Calculation of Friction Factor, and other parameters in different type of pipe problems

Prateek Jha, 200040106

Key Message(s):

- A simple Model to solve 3 major type of simple pipe flow problems
- Coded on python and is based on Colebrook's Equation and Darcy-Weisbach's law

Abstract

Simple Pipe flow problems is one of the sectors which hydraulic engineers are interested to study. We have various situations and cases where we have to find friction factor an head loss by assuming something and iterating it to get what we need. This Project is a simple integration of all the tedious method in a click using Darcy's law and Moody Diagram's estimation.

1 Introduction

The Darcy–Weisbach equation is an empirical equation in fluid dynamics that connects the head loss, or pressure loss, due to friction along a given length of pipe to the average fluid flow velocity for an incompressible fluid.
The Moody chart or Moody diagram is a non-dimensional graph that links the Darcy–Weisbach friction factor f, Reynolds number Re, and surface roughness in a circular pipe for fully developed flow. It can calculate the pressure drop or flow rate down a pipe.
Various moody chart estimation equations are used to calculate friction factors; some of them are Swamee and Jain, Colebrook's formula, and Nikuradse's result. Miller's estimation is used in this project because it is a good approximation in turbulent flow and is applicable for Reynolds number in the range up to 10 raised to power 8.

1.1 Friction Factor-

The friction factor depends on Reynold's number and relative roughness for rough pipes, and in the case of smooth pipes, it depends only on Reynold's Number. For a simple pipe flow, the loss in head across length is given by

$$h = \frac{\mathrm{f\,L}\,v^2}{2\,g\,D} \qquad \textbf{\textit{1}}$$

where f is the friction factor.

1.2 Problem types:

There can be three types of Problems involved in Simple pipe problems:
1. In this case, we have to find head loss across the length of the pipe; we are given Discharge, pipe's aspects, and roughness height
2. In this case, we have to find Discharge through the pipe. We are given head-loss, pipe's aspects, and roughness height
3.In this case, we have to find the Diameter, which we are given with Discharge, pipe's length, and roughness height

2 Methods and Data

We need to find Reynold's Number and Relative roughness to estimate the friction factor from Moody's Chart. The equation used to estimate the friction factor will be Miller's Estimation, that is

$$f = \frac{1.325}{(\ln\left(\frac{\epsilon}{3.7D}+\frac{5.74}{Re^{0.9}}\right))^2} \qquad \textbf{2}$$

Miller stated that if we estimate the friction factor from the above equation, which is a modified form of Colebrook's equation, the Error will be less than one percent.
Based on the type of problem, the approach will be different:

For Case 1: If we are given with Discharge, Pipe's Dimensions, i.e., Length and Diameter, Roughness height and Kinematic Viscosity and we need to find Head loss across the pipe. This is the most simple case. We first have to calculate the Reynolds number using Discharge and directly use the estimation mentioned above to get the friction factor. Now, the head loss can be calculated using Darcy Weisbach equation 1.
For Case 2: We will be given the head loss in this case, and Discharge needs to be found. This problem is approached by first assuming a very high Reynolds number (108)and then calculating the friction factor using equation 2. With the help of the calculated friction factor and other pipe aspects, we can calculate Discharge. We can find the correct Reynolds number and new friction factor with the calculated Discharge. We can repeat this step 2-3 times to get a final Discharge and Friction factor.
For Case 3: Here Diameter of the pipe will be unknown; Rest is known. For this problem, We will assume a friction factor(say 0.04) and find an unknown Diameter in terms of the assumed friction factor. Then we will calculate the Reynolds number and Relative roughness considering the calculated Diameter. We will get the friction factor using both the parameters and Equation 2. We will calculate the new Diameter and repeat the whole step from this friction factor. We will do iteration until the consecutive friction factor is almost the same.

3 Results

On running the code, First it is asked , What is the unknown in your problem and to enter the number accordingly. Then, Input of 5 parameters are taken and are used to calculate the 6^{th} parameter on the basis of unknown quantity. The algorithm to solve any problem with any particular unknown is different. It is created with if and elif statement, a simple model, which on input from user puts under one of the statements. Inside each statement is a mathematical algorithms which are derived to solve such problems.
Checked correctness of the model by testing it with some Questions on simple pipe flow problems and matching the answer, and then using same data and different missing parameters to solve all 3 types of problems. The result was satisfactory and answers displayed were matching with the given solution in all the cases.

```
Simple Pipe Flow Problems
If unknown quantity is Headloss, Enter 1
If unknown quantity is Discharge, Enter 2
If unknown quantity is Diameter, Enter 3
Enter Type of Problem: 1
Enter discharge in Litre/sec : 140
Enter Length in m: 400
Enter Diameter in mm: 200
Enter Kinematic Viscosity in m2/sec: 0.00001
Enter Roughness height in mm: 0.25
friction factor is 0.023395212158418718
Head loss is 47.40834076023774 m
```

Figure 1. Here, Type 1 problem is solved. Discharge, Length and Diameter of pipe, Kinematic Viscosity and Roughness is entered and Headloss is calculated.

```
Simple Pipe Flow Problems
If unknown quantity is Headloss, Enter 1
If unknown quantity is Discharge, Enter 2
If unknown quantity is Diameter, Enter 3
Enter Type of Problem: 2
Enter head loss in m : 47.40
Enter Length in m : 400
Enter Diameter in mm: 200
Enter Kinematic Viscosity in m2/sec: 0.00001
Enter Roughness height in mm: 0.25
friction factor is 0.0233899051391129
Speed of flow is 4.458712238840607 m/sec
Discharge is : 140.0704449831777 L/sec
```

Figure 2. Type 2 problem, Discharge is unknown. Values are same as shown in Figure 1 and it can be verified that all the values are coming same

We solved only three types of problems, but there can be more of them. Unknowns can be Roughness, height, Viscosity, or Length. for these problems, the method will be similar to case 3, i.e., finding diameter. We could also use other implicit functions like Nikuradse's result to estimate friction factors to get different results. We can also add more iterations in the code to get a more accurate result, but it will take a long time to compile.

Table 1. This table depicts two things – a) Accuracy of three methods to solve the given problem; a tutorial question of type 1 is solved using the methodology defined for it and thecross-checked by missing other parameters and using the answer of it to check if it matches with the given value in the original question. b) the second row checks the correctness of each methodology defined by verifying it with random questions and answers taken from different sources.

Type 1	**Kinematic Viscosity**	**Length**	**Roughness Height**	**Diameter**	**Discharge**	**Head Loss**	**Friction factor**
	0.00001 m^2/sec	400 m	0.25 mm	200 mm	140 L/sec	*47.40 m*	0.0233
	1.13 x 10^{-6} m^2/sec	300 m	3 mm	300 mm	124.24	*6.005 m*	0.0381
Type 2	**Kinematic Viscosity**	**Length**	**Roughness Height**	**Diameter**	**Head Loss**	**Discharge**	**Friction factor**
	0.00001 m^2/sec	400 m	0.25 mm	200 mm	47.40 m	*140.07 L/sec*	0.0233
	0.000024 m^2/sec	200 m	0.75 mm	250 mm	15 m	*172.73 L/sec*	0.0297
Type 3	**Kinematic Viscosity**	**Length**	**Roughness Height**	**Discharge**	**Head Loss**	**Diameter**	**Friction factor**
	0.00001 m^2/sec	400 m	0.25 mm	140 L/sec	47.40 m	*199.95 mm*	0.0233
	0.0001 m^2/sec	100 m	1 mm	132 L/sec	77.8 m	*149.03 mm*	0.0397

4 Conclusions

We can conclude that Miller's Estimation of Colebrook's Equation is an excellent method to solve tedious, simple pipe flow problems without using the moody chart. However, to increase accuracy, we can construct an ML model which can read the Moody chart and calculate Friction factors by intersecting the Reynolds number and Relative roughness graph. We can also go for Swamee and Jain equation to get the friction factor.

5 References

[1] Farshad, F., Rieke, H., Garber, J.: New developments in surface roughness measurements, characterization, and modeling fluid flow in pipe. J. Petrol. Sci. Eng. **29**, 139–150 (2001)

[2] Colebrook, C.F., White, C.M.: Experiments with fluid friction in roughened pipes. Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci. **161**, 367–381 (1937)

https://link.springer.com/content/pdf/bbm%3A978-4-431-54020-5%2F1.pdf

Appendix

```
import numpy as np
print("Simple Pipe Flow Problems")
print("If unknown quantity is Headloss, Enter 1" )
print("If unknown quantity is Discharge, Enter 2" )
print("If unknown quantity is Diameter, Enter 3" )
i = int(input("Enter Type of Problem: "))
if i == 1:
    q0 = float(input("Enter discharge in Litre/sec : "))
    L = float(input("Enter Length in m: "))
    D0 = float(input("Enter Diameter in mm: "))
    u = float(input("Enter Kinematic Viscosity in m2/sec: "))
    b0 = float(input("Enter Roughness height in mm: "))
    D = D0*0.001
    q = q0*0.001
    b = b0*0.001
    v = 4 * q / (3.14 * D * D)
    Rr = b / D
    R = v * D / u
    Re = pow(R, 0.9)
    f1 = ((Rr / 3.7) + (5.74 / Re))
    f2 = np.log(f1)
    f = 1.325/(f2*f2)
    h = f*L*v*v/(19.62*D)
    print("friction factor is",f)
    print("Head loss is", h,"m")
elif i ==2:
    h = float(input("Enter head loss in m : "))
    L = float(input("Enter Length in m : "))
    D0 = float(input("Enter Diameter in mm: "))
    u = float(input("Enter Kinematic Viscosity in m2/sec: "))
    b0 = float(input("Enter Roughness height in mm: "))
    D = D0*0.001
    b = b0*0.001
    Rr = b / D
    Re = pow(10,7.2)
    f0 = ((Rr / 3.7) + (5.74 / Re))
```

```
        ff = np.log(f0)
        f1 = 1.325/(ff*ff)
        o = (19.62*D*h/(f1*L))
        v1 = pow(o,0.5)
        R1 = v1*D/u
        R2 = pow(R1,0.9)
        f2 = ((Rr / 3.7) + (5.74 / R2))
        f3 = np.log(f2)
        f = 1.325/(f3*f3)
        v2 = (19.62*D*h/(f*L))
        v3 = pow(v2,0.5)
        R3 = v3*D/u
        R4 = pow(R3,0.9)
        f4 = ((Rr / 3.7) + (5.74 / R4))
        f5 = np.log(f4)
        f6 = 1.325/(f5*f5)
        v4 = (19.62*D*h/(f6*L))
        v = pow(v4,0.5)
        print("friction factor is",f6)
        print("Speed of flow is", v, "m/sec")
        Q = 3.1415*D*D*v*250
        print("Discharge is :",Q, "L/sec" )
    elif i ==3:
        h = float(input("Enter head loss in m : "))
        L = float(input("Enter Length in m: "))
        Q0 = float(input("Enter Discharge in L/sec: "))
        u = float(input("Enter Kinematic Viscosity in m2/sec: "))
        b0 = float(input("Enter Roughness height in mm: "))
        Q = Q0*0.001
        b = b0*0.001
        x = 0.04
        D0 = 0.0826*L*Q*Q*x/h
        D1 = pow(D0,0.2)
        R0 = 1.273*Q/(u*D1)
        R1 = pow(R0,0.9)
        Rr = b/D1
        f1 = ((Rr/3.7) + (5.74/R1))
        f2 = 1.325/(np.log(f1)*np.log(f1))
        D2 = (0.0826*L*Q*Q*f2/h)
        D3 = pow(D2,0.2)
        R2 = 1.273*Q/(u*D3)
        R3 = pow(R2,0.9)
        Rrr = b/D3
        f3 = ((Rrr/3.7) + (5.74/R3))
        f4 = 1.325/(np.log(f3)*np.log(f3))
        Dd = 0.0826*L*Q*Q*f4/h
        D = pow(Dd,0.2)
        R4 = 1.273*Q/(u*D)
        R5 = pow(R4,0.9)
```

```
    Rrrr = b/D
    f5 = ((Rrrr/3.7) + (5.74/R5))
    f6 = 1.325/(np.log(f5)*np.log(f5))
    Ddd = (0.0826*L*Q*Q*f6/h)
    D4 = pow(Ddd,0.2)
    print("friction factor is",f6)
    print("Diameter of pipe is:", D4*1000, "mm")
else :
    print("Try Running the code again and input 1,2 or 3")
```

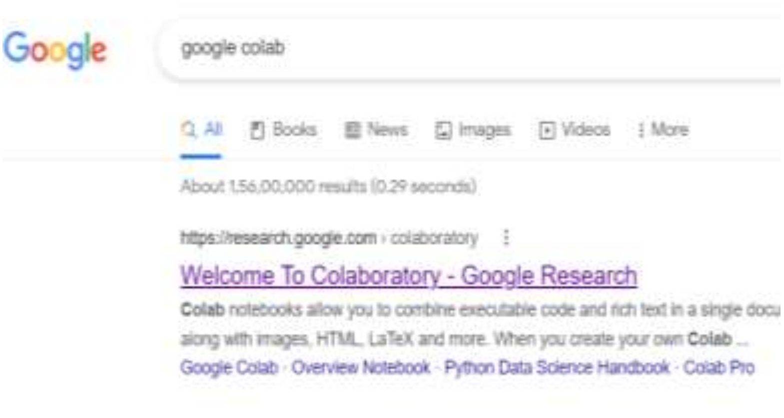
Steps- 1. Copy whole code

2. paste it on any python compiler like PyCharm

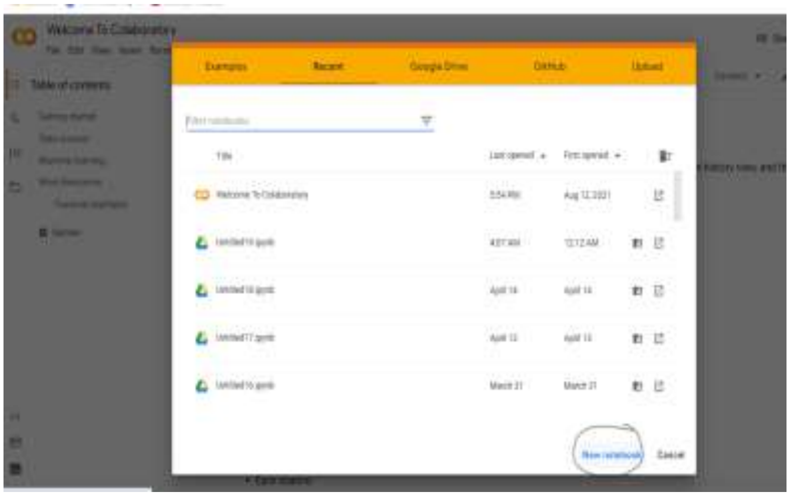
3. or paste it on any online notebook like google colab

4. Run and input values to get results

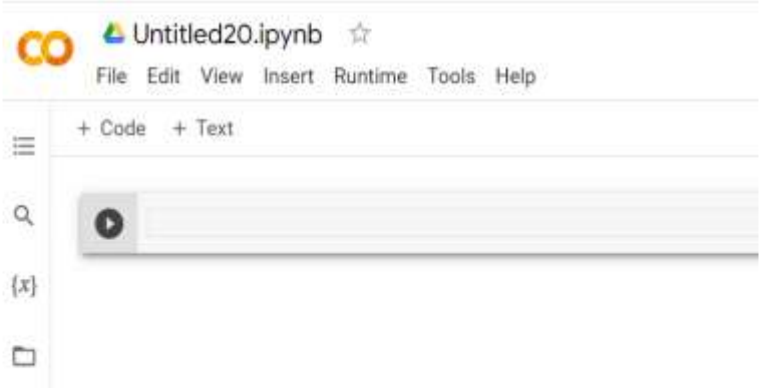
For Online Compilers-



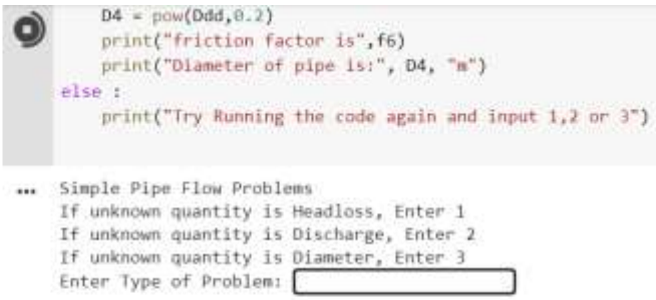
Open Google Colab



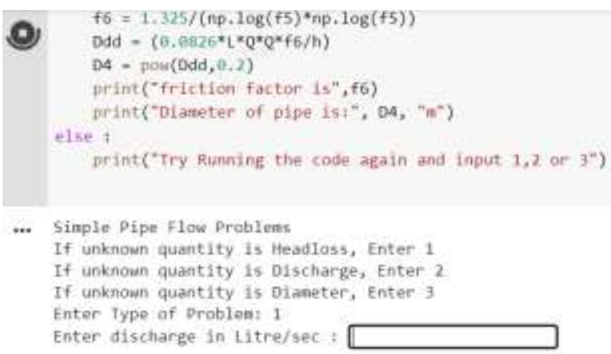
Open new notebook

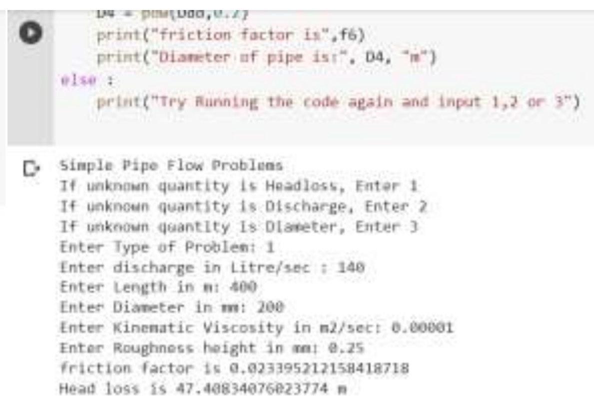


Paste the code in the block and Run



Enter type of problem





Enter all the known inputs, press enter after every input and finally we will get the uknown quantity