

Coursework 1

Hassan Miah

Contents

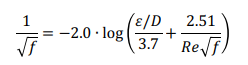
[Part a) 3](#_Toc2258816)

[Part b) 5](#_Toc2258817)

[Part c) 6](#_Toc2258818)

## Part a)

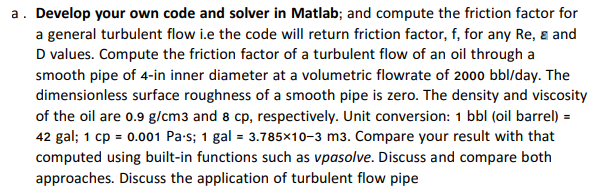
For turbulent flow inside pipes, friction factor/coefficient (ƒ) is dependent on the dimensionless surface roughness (ε/D) of the pipe inner surface and the dimensionless Reynolds number (Re), as given by the Colebrook equation below:





Where Re can be obtained from:

In the equation of Reynolds number above, Q is the volumetric flowrate of the fluid, D is the inner diameter of the pipe, A is the cross-sectional area of the pipe, and ρ and μ are the density and viscosity of the fluid, respectively.



Bisection Method

Matlab Code:

error = 1;

tol = 10e-5;

a = -5; %the lower endpoint of the [a,b] interval

b = 5; %the higher endpoint of the [a,b] interval

k = 0; %main algorithm

while error > tol % Loop ends when condition is violated.

This first paragraph of the code explains to Matlab the range for this bisection to calculate the x-values for between -5 and 5. A tolerance is set for the error to be less than 10^-5 therefore the error is very small. If error is greater than tolerance the loop will stop iterating. K = 0 is the main equation for this loop to iterate?

p = 0.9\*1000; % density (kg/m3)

u = 8\*0.001; % viscosity (Pa.s)

D = 4\*0.0254; % Inner Diameter of pipe (m)

Ar = (pi\*D^2)/4; % Area of the pipe (m2)

Q = (2000\*42\*(3.785\*10^-3))/(24\*60\*60); % Volumetric flowrate (m3/s)

e = 0; % Dimensionless surface roughness of pipe

The variables above is given in the question, however conversion is required for output.

Re = (p\*Q\*D)/(u\*Ar); % Reynold's Number

x = (a + b)/2; % Calculates the midpoint of a and b.

f\_x = -2\*log10(((e/D)/3.7)+(2.51/(Re\*x^0.5)))-(1/x^0.5) % Colebrook equation

Reynold’s number is calculated from the input variables that was calculated from paragraph 2 code. An equation for x is set for bisection to find midpoint of range between -5 and 5 set previously. The Colebrook equation given in question is rearranged to make it equal 0 on one side. The code will add x values in to the f\_x equation where an output is calculated.

if f\_x < 0

a = x; % Reassigns a

elseif f\_x > 0

b = x; % Reassigns b values

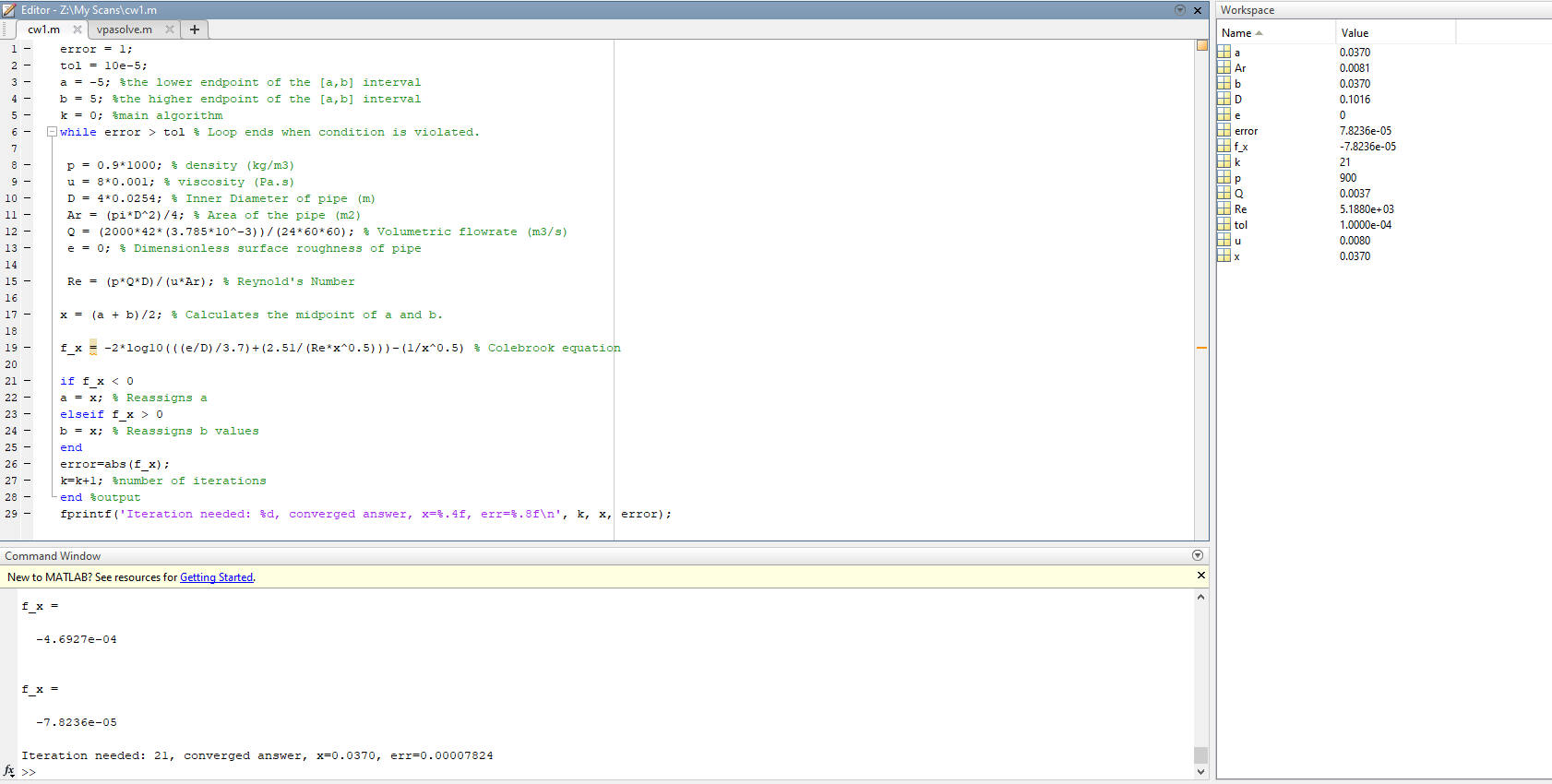
end

error=abs(f\_x);

k=k+1; %number of iterations

end %output

fprintf('Iteration needed: %d, converged answer, x=%.4f, err=%.8f\n', k, x, error);



Built in function, vpasolve:

p = 0.9\*1000;

u = 8\*0.001;

D = 4\*0.0254;

Ar = (pi\*D^2)/4;

Q = (2000\*42\*(3.785\*10^-3))/(24\*60\*60);

e = 0;

syms x

f\_x = -2\*log10(((e/D)/3.7)+(2.51/(Re\*x^0.5)))-(1/x^0.5);

x = double(vpasolve(f\_x==0,x,[0 inf]))

## Part b)

Plot on the same graph friction factor vs. Re (5000 ≤ Re ≤ 100,000) for ε/D = 0, 0.002, 0.004, 0.006, 0.008. From these plots, provide comments on the effects of volumetric flowrate and surface roughness on the friction factor in turbulent flow.

clear all;

clc;

for i=0:0.002:0.008

Re=linspace(5000,100000,1000);

error=linspace(1,1,1000);

tol=linspace(10e-10,10e-10,1000);

a=linspace(0,0,1000);

b=linspace(1,1,1000);

k=0;

while error > tol

x = (a+b)/2;

f\_x = -1./sqrt(x)-(2.\*log10((((i)/3.7)+(2.51./(Re.\*sqrt(x))))));

% Colebrook equation

for j=1:1000

if f\_x(j) < 0

a(j) = x(j);

elseif f\_x(j) > 0

b(j) = x(j);

end

error=abs(f\_x(j));

end

k=k+1;

end

disp(x)

hold on

plot(Re,x)

grid on

title ('friction')

xlabel('Re')

ylabel('friction factor')

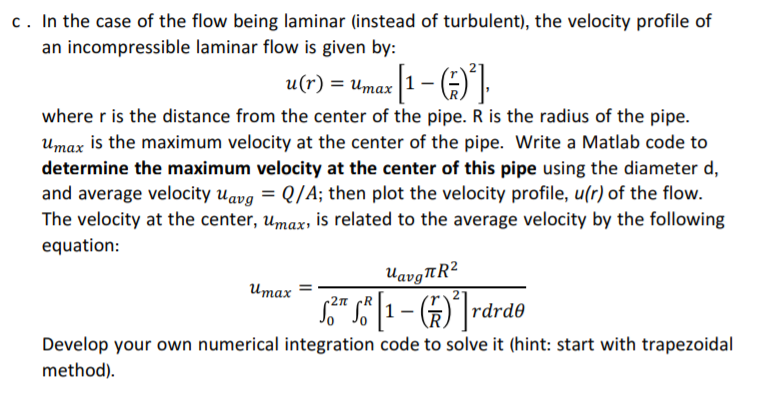
legend ('0', '0.002', '0.004', '0.006', '0.008');

fprintf('Iteration needed: %d, converged answer, x=%.4f, err=%.8f\n', k, x, error);

hold off

end

## Part c)



D = 4\*0.0254; % Inner Diameter of pipe (m)

R = D/2;

Ar = (pi\*D^2)/4; % Area of the pipe (m2)

Q = (2000\*42\*(3.785\*10^-3))/(24\*60\*60); % Volumetric flowrate (m3/s)

These Variables are constant in the maximum velocity (u max) equation. This equation includes a double integral which needs to be dealt with to solve for u max.

clear all;

clc;

a = 0;

b = R;

n = 100;

x = linspace(a, b, n);

f = x.\*((1-((x)/R).^2));

I\_trap = trapz(x, f);

The first integration is computed with the trapezium method. The first integration takes place between a distance which is from the centre of pipe diameter (0) to the wall of pipe at length R hence the limits given for a and b.

uavg = Q/Ar;

umax = ((((uavg)\*pi\*(R)^2)/ (I\_trap\*2\*pi)));

velocity = umax\*(1-(x/R).^2);

plot(x, velocity)