193_exp_3_pipe_friction

November 1, 2021

###

Exp 3 DETERMINATION OF LOSS COEFFICIENTS FOR PIPE FITTINGS (MINOR LOSES)

0.0.1 Objective:

To study the loss of head in pipe fittings at the various water flow rates.

0.0.2 Aim:

To determine the loss co-efficient for different pipe fittings (expansion, contraction, 45 degree bend and 90 degree bend). Compare your results with theoretical predictions.

0.0.3 Theory:

When there is any type of bend in pipe, the velocity of flow changes, due to which the separation of the flow from the boundary and also formation of eddies, takes place. Thus the energy is lost. The losses of head due to bend in pipe:

$$h_L = K_L \frac{V^2}{2g_c}$$

The minor losses in contraction can be expressed as:

$$h_L = K_L \frac{V^2}{2g_c}$$

The minor losses in enlargement can be expressed as:

$$h_L = K_L \frac{(V_1 - V_2)^2}{2g_c}$$

Theoretically, the loss coefficients can be estimated as follows Expansion:

$$K_e = H_L \frac{2g}{V_a^2} = \left(1 - \frac{s_a}{s_b}\right)^2$$

Contraction:

$$K_c = 0.4 \left(1 - \frac{S_a}{S_b}\right)$$

Here, $S_a$is the cross-sectional area of smaller pipe and $S_{b} $is the cross sectional area of the bigger pipe. For 45 degree and 90 degree bends, the loss coefficient is reported to be around 0.2. Compare your results with these theoretical values and comment on it.

0.0.4 Procedure:

Starting Procedure: 1. Close the drain valve provided on the sump tank. 2. Fill Sump tank ¾ with clean water and ensure that no foreign particles are there. 3. Close all Flow Control Valves given on the water line and open by-pass Valve. 4. Close all pressure taps of manometer connected to different pipe fittings. 5. Ensure that On/Off Switch given on the Panel is at OFF position. 6. Fill manometric fluid in the manometer. 7. Now switch on the main power supply. 8. Switch on the pump. 9. Operate valve to regulate the flow of water in the desired Test Section. 10. Open the pressure taps of Manometer of related test section, very slowly to avoid the blow of water on manometer fluid. 11. Now open the air release valve provided on the manometer, slowly to release the air from manometer. 12. When there is no air in the manometer, close the air release valves. 13. Adjust water flow rate in desired section with the help of control valve. 14. Record the manometer reading. 15. Measure the flow of water, discharged through desired test section, using stopwatch and measuring tank. 16. Repeat same procedure for different flow rates of water, operating control valve and by-pass valve. 17. Repeat same procedure for different test sections.

Closing Procedure: 1. When experiment is over, close all Manometers Pressure Taps first. 2. Switch off pump. 3. Switch off power supply to panel. 4. Drain all the tanks.

0.0.5 Formulae:

1. Discharge:

$$Q = \left(\frac{V}{t}\right)$$

$$\mathrm{R = (R1 - R2)/1000\ m^3}$$

2. Velocity:

$$V_a = \frac{Q}{s_a}$$

3. Loss of head

$$\mathbf{H_L} = \frac{h_L}{\mathbf{100}}\left(\frac{\rho_m}{\rho_w} - \mathbf{1}\right)$$

   m of water

4. Loss Coefficient (for Contraction):

$$K_c = H_L\frac{2g}{V_a^2} = 0.4\left(1 - \frac{S_a}{S_b}\right)$$

5. Loss Coefficient (for Expansion):

$$K_e = H_L\frac{2g}{V_a^2} = \left(1 - \frac{s_a}{s_b}\right)^2$$

6. Loss Co-efficient for pipe fittings:

$$K_f = H_L\frac{2g}{V^2}$$

0.0.6 Observations:

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Contraction

[2]:
```
R = np.array([2,2,2,2])                    #Rise in water level in hydraulic bench␣
 ↪(L)
t = np.array([18.81,20.28,24.75,33.03]) #Time (sec) required for the rise in␣
 ↪level
h = np.array([6.1,4.8,2.9,1.8])          #manometer level difference (cm)
```

[3]:
```
obs_c = pd.DataFrame({'R(L)':R,'Time(sec)':t,'h(cm of Hg)':h})
print(obs)
```

```
   R(L)  Time(sec)  h(cm of Hg)
0     2      18.81          6.1
1     2      20.28          4.8
2     2      24.75          2.9
3     2      33.03          1.8
```

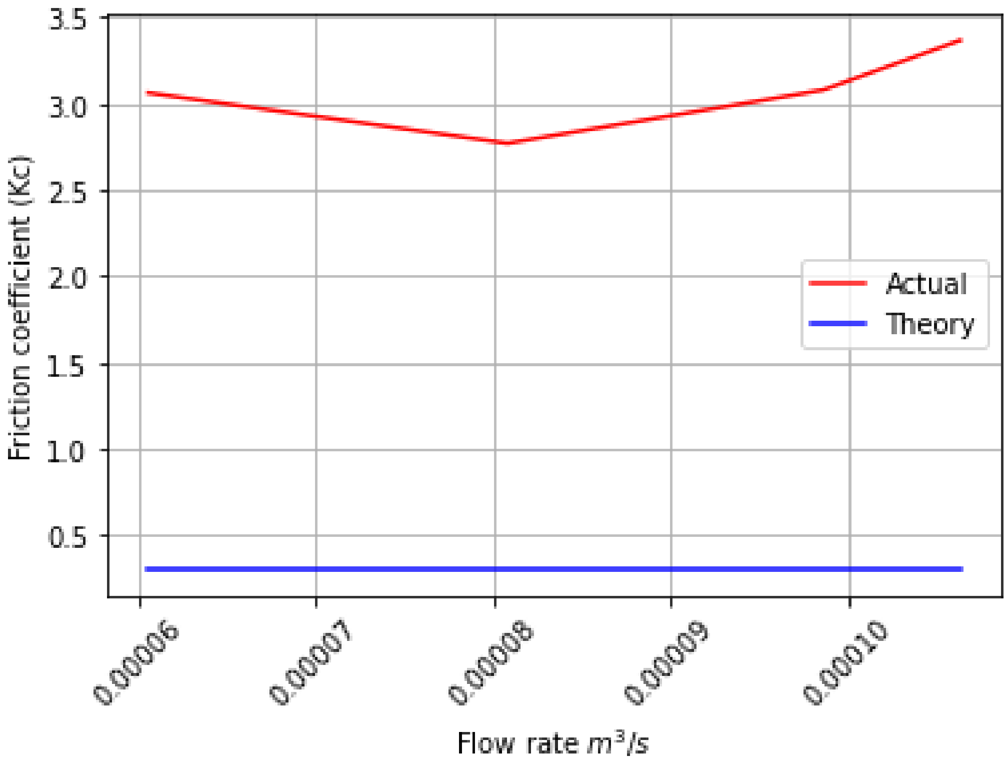
[5]:
```
D1 = 8e-3            #Diameter of the smaller pipe (m)
S1 = np.pi*D1**2/4
D2 = 17e-3           #Diameter of the bigger pipe (m) To be rechecked
S2 = np.pi*D2**2/4
Q = R*1e-3/t
V = Q/S1
H = h*12.6/100
Kc = H*2*9.8/V**2
Kct = np.ones(Kc.shape)*0.4*(1-S1/S2)
```

[6]:
```
calc_c = pd.DataFrame({'Q(m3/s)':Q,'V(m/s)':V,'HL(m)':H,'Kc':Kc})
print (calc_c)
```

```
    Q(m3/s)    V(m/s)   HL(m)        Kc
0  0.000106  2.115297  0.7686  3.366772
1  0.000099  1.961969  0.6048  3.079523
2  0.000081  1.607626  0.3654  2.771116
3  0.000061  1.204624  0.2268  3.063346
```

[7]:
```
plt.figure()
plt.plot(Q,Kc,'r',label='Actual')
plt.plot(Q,Kct,'b',label='Theory')
plt.xlabel(r'Flow rate $m^3/s$')
plt.xticks(rotation=45)
plt.ylabel('Friction coefficient (Kc)')
```

```
#plt.ylim([0,0.8])
plt.legend()
plt.grid()
```



Expansion

[8]:
```
R = np.array([2,2,2,2])                  #Rise in water level in hydraulic bench (L)
t = np.array([6.19,6.22,8.07,9.03])  #Time (sec) required for the rise in level
h = np.array([2.9,2.4,1.5,1.3])       #manometer level difference (cm)
```

[9]:
```
obs_e = pd.DataFrame({'R(L)':R,'Time(sec)':t,'h(cm of Hg)':h})
print (obs_e)
```

```
   R(L)  Time(sec)  h(cm of Hg)
0     2       6.19          2.9
1     2       6.22          2.4
2     2       8.07          1.5
3     2       9.03          1.3
```

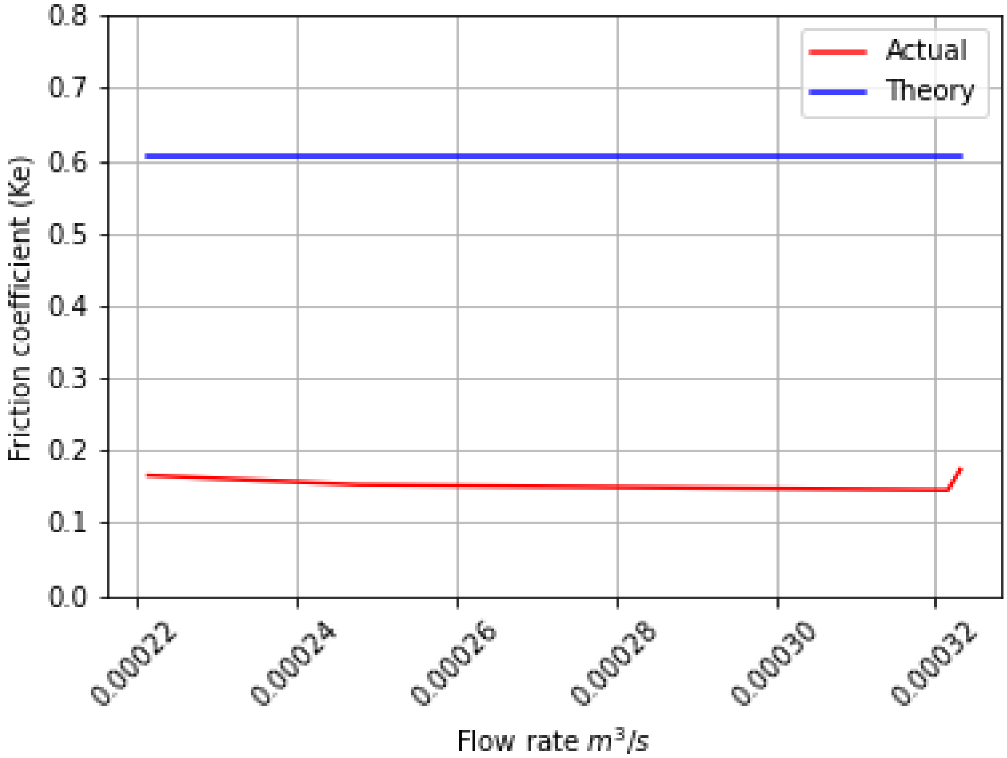
[10]:
```
D1 = 8e-3            #Diameter of the smaller pipe (m)
S1 = np.pi*D1**2/4
D2 = 17e-3          #Diameter of the bigger pipe (m) To be rechecked
```

```
S2 = np.pi*D2**2/4
Q = R*1e-3/t
V = Q/S1
H = h*12.6/100
Ke = H*2*9.8/V**2
Ket = np.ones(Ke.shape)*(1-S1/S2)**2
```

[11]:
```
calc_e = pd.DataFrame({'Q(m3/s)':Q,'V(m/s)':V,'HL(m)':H,'Ke':Ke})
print (calc_e)
```

```
    Q(m3/s)    V(m/s)   HL(m)        Ke
0  0.000323  6.427906  0.3654  0.173335
1  0.000322  6.396903  0.3024  0.144843
2  0.000248  4.930451  0.1890  0.152386
3  0.000221  4.406283  0.1638  0.165358
```

[12]:
```
plt.figure()
plt.plot(Q,Ke,'r',label='Actual')
plt.plot(Q,Ket,'b',label='Theory')
plt.xlabel(r'Flow rate $m^3/s$')
plt.xticks(rotation=45)
plt.ylabel('Friction coefficient (Ke)')
plt.ylim([0,0.8])
plt.legend()
plt.grid()
```



45 degree bend

[13]:
```
R = np.array([2,2,2,2])              #Rise in water level in hydraulic bench (L)
t = np.array([4.44,6.9,9,16.09])    #Time (sec) required for the rise in level
h = np.array([0.8,0.3,0.2,0.1])     #manometer level difference (cm)
```

[14]:
```
obs_45 = pd.DataFrame({'R(L)':R,'Time(sec)':t,'h(cm of Hg)':h})
print (obs_45)
```

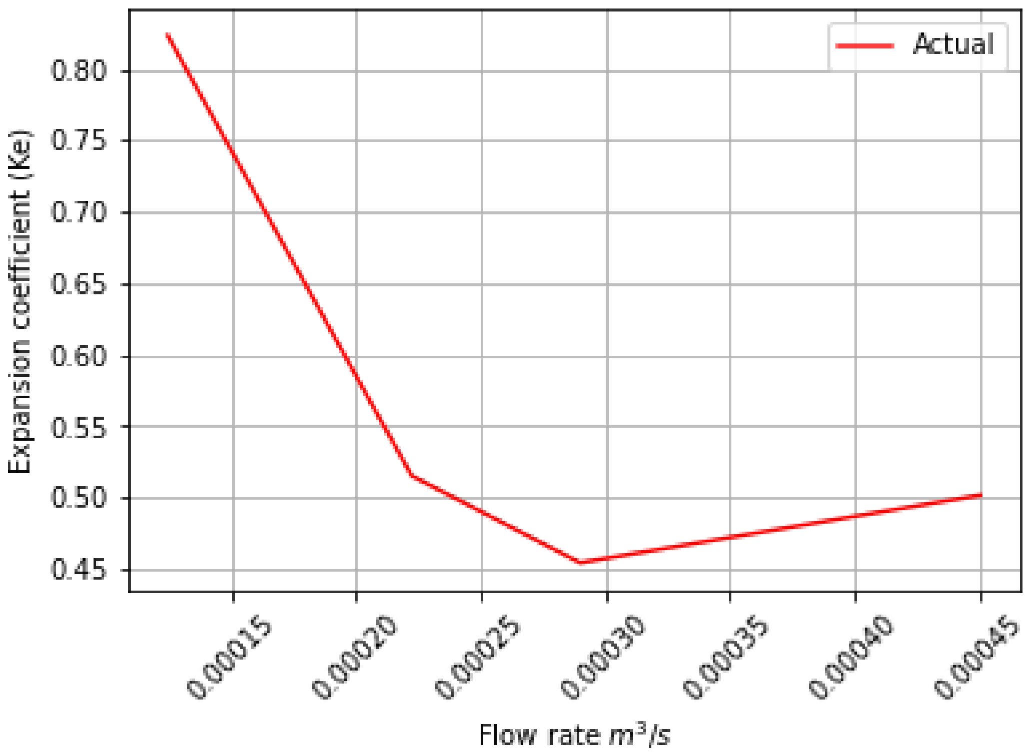
```
   R(L)  Time(sec)  h(cm of Hg)
0     2       4.44          0.8
1     2       6.90          0.3
2     2       9.00          0.2
3     2      16.09          0.1
```

[16]:
```
D2 = 17e-3 #Diameter of the bigger pipe (m) To be rechecked
S2 = np.pi*D2**2/4
Q = R*1e-3/t
V = Q/S2
H = h*12.6/100
Kf = H*2*9.8/V**2
```

[17]:
```
calc_45 = pd.DataFrame({'Q(m3/s)':Q,'V(m/s)':V,'HL(m)':H,'Kf':Kf})
print (calc_45)
```

```
    Q(m3/s)    V(m/s)   HL(m)        Kf
0  0.000450  1.984537  0.1008  0.501647
1  0.000290  1.277007  0.0378  0.454320
2  0.000222  0.979038  0.0252  0.515296
3  0.000124  0.547629  0.0126  0.823482
```

[18]:
```
plt.figure()
plt.plot(Q,Kf,'r',label='Actual')
plt.xlabel(r'Flow rate $m^3/s$')
plt.xticks(rotation=45)
plt.ylabel('Expansion coefficient (Ke)')
plt.legend()
plt.grid()
```



90 degree bend

[19]:
```
R = np.array([2,2,2,2])          #Rise in water level in hydraulic bench (L)
t = np.array([4.5,5.34,8,10.5])         #Time (sec) required for the rise in␣
 ↪level
h = np.array([1.9,1.3,0.6,0.3]) #manometer level difference (cm)
```

[20]:
```
obs_90 = pd.DataFrame({'R(L)':R,'Time(sec)':t,'h(cm of Hg)':h})
print (obs_90)
```

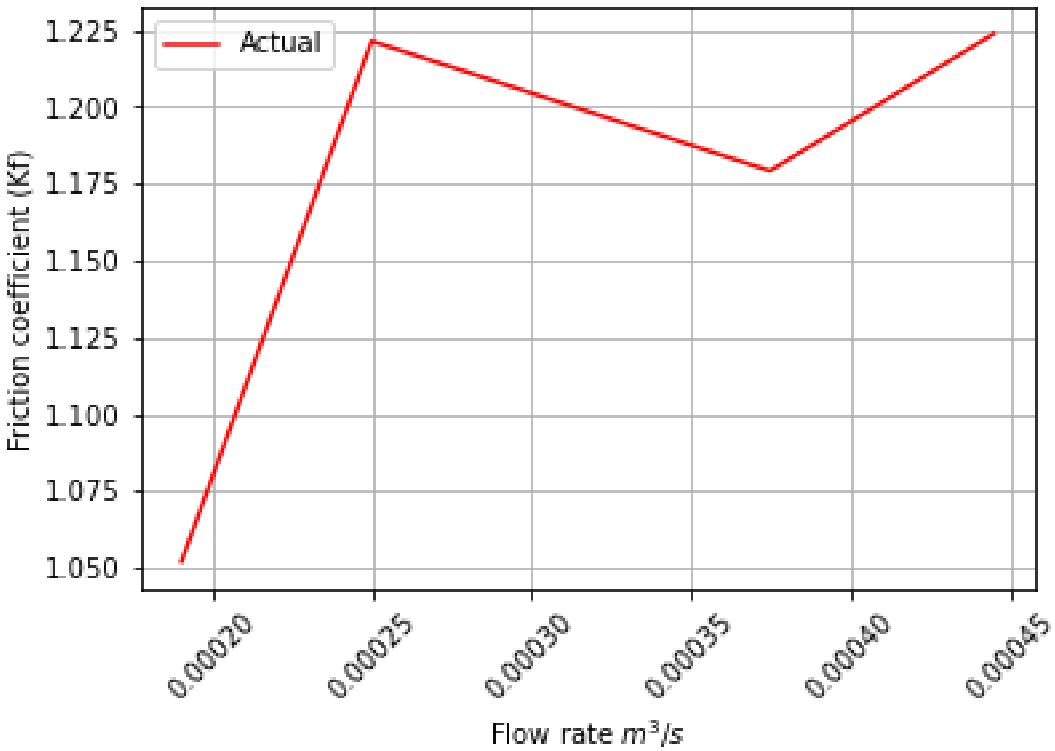
```
   R(L)  Time(sec)  h(cm of Hg)
0     2       4.50          1.9
1     2       5.34          1.3
2     2       8.00          0.6
3     2      10.50          0.3
```

[21]:
```
D2 = 17e-3          #Diameter of the bigger pipe (m) To be rechecked
S2 = np.pi*D2**2/4
Q = R*1e-3/t
V = Q/S2
H = h*12.6/100
Kf = H*2*9.8/V**2
```

[22]:
```
calc_90 = pd.DataFrame({'Q(m3/s)':Q,'V(m/s)':V,'HL(m)':H,'Kf':Kf})
print (calc_90)
```

```
    Q(m3/s)    V(m/s)   HL(m)        Kf
0  0.000444  1.958077  0.2394  1.223829
1  0.000375  1.650065  0.1638  1.179147
2  0.000250  1.101418  0.0756  1.221443
3  0.000190  0.839176  0.0378  1.052063
```

[23]:
```
plt.figure()
plt.plot(Q,Kf,'r',label='Actual')
plt.xlabel(r'Flow rate $m^3/s$')
plt.xticks(rotation=45)
plt.ylabel('Friction coefficient (Kf)')
plt.legend()
plt.grid()
```



0.0.7 Conclusions/Inferences:

1. The loss coefficients for the above cases have been determined.
2. To decrease the minor loss coefficient, and in turn the head loss, avoid sharp turns must be avoided and constricting the flow of a liquid with a contraction of the pipe.
3. Analyzing the graphs plotted, we can see that the head loss in 90 degrees bend is higher than the 45 degrees bend, whereas the head loss in sudden expansion in the pipe diameter is higher than for sudden contraction in pipe.

0.0.8 Industrial Applications:

In a pipe system with many fittings and valves, the minor losses can be greater than the major (friction) losses. Thus, an accurate K value for all fittings and valves in a pipe system is necessary to predict the actual head loss across the pipe system. K values assist engineers in totaling all of the minor losses by multiplying the sum of the K values by the velocity head to quickly determine the total head loss due to all fittings. Knowing the K value for each fitting enables engineers to use the proper fitting when designing an efficient piping system that can minimize the head loss and maximize the flow rate.