MATH 376: Numerical Analysis

Project 2: Pipe Friction

Quan Vu

October 10, 2017

1 Abstract

This project aims at examining the flow of fluid through pipes and tubes. The project computes the dimensionless *friction factor* in turbulent flows. By computing this factor using the bisection method, the false position method, Newton's method and lastly the fixed point iteration method, this paper draws out comparisons between how effective these methods are and what are the constraints associated with them.

2 Introduction

2.1 Background information

Determining the friction factor is of great relevance to many field of engineering and science. Some of these include the flow of liquid and gases through pipelines and cooling systems. Scientists are interested in topics ranging from the flow in blood vessels to nutrient transmission through a plant's vascular system.

2.2 Problem description

In turbulent flows, the *Colebrook equation* provides a means to calculate the friction factor using the equation

$$0 = \frac{1}{\sqrt{f}} + 2.0\log_{10}\left(\frac{\varepsilon}{3.7D} + \frac{2.51}{Re\sqrt{f}}\right)$$

where ε is the roughness (m), D is the diameter (m), and Re is the Reynold's number, as calculated by

$$Re = \frac{\rho V D}{\mu}$$

where ρ is the fluid's density (kg/m^3), V is the velocity (m/s), and μ is the dynamic viscosity $(N.s/m^2)$

2.3 Outline

By computing Reynold's number using the given values, we can substitute it back into the equation and using the numerical methods discussed, we can find the value of the friction factor. The given values are $\rho = 1.23kg/m^3$, $\mu = 1.79 \times 10^{-5} N.s/m^2$, $D = 0.005m$, $V = 40m/s$, and $\varepsilon = 0.0015mm$

3 Numerical methods

The numerical methods involved in the calculation of the friction factor are the bisection method, false position method, Newton's method, and fixed-point iteration method. Consult the attached Matlab project file to see the details of the calculations

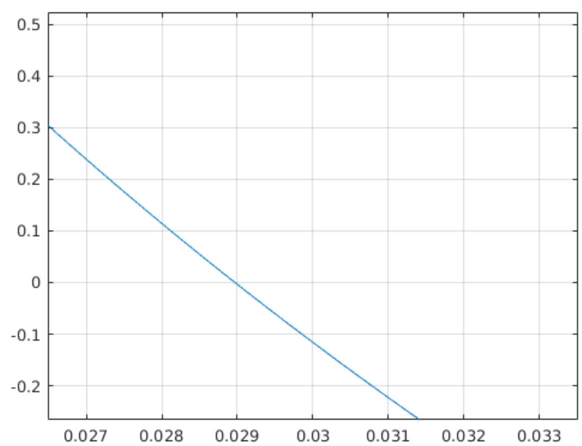
4 Results

4.1 Reynold's number

Using Matlab, we are able to determine Reynold's number $= 1.374 \times 10^4$

4.2 Plotting graph and estimation

Using Matlab to plot the graph, the friction factor seems to be somewhere around 0.029



4.3 Bisection method and False position method

For both of these methods, we approximate the value of the friction factor by using a = 0.008 and b = 0.08, with a tolerance of 10^{-8}

4.3.1 Bisection method

Took 26 iterations to calculate the answer. The method converges to 0.02896781

4.3.2 False position method

Took 31 iterations to calculate the answer. The method converges to 0.02896782

4.3.3 Discussion

We see that both method converges to the solution, however the bisection method is faster in this case.

4.4 Newton's method

We can calculate the derivative of the function quite easily:

$$h'(f) = \frac{-1}{2f\sqrt{f}} + 2.0\left(\frac{log_{10}e \times \frac{-2.51}{2Ref\sqrt{f}}}{\frac{\varepsilon}{3.7D} + \frac{2.51}{Re\sqrt{f}}}\right)$$

This is then used for the calculation for Newton's method in the Matlab file.

4.4.1 With initial guess 0.008

Took 6 iteration to calculate the answer. The method converges to 0.02896781

4.4.2 With initial guess 0.08

The approximations do not converge. They fluctuate and eventually go to infinity

4.4.3 Discussion

We see that Newton's method converges extremely fast if we choose the correct initial guess. This is due to the nature of the function around the root.

4.5 Using fzero

We use the built-in fzero function with options = optimset('Display','iter', 'TolX', 1e-8).This function searches for a point near the guess where the sign of the function changes

4.5.1 With initial guess 0.008

The method does not converge to the solution. Complex function value encountered during search

4.5.2 With initial guess 0.08

The method manages to find an interval [0.0288, 0.116204] where the function changes sign. It then continues to search for the root by evaluating possible values in this interval, and eventually got to f = 0.0289678

4.5.3 Discussion

As opposed to Newton's method, fzero was able to find the root with initial guess 0.08, but was unable to do so with initial guess 0.008.

4.6 Fixed Point Iteration

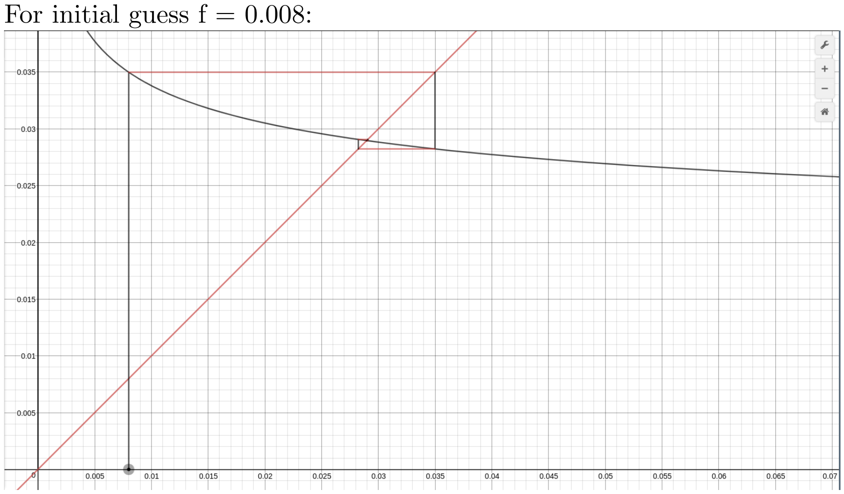
4.6.1 Iteration function

One of the possible iteration functions is:

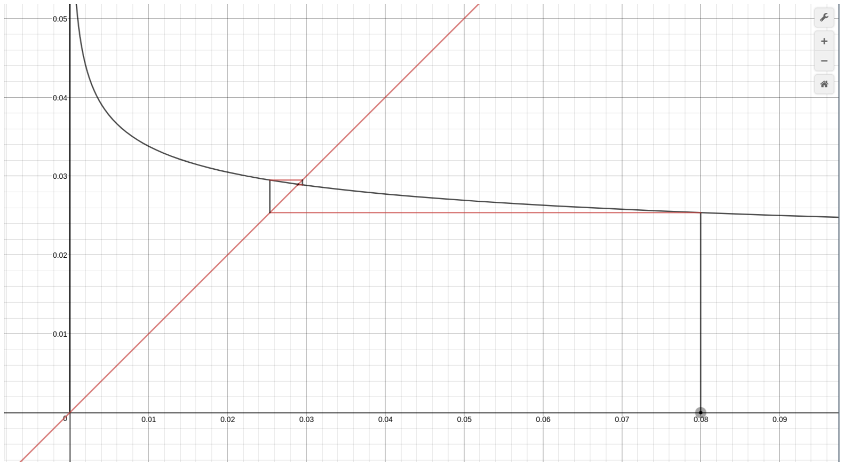
$$g(f) = \left(\frac{1}{2.0 \log_{10} \left(\frac{\varepsilon}{3.7D} + \frac{2.51}{Re\sqrt{f}} \right)} \right)^2$$

4.6.2 Cobweb Diagrams

For initial guess f = 0.008:



For initial guess f = 0.08:



4.7 Discussion

With both initial guesses at 0.008 and 0.08, the method was able to find the approximation in 9 iterations. This is because $|g'(x)| < 1$ at the root and therefore guarantees convergence.