculated numerically”, fnumeric

If choice equal 2

Print “enter fluid density”

Input dens

while (dens<0.5 or dens>2000)

Print “incorrect input, enter fluid density”

Input dens

Print “enter fluid velocity (m/s)”

Input vel

Print “'enter fluid dynamic viscosity (N s/m^2)”

Input visc

while (visc<0.000001 or visc>300)

Print “Incorrect input, enter fluid dynamic viscosity (N s/m^2)”

Input visc

Print “enter roughness (mm)”

Input rough

while (rough<0.0001 or rough>3)

Print “Incorrect input, enter roughness (mm)”

Input rough

Initialize step to 0.1

Print “enter starting conduit diameter (m)”

Input diam1

Print “enter ending conduit diameter (m)”

Input diam2

c = (diam2-diam1)/step

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to numericalValues

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to analytical

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to xtemp

initialize i to 1

while (i<c)

cdiam = diam1 + (step\*i)

assign cdiam to xtemp(i)

call function Re with arguments dens, vel, cdiam, visc and assign the return to Ren

If (Ren< 4000)

Print “Inputs are incorrect, calculated Reynolds number is< 4000”

Call function fnum with arguments cdiam, Ren, rough and assign the return to numericalValues(i)

Call function f with arguments cdiam, Ren, rough and assign the return to analytical(i)

increment i

call function plot with arguments xtemp, analytical to graph the function f

call function plot with arguments xtemp, analytical to graph the function fnum

if choice equal 3

Print “enter fluid density”

Input dens

while (dens<0.5 or dens>2000)

Print “incorrect input, enter fluid density”

Input dens

Print “enter conduit diameter (m)”

Input diam

Print “enter fluid velocity (m/s)”

Input vel

Print “'enter fluid dynamic viscosity (N s/m^2)”

Input visc

while (visc<0.000001 or visc>300)

Print “Incorrect input, enter fluid dynamic viscosity (N s/m^2)”

Input visc

call function Re with arguments dens, vel, cdiam, visc and assign the return to Ren

If (Ren< 4000)

Print “Inputs are incorrect, calculated Reynolds number is< 4000”

Print “enter starting roughness (mm)”

Input rough1

while (rough1<0.0001 or rough1>3)

Print “Incorrect input, enter roughness (mm)”

Input rough1

Print “enter ending roughness (mm)”

Input rough2

while (rough2<0.0001 or rough2>3)

Print “Incorrect input, enter roughness (mm)”

Input rough2

Initialize step to 0.0001

c = (rough2-rough1)/step

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to numericalValues

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to analytical

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to xtemp

initialize i to 1

while (i<c)

crough = rough1 + (step\*i)

assign crough to xtemp(i)

Call function fnum with arguments diam, Ren, crough and assign the return to numericalValues(i)

Call function f with arguments diam, Ren, crough and assign the return to analytical(i)

increment i

call function plot with arguments xtemp, analytical to graph the function f

call function plot with arguments xtemp, analytical to graph the function fnum

if choice equal 4

Print “enter conduit diameter (m)”

Input diam

Print “enter fluid velocity (m/s)”

Input vel

Print “'enter fluid dynamic viscosity (N s/m^2)”

Input visc

while (visc<0.000001 or visc>300)

Print “Incorrect input, enter fluid dynamic viscosity (N s/m^2)”

Input visc

Print “enter roughness (mm)”

Input rough

while (rough<0.0001 or rough>3)

Print “Incorrect input, enter roughness (mm)”

Input rough

Print “enter starting fluid density”

Input dens1

while (dens1<0.5 or dens1>2000)

Print “incorrect input, enter fluid density”

Input dens1

Print “enter ending fluid density”

Input dens2

while (dens2<0.5 or dens2>2000)

Print “incorrect input, enter fluid density”

Input dens2

Initialize step to 0.1

c = (dens2-dens1)/step

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to numericalValues

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to analytical

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to xtemp

initialize i to 1

while (i<c)

cdens = dens1 + (step\*i)

assign cdens to xtemp(i)

call function Re with arguments cdens, vel, diam, visc and assign the return to Ren

If (Ren< 4000)

Print “Inputs are incorrect, calculated Reynolds number is< 4000”

Call function fnum with arguments diam, Ren, rough and assign the return to numericalValues(i)

Call function f with arguments diam, Ren, rough and assign the return to analytical(i)

increment i

call function plot with arguments xtemp, analytical to graph the function f

call function plot with arguments xtemp, analytical to graph the function fnum

if choice equal 5

Print “enter fluid density”

Input dens

while (dens<0.5 or dens>2000)

Print “incorrect input, enter fluid density”

Input dens

Print “enter conduit diameter (m)”

Input diam

Print “enter fluid velocity (m/s)”

Input vel

Print “enter roughness (mm)”

Input rough

while (rough<0.0001 or rough>3)

Print “Incorrect input, enter roughness (mm)”

Input rough

Print “'enter starting fluid dynamic viscosity (N s/m^2)”

Input visc1

while (visc1<0.000001 or visc1>300)

Print “Incorrect input, enter fluid dynamic viscosity (N s/m^2)”

Input visc1

Print “'enter ending fluid dynamic viscosity (N s/m^2)”

Input visc2

while (visc2<0.000001 or visc2>300)

Print “Incorrect input, enter fluid dynamic viscosity (N s/m^2)”

Input visc2

Initialize step to 0.1

c = (visc2-visc1)/step

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to numericalValues

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to analytical

call function zeros with arguments 1,c to create a 1 by c matrix of zeros and assign it to xtemp

initialize i to 1

while (i<c)

cvisc = visc1 + (step\*i)

assign cvisc to xtemp(i)

call function Re with arguments dens, vel, diam, cvisc and assign the return to Ren

If (Ren< 4000)

Print “Inputs are incorrect, calculated Reynolds number is< 4000”

Call function fnum with arguments diam, Ren, rough and assign the return to numericalValues(i)

Call function f with arguments diam, Ren, rough and assign the return to analytical(i)

increment i

call function plot with arguments xtemp, analytical to graph the function f

call function plot with arguments xtemp, analytical to graph the function fnum

if choice is not equal 1 or 2 or 3 or 4 or 5

print “exit”

assign false to x

Function definitions:

Re:

R= (dens\*vel\*diam)/visc

Return the value of R

f:

fric= 1.325/((ln((rough/(3.7\*D))+(5.74/(Re^0.9))))^2);

return the value of fric

fnum:

initialize tolerance as 0.0008;

initialize xl as 0.0008;

initialize xu as 0.8;

xr = (xu+xl)/2;

fricu=1/( xu )+2\*log((rough/(3.7\*D))+(2.51/(Re\* xu )))

fricr =1/( xr )+2\*log((rough/(3.7\*D))+(2.51/(Re\* xr )))

fricl = 1/( xl )+2\*log((rough/(3.7\*D))+(2.51/(Re\* xl )))

if (fricl\*fricu< 0)

while( **|** fricr\*fricl **|** > tolerance)

if(fricr\*fricl<0)

assign xr to xu

fricu=1/( xu )+2\*log((rough/(3.7\*D))+(2.51/(Re\* xu)))

otherwise if (fricr\*fricu<0)

assign xr to xl

fricl =

1/( xl )+2\*log((rough/(3.7\*D))+(2.51/(Re\* xl)))

xr = (xu+xl)/2

fricr=1/( xr )+2\*log((rough/(3.7\*D))+(2.51/(Re\* xr)))

otherwise

print “Invalid range for xl and xu”

**Step 4: Code or implementation (MATLAB)**

Menu script:

%Menu

disp('1- calculate friction factor')

disp('2- graph friction factor vs conduit diameter')

disp('3- graph friction factor vs pipe roughness')

disp('4- graph friction factor vs fluid density')

disp('5- graph friction factor vs dynamic viscosity')

disp('press other number to exit menu')

x=true;

while(x==true) %to loop menu

choice = input('enter choice 1, 2, 3, 4 or 5 ');

switch choice

case 1

% determine friction factor given parameters

% display numerical and analytical

dens = input('enter fluid density (Kg/m^3) '); %density

while (dens<0.5 || dens>2000) %validating user input for dens

dens = input('Incorrect value, enter fluid density (Kg/m^3) ');

end

vel = input('enter fluid velocity (m/s) '); %velocity

visc = input('enter fluid dynamic viscosity (N s/m^2) '); %viscosity

while (visc<0.000001 || visc>300) %validating user input for visc

visc = input('Incorrect value,enter fluid dynamic viscosity (N s/m^2) ');

end

diam = input('enter conduit diameter (m) '); %diameter

rough = input('enter roughness (mm) '); %roughness

while (rough<0.0001 || rough>3) %validating user input for rough

rough = input('Incorrect value, roughness (mm) ');

end

Ren = Re(dens, vel, diam, visc); %calculating reynolds number

if(Ren< 4000) %validating Re

disp('Inputs are incorrect, calculated Reynolds number is< 4000')

end

fana= f(diam, Ren, rough); %calculating friction factor analytically

fnumeric= fnum(diam,Ren,rough); %calculating frcition factor numerically

fprintf("friction factor calculated analytically: %f \n",fana); %printing both friction factors

fprintf("friction factor calculated numerically: %f \n",fnumeric);

case 2

%friction factor vs conduit diameter

%user provides lower value L and upper value H

%plot graph both analytical and numeric on same plot

%save graph as .png image

dens = input('enter fluid density (Kg/m^3) ');

while (dens<0.5 || dens>2000) %validation

dens = input('Incorrect value, enter fluid density (Kg/m^3) ');

end

vel = input('enter fluid velocity (m/s) ');

visc = input('enter fluid dynamic viscosity (N s/m^2) ');

while (visc<0.000001 || visc>300) %validation

visc = input('Incorrect value,enter fluid dynamic viscosity (N s/m^2) ');

end

rough = input('enter roughness (mm) ');

while (rough<0.0001 || rough>3) %validation

rough = input('Incorrect value, roughness (mm) ');

end

step =0.1; %step size for graph

diam1 = input('enter starting conduit diameter (m)'); %lower bound of diameter

diam2 = input('enter ending conduit diameter (m)'); %upper bound of diameter

c=(diam2-diam1)/step; % finding the number of points that will be plotted

numericalValues = zeros(1,c); %creating an array that will store all the calculated friction factors numerically

analytical=zeros(1,c); %creating an array that will store all the calculated friction factors analytically

xtemp = zeros(1, c); %creating an array that will store all the x values

for i= 1:1:c %loop to go through each value of the diameter and calculate friction factor for each

cdiam = diam1 + (step\*i); %variable that stores the current diameter in the loop

xtemp(i) = cdiam; %assigning the current diameter to the current x value

Ren= Re(dens, vel, cdiam, visc); %calculating Re

if(Ren< 4000) %validating Re

disp('Inputs are incorrect, calculated Reynolds number is< 4000')

end

numericalValues(i)= fnum(cdiam,Ren,rough); %calculating numerically for current diameter

analytical(i)=f(cdiam,Ren,rough);% calculating analytic for current diameter

end

plot(xtemp,analytical,'-g') %plotting analytical graph

hold on

plot(xtemp,numericalValues,'-r') %plotting numeric graph

hold off

legend('analyical method','numerical method','Location','NorthEastOutside')

title('friction factor vs conduit diameter')

xlabel('conduit diameter (m)')

ylabel('friction factor')

saveas(gcf,'conduit\_diameter.png')

case 3

%friction factor vs pipe roughness

%user provides lower value L and upper value H

%plot graph both analytical and numeric on same plot

%save graph as .png image

dens = input('enter fluid density (Kg/m^3) ');

while (dens<0.5 || dens>2000) %validation

dens = input('Incorrect value, enter fluid density (Kg/m^3) ');

end

diam = input('enter conduit diameter (m)');

vel = input('enter fluid velocity (m/s) ');

visc = input('enter fluid dynamic viscosity (N s/m^2) ');

while (visc<0.000001 || visc>300) %validation

visc = input('Incorrect value,enter fluid dynamic viscosity (N s/m^2) ');

end

Ren= Re(dens, vel, diam, visc);

if(Ren< 4000) %validation

disp('Inputs are incorrect, calculated Reynolds number is< 4000')

end

step =(0.0001); %step size of x-axis

rough1 = input('enter starting roughness (mm) ');

while (rough1<(0.0001) || rough1>(3)) %validation

rough1 = input('Incorrect value, roughness (mm) ');

end

rough2 = input('enter ending roughness (mm) ');

while (rough2<(0.0001) || rough2>(3)) %validation

rough2 = input('Incorrect value, roughness (mm) ');

end

c=((rough2)-(rough1))/step; %calculating the number of points plotted

numericalValues = zeros(1,c); %making an array for all the values calculated numerically

analytical=zeros(1,c); %array for all values calculated analytically

xtemp = zeros(1, c); %array storing all roughness values

for i= 1:1:c %loop to go through all the values of roughness and calculate friction factor

crough = (rough1) + (step\*i); % variable to store current roughness

xtemp(i) = crough; %variable to store the current roughness as the current x-coordinate

numericalValues(i)= fnum(diam,Ren,crough); %calculating friction numerically for current roughness

analytical(i)=f(diam,Ren,crough); %calculating friction factor analytically fro current roughness

end

plot(xtemp,numericalValues,'-r') %plotting numerical

hold on

plot(xtemp,analytical,'-g') %plotting analytical

hold off

legend('analytical method','numerical method','Location','NorthEastOutside')

title('friction factor vs pipe roughness')

xlabel('pipe roughness (mm)')

ylabel('friction factor')

saveas(gcf,'pipe\_roughness.png')

case 4

%friction factor vs fluid density

%user provides lower value L and upper value H

%plot graph both analytical and numeric on same plot

%save graph as .png image

diam = input('enter conduit diameter (m)');

vel = input('enter fluid velocity (m/s) ');

visc = input('enter fluid dynamic viscosity (N s/m^2) ');

while (visc<0.000001 || visc>300) %validation

visc = input('Incorrect value,enter fluid dynamic viscosity (N s/m^2) ');

end

rough = input('enter roughness (mm) ');

while (rough<0.0001 || rough>3) %validation

rough = input('Incorrect value, roughness (mm) ');

end

step =0.1; %step size for x-axis

dens1 = input('enter lower value for fluid density (Kg/m^3) '); %lower value for density

while (dens1<0.5 || dens1>2000) %validation

dens1 = input('Incorrect value, enter fluid density (Kg/m^3) ');

end

dens2 = input('enter upper value for fluid density (Kg/m^3) '); %upper value for density

while (dens2<0.5 || dens2>2000) %validation

dens2 = input('Incorrect value, enter fluid density (Kg/m^3) ');

end

c=(dens2-dens1)/step; %calculating the number of points plotted

numericalValues = zeros(1,c); %making an array for all the values calculated numerically

analytical=zeros(1,c); %making an array for all the values calculated analytically

xtemp = zeros(1, c); %making an array for all the values of x-coordinate

for i= 1:1:c %loop to go through all the values of density and calculate friction factor

cdens = dens1 + (step\*i); %current density

xtemp(i) = cdens;

Ren= Re(cdens, vel, diam, visc); %calculating Re for each value of density

if(Ren< 4000) %validation

disp('Inputs are incorrect, calculated Reynolds number is< 4000')

end

numericalValues(i)= fnum(diam,Ren,rough); %calculating friction numerically for each value of density

analytical(i)=f(diam,Ren,rough); %calculating friction analytically for each value of density

end

plot(xtemp,analytical,'-g')

hold on

plot(xtemp,numericalValues,'-r')

hold off

legend('analyical method','numerical method','Location','NorthEastOutside')

title('friction factor vs fluid density')

xlabel('fluid density (Kg/m^3)')

ylabel('friction factor')

saveas(gcf,'fluid density.png')

case 5

%friction factor vs dynamic viscosity

%user provides lower value L and upper value H

%plot graph both analytical and numeric on same plot

%save graph as .png image

dens = input('enter fluid density (Kg/m^3) ');

while (dens<0.5 || dens>2000)

dens = input('Incorrect value, enter fluid density (Kg/m^3) ');

end

diam = input('enter conduit diameter (m)');

vel = input('enter fluid velocity (m/s) ');

rough = input('enter roughness (mm) ');

while (rough<0.0001 || rough>3)

rough = input('Incorrect value, roughness (mm) ');

end

visc1 = input('enter lower value for fluid dynamic viscosity (N s/m^2) '); %lower value of viscosity

while (visc1<0.000001 || visc1>300)

visc1 = input('Incorrect value,enter fluid dynamic viscosity (N s/m^2) ');

end

visc2 = input('enter upper value for fluid dynamic viscosity (N s/m^2) '); %upper value fo viscosity

while (visc2<0.000001 || visc2>300)

visc2 = input('Incorrect value,enter fluid dynamic viscosity (N s/m^2) ');

end

step =0.1;

c=(visc2-visc1)/step;

numericalValues = zeros(1,c);

analytical=zeros(1,c);

xtemp = zeros(1, c);

for i= 1:1:c %loop to go through all the values of viscosity and calculate friction factor

cvisc = visc1 + (step\*i);

xtemp(i) = cvisc;

Ren= Re(dens, vel, diam, cvisc); %calculating reyolds number for each value of viscosity

if(Ren< 4000)

disp('Inputs are incorrect, calculated Reynolds number is< 4000')

end

numericalValues(i)= fnum(diam,Ren,rough);

analytical(i)=f(diam,Ren,rough);

end

plot(xtemp,analytical,'-g')

hold on

plot(xtemp,numericalValues,'-r')

hold off

legend('analyical method','numerical method','Location','NorthEastOutside')

title('friction factor vs dynamic viscosity')

xlabel('dynamic viscosity (N s/m^2)')

ylabel('friction factor')

saveas(gcf,'dynamic\_viscosity.png')

otherwise

%exit menu

disp('exit')

x=false; %end menu loop

end

end

Re function script:

%calculating Reynolds number

function [R] = Re(dens, vel, diam, visc)

R=(dens\*vel\*diam)/visc;

end

f function script:

% calculating friction factor analytically

function [fric]= f(D, Re, rough)

fric= 1.325/((log((rough/(3.7\*D))+(5.74/(Re^0.9))))^2);

end

fnum function script:

%calculates friction factor numerically

function xr= fnum(D,Re,rough) %final value of xr is the friction factor

tolerance = 0.0008;

xl= 0.0008;

xu= 0.8;

xr = (xu+xl)/2;

fricu=1/(sqrt(xu))+2\*log10((rough/(3.7\*D))+(2.51/(Re\*sqrt(xu))));

fricr=1/(sqrt(xr))+2\*log10((rough/(3.7\*D))+(2.51/(Re\*sqrt(xr))));

fricl = 1/(sqrt(xl))+2\*log10((rough/(3.7\*D))+(2.51/(Re\*sqrt(xl))));

if(fricl\*fricu< 0) %cecking if there is a root between xl and xu

while(abs(fricr\*fricl)>tolerance) %repeating the bisection method until upper and lower bounds of xl and xu are less than tolerance

if(fricr\*fricl<0) %checking if the root lies between xr and xl

xu=xr; %xr will be the new upper bound

fricu=1/(sqrt(xu))+2\*log10((rough/(3.7\*D))+(2.51/(Re\*sqrt(xu)))); %updating value of fricu

elseif(fricr\*fricu<0) % checking if root lies between xr and xu

xl=xr; %wr will be new lower bound

fricl = 1/(sqrt(xl))+2\*log10((rough/(3.7\*D))+(2.51/(Re\*sqrt(xl)))); %update value for fricl

end

xr = (xu+xl)/2; %updating value of xr

fricr=1/(sqrt(xr))+2\*log10((rough/(3.7\*D))+(2.51/(Re\*sqrt(xr)))); %updating value of fricr

end

else

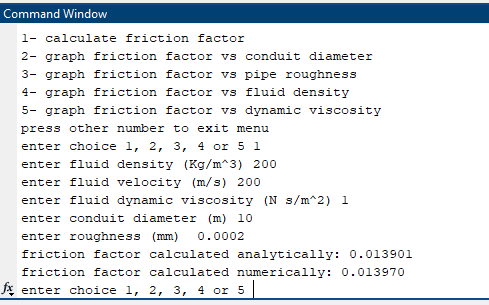
disp('Invalid range for xl and xu') %if no root lies between xl and xu display message

end

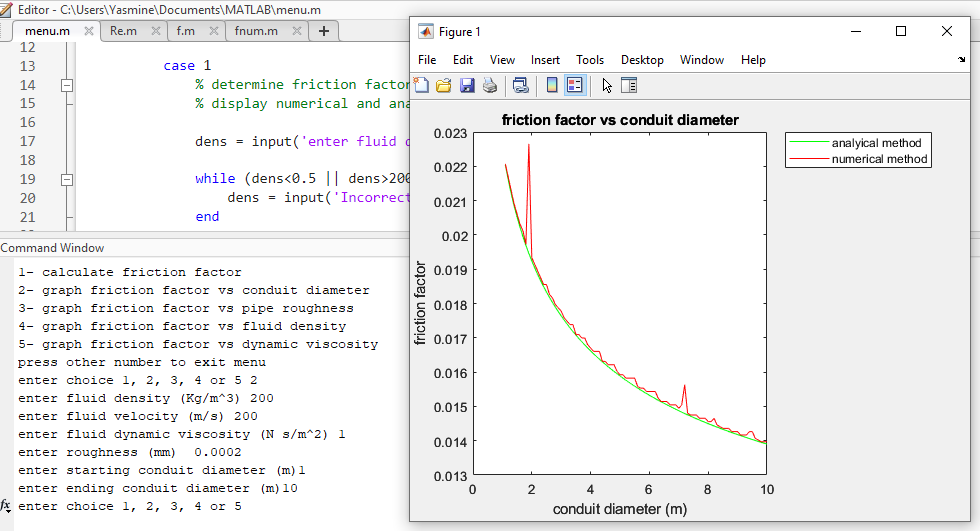
end

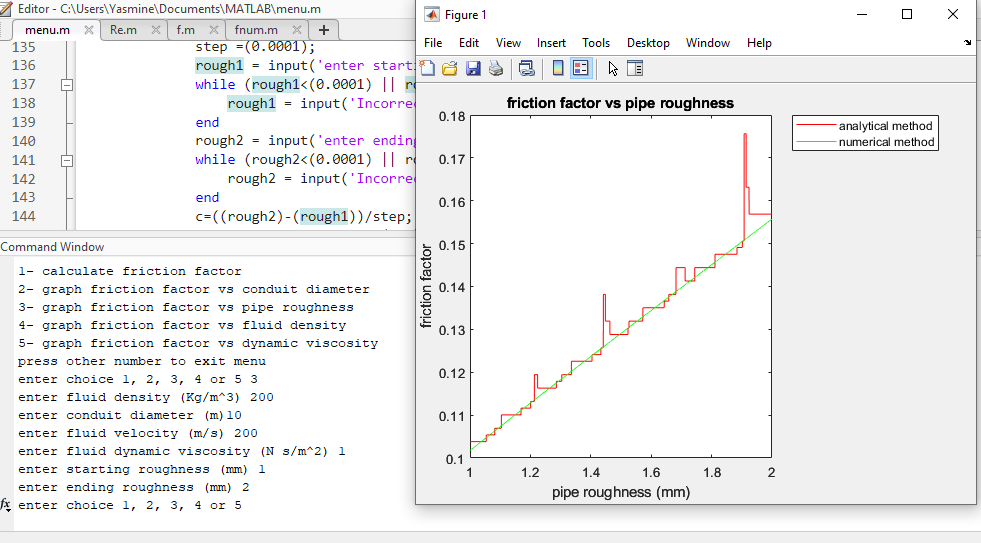
**Step 5: Test and Verification**

**Test 1:**

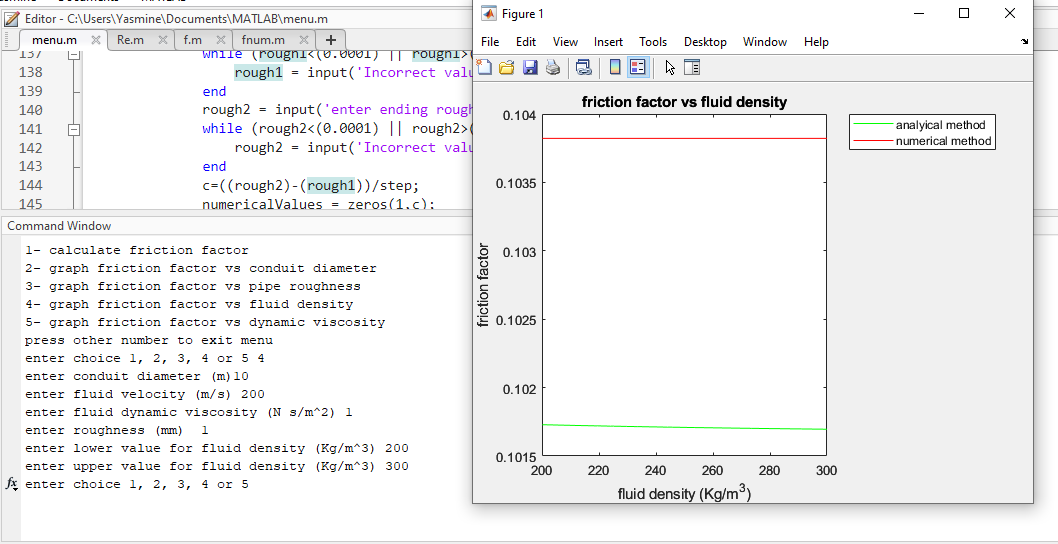


**Test 2:**

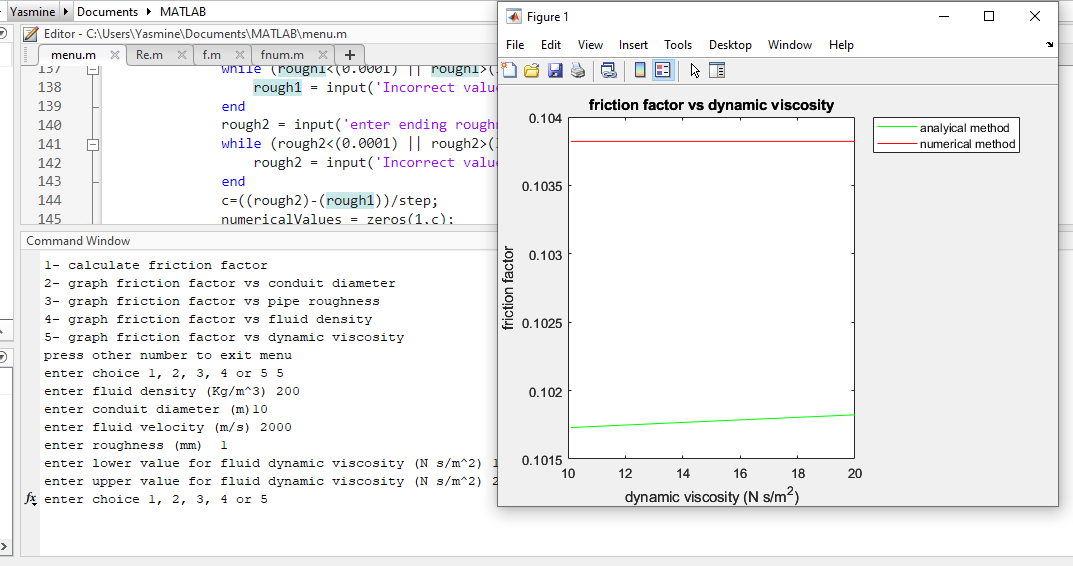


**Test 3:** 

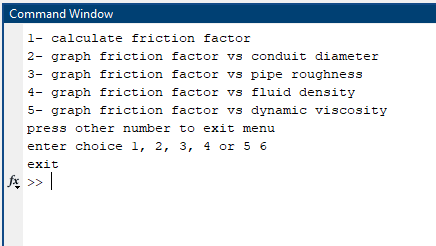
**Test4:**



**Test5:**



**Test6:**



**Test7:**

