Anuradha Ramachandran
1/11/2019

CHBE 553- Assignment 1

Pressure loss and friction factor:

Head loss, due to friction for fluids traveling through pipes and tubes is a critical parameter for solving fluid-flow problems. The major head loss or pressure loss due to friction can be related to the length of pipe and the average velocity of the fluid by Darcy-Weisbach equation.

For a fully developed, steady incompressible single-phase flow,

$$\Delta p = f \frac{\rho V^2 L}{2D} \qquad (1)$$

Where, Δp = pressure loss,

f = friction factor,

L = length of the pipe

D = inner diameter of the pipe

V = average velocity of fluid flow

ρ = density of fluid

With the exception of the Darcy friction factor, each of the terms in (1) can be easily measured. The Darcy friction factor takes the fluid properties of density and viscosity into account, along with the pipe roughness.

For a smooth pipe, the roughness term is neglected and the magnitude of the friction factor is determined by Reynolds number alone. The Blasius equation is the most simple equation for solving the Darcy friction factor. Because the Blasius equation has no term for pipe roughness, it is valid only to smooth pipes.

Blasius equation for smooth pipes, valid for Re < 10^5 [1],

$$f = \frac{0.316}{\mathrm{Re}^{0.25}} \qquad (2)$$

Where, $\mathrm{Re} = \frac{D V \rho}{\mu}$

The friction factor for rough pipes is calculated by using the Colebrook relation.

$$\frac{1}{\sqrt{f}} = -0.86 \ln\left(\frac{\varepsilon/D}{3.7} + \frac{2.51}{Re\sqrt{f}}\right) \qquad (3)$$

Where, ε – roughness of the pipe

D – inner diameter of the pipe

Re – Reynolds number

f – friction factor

The Colebrook equation contains flow friction factor *f* in an implicit logarithmic form where it is, aside of itself; *f*, a function of the Reynolds number Re and the relative roughness of inner pipe surface ε/D; *f* =f(*f*, Re, ε/D). The most accurate way of solving the implicit form of the Colebrook equation is by using some kind of iterative methods. Newton Raphson method is one such iterative methods that can be used to solve the given Colebrook equation. The iteration needs an initial guess of Darcy friction factor. How fast the iteration is, depends on how good the initial guess is.

The explicit approximation for the Colebrook equation is given by moody's chart which establishes the trend between friction factor for turbulent flow regime from 4000 - $10^{\wedge 8}$, and ε/D ranging from $10^{\wedge -6}$ – 0.05. The MATLAB code for solving the implicit friction factor has two blocks: first, to create an array for to store Reynolds number (Re) from $10^{\wedge 4}$ – $10^{\wedge 7}$, and a function call statement and second, Newton Raphson (NR) solver block that returns the friction factor value to block 1.

From Moody's plot, it can be seen that the friction factor values lie between 0.01-0.1 [2]. Therefore, the initial guess value should be bound within the same range. In the first trial, 0.1 was assumed as the initial guess, but it can be seen that the MATLAB code returns NaN (Not a number) value for the even after 100 iterations in the NR function (Fig1).

fricfac.m | NR.m* | +

```
function xold = NR(Re, ed) %function declaration, the function returns xold=frictionfactor
x0 = 0.1; %initial guess, for other initial guess fail at higher order of Reynold's number
maxIter = 100; %iteration required for convergence
x = x0;
xold = x0;
for i = 1 : maxIter %iteration start
    f = x^-(1/2) + 0.86*log((ed / 3.7)+(2.51/Re)*(x^-(1/2))); %function declaration, colebrook equation
    df1 = (x^(1/2)*((-0.5*ed*Re)-3.99341)-4.6435)/((ed*Re*(x^2))+(9.287*(x^(3/2)))); %derivative of the function f

    x = x - f/df1; %Newton Raphson solver
    err = abs (x-xold); %Calculate the difference between the x(i) and x(i+1)
    xold = x; %to store the x(i+1) value for the next iteration
    if( err<eps) %tolerance limit, if the differnce between x(i) and x(i+1) < machine epsilon, the loop will end

        break;
    end
end
 disp(i)
end
```

Command Window

```
>> NR(10^4,10^-4)
   100


ans =

      NaN +     NaNi
```

Fig 1: Initial guess *f* = 0.1

For the second trial, the initial guess was assumed to be 0.05, it was seen that for higher orders of Re (>=10^5), this assumption fails, as the function returns NaN. To solve this, the initial guess for the friction factor was set at 0.01. The function was seen to work for higher orders of Re and the NR solver function took 6-7 iterations to converge.

To solve for f(x) = 0, NR method uses the derivative of the function (df/dx) to find the roots. According to NR method,

x(i+1) = x(i) – f(x(i))/df (x(i))

Where f(x), df is calculated at the i^{th} value of x, and the iteration is carried out until the difference between x(i+1) and x(i) is negligible. In this code, the tolerance for this function is when the difference between the old and new value of x is less than machine epsilon ($\sim 10^{\wedge-16}$, in MATLAB). If this condition returns true, the loop is terminated and the function returns the last value stored in x to block 1, and the loop continues for the next Re value. Until then, the iteration is continued. For the given relative roughness of 10^-4 and Re range of 10^4-10^7, the following plot was obtained. As the Re value increases, the friction factor is seen to decrease [Fig2].

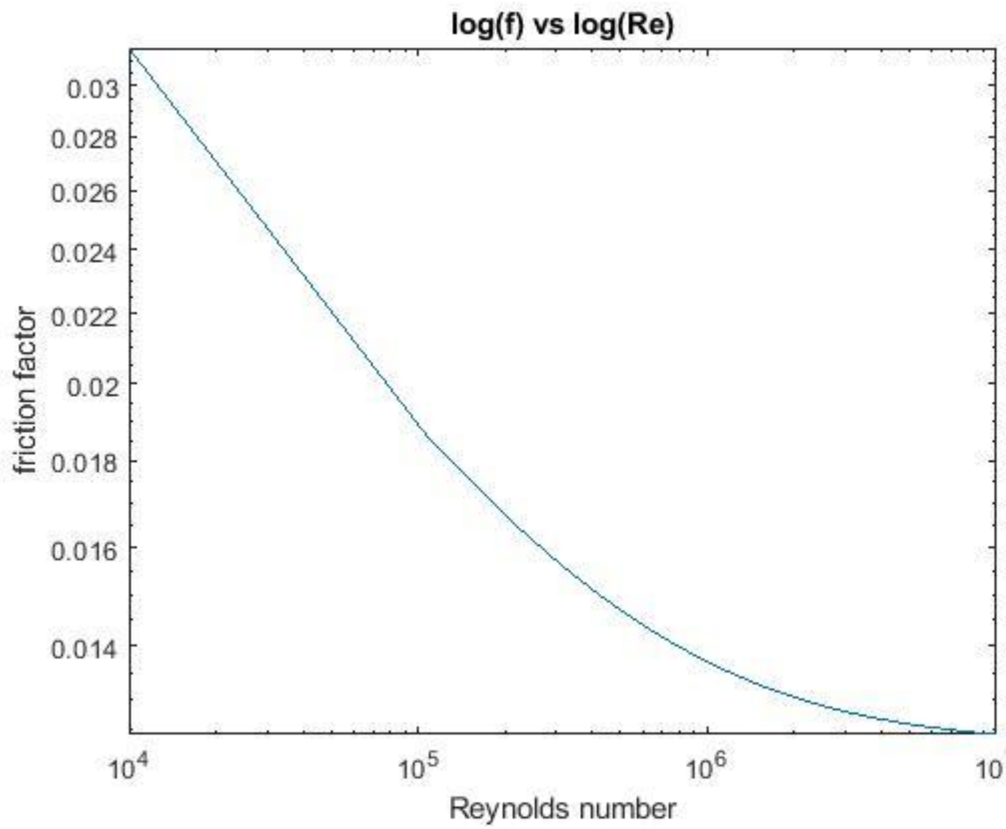
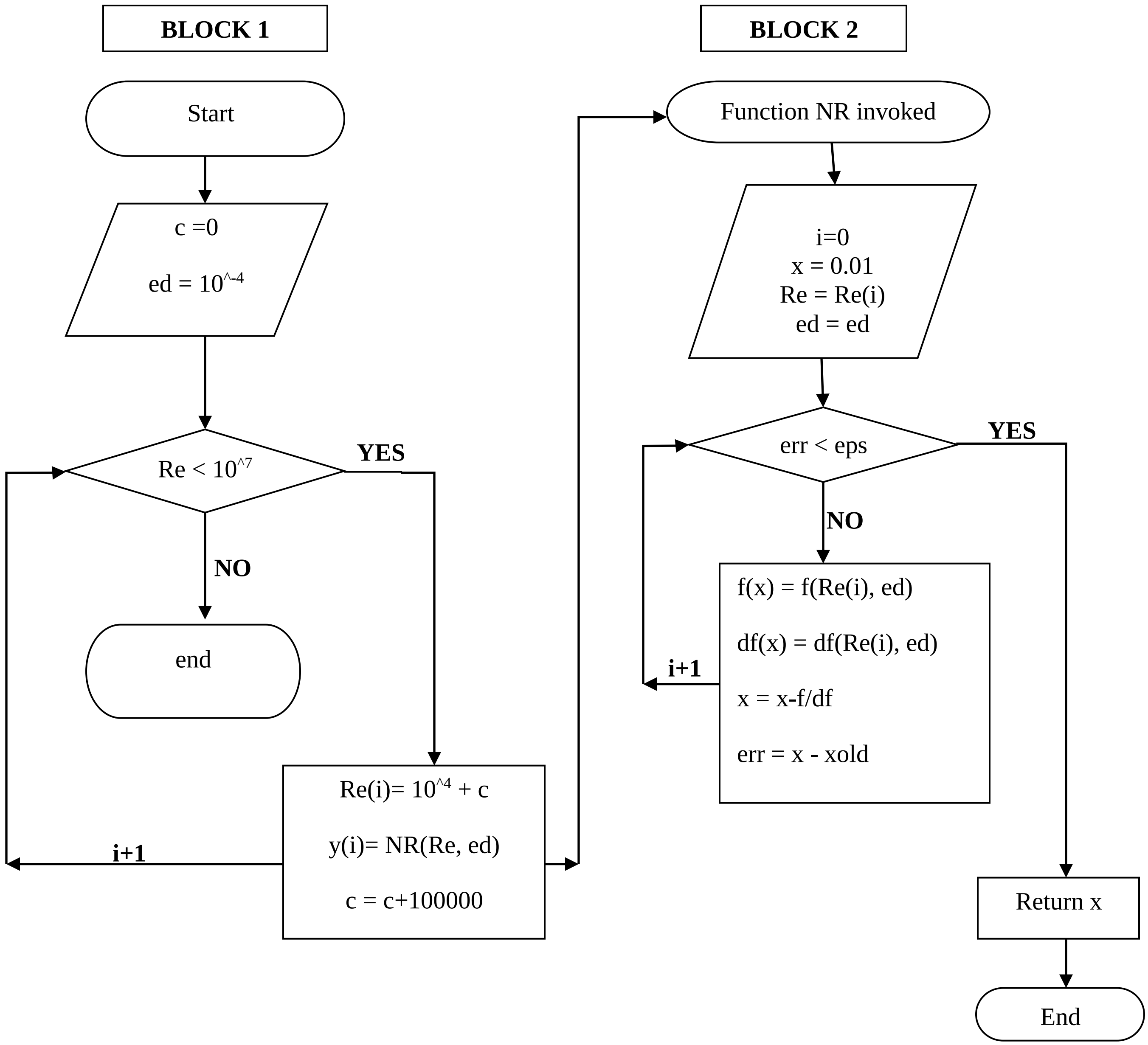


Fig 2: Loglog plot of f vs Re for $10^{\wedge4}$<Re<$10^{\wedge7}$, $\varepsilon/D = 10^{\wedge-4}$

Flow chart:



References:

1 : Jukka Kiij¨arvi Lunowa, Fluid Mechanics Paper 110727, Darcy Friction Factor Formulae in Turbulent Pipe Flow

2. https://en.wikipedia.org/wiki/Moody_chart#/media/File:Moody_EN.svg

Date accessed : Jan 2019