ESD - Assignment 1 - Alt658

September 23, 2024

1 Energy Systems Design - Assignment 1

1.0.1 By: Andrew Trepagnier

1.0.2 NetID: alt658

1.0.3 Date: 9/20/2024

1.0.4 Instructor: Jian Zhoa

[865]:
```
import numpy as np
import matplotlib.pyplot as plt
import sympy as sp
from IPython.display import Image
```

2 Question (1)

Using the Moody diagram to determine the following properties:

(a) For a segment of a pipe, =0.06 mm, D = 5 cm, V = 4 m/s, and =6×10 m^2/s. Find the fiction factor f and the fully rough friction factor f_T

[866]:
```
D = 5/100
V = 4
vis = 6e-6
ep = 0.06/1000


Re = D*V/vis
print("Re ", Re)

roughness = ep/D
print("roughness", roughness)
print("The fully rough friction factor is 0.025")
print("The calculated friction factor is approx. 0.035")
```

```
Re  33333.333333333336
roughness 0.0012
The fully rough friction factor is 0.025
The calculated friction factor is approx. 0.035
```

A1t658

Question 1 - Part B

f	Red	Flow Regime	ε/D
0.036	1,000,000	Fully Turbulent	0.0056
0.00155	10,000,000	Fully Turbulent	5×10^{-4}
0.036	1400	Laminar	0.01
0.08	1,000	Laminar	0.055
0.015	1,000,000	Transitionally Rough	2.5×10^{-4}
0.02	100,000	Transitionally Rough	0.001
0.036	2500	Laminar	0.0056

3 Question (2)

Question 2 AH658

	D	f_T	K	h_f
1.	½"	0.027	~~30~~ 0.81	1.62
2.	1"	0.023	0.69	1.38
3.	2"	0.019	0.57	1.14
4.	6"	0.015	0.45	0.9

$V = 2\,m/s$

$\cancel{\frac{1}{2} in \times \frac{0.0254m}{1in} = 0.0127m}$

STANDARD 90 => $\underline{K = 30 f_T}$

$$h_{f_1} = K \cdot \frac{V^2}{2g_c} = 0.81 \cdot \frac{2^2}{2 \cdot (1)} = 1.62$$

$$h_{f_2} = 0.69 \cdot \frac{2^2}{2(1)} = 1.38$$

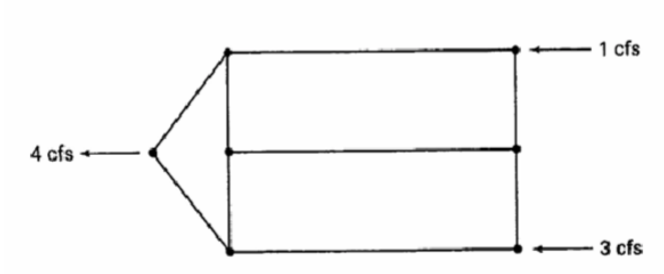
$$h_{f_3} = 0.57 \cdot \frac{4}{2} = 1.14$$

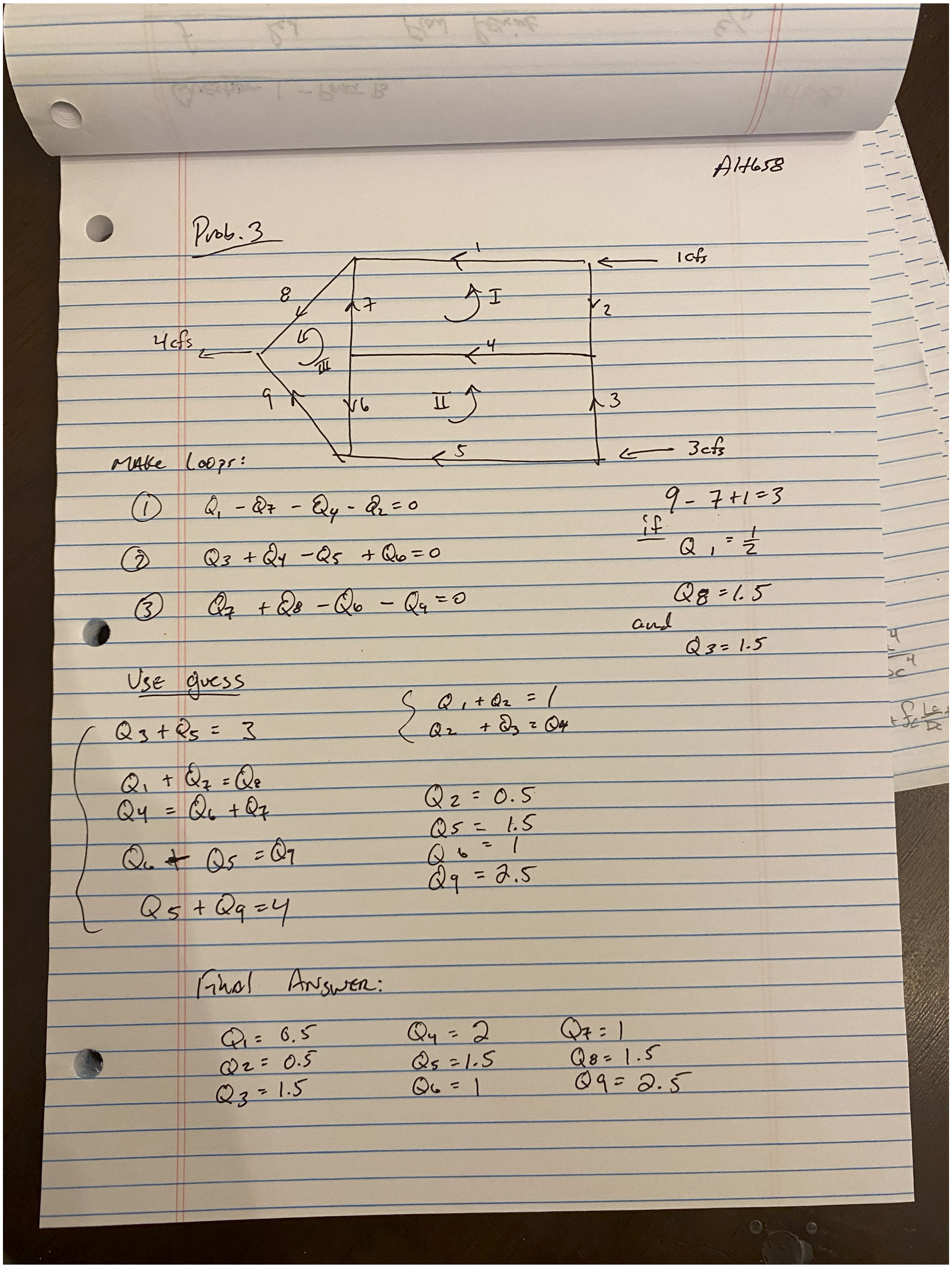
$$h_{f_4} = 0.45 \cdot 2 = 0.9$$

[ ]:

4 Question (3)

Generate a set of initial flow-rate guesses (Hardy-Cross procedure) that satisfies conservational of mass for the following network. Indicate the magnitude and direction of each flow rate.





5 Question (4)

A 14-inch (nominal) schedule 40 cast-iron pipe is used to convey 12 million gallons per day of benzene. The pipeline is 3 miles long, the pump motor is 39% efficient, and electricity costs 7.5 cents per kWh. An arrangement with a 12-inch and a 16-inch line in parallel with the existing

14-inch line is to be investigated. (a) What is the yearly pumping cost for the single pipe?

[ ]:

```

```

[867]:

```
Di = 13.124/12 #ft
rho = 55.5/32.2 #lbm/ft^3
u = 4.60e-4/32.2 # lbm/ft/s
ep = 0.00015
g = 32.2

Q_T = (12e6*0.134)/(24*3600) # ft^3/s
l = 3*5280 # ft
n = 0.39

print("Q", Q_T)
V = Q_T/(0.25*sp.pi*(Di**2))
print("V [m/s] = ", float(V))
print("Di", Di)
```

```
Q 18.61111111111111
V [m/s] =  19.811276951504023
Di 1.0936666666666668
```

[868]:

```
Rey = V*Di*rho/u
print("Rey ", float(Rey))
# based off moody diagram, Re = 2.6e6

rough = ep/Di

print("The relative roughness is ", rough)
print("Based on the moody diagram, relative roughness, reynolds number, the␣
↪friction factor f is approx. 0.0134")
f = 0.0134
```

```
Rey  2614162.5957410154
The relative roughness is  0.00013715330691862235
Based on the moody diagram, relative roughness, reynolds number, the friction
factor f is approx. 0.0134
```

[ ]:

```

```

[869]:

```
Ws = (V**2)*(f*l)/(2*g*Di)
print("Ws ", float(Ws))
```

```
Ws  1182.8075038241286
```

[870]:

```
power_to_fluid = 55.5*Ws*Q_T*(1/737.562)
print("Power to fluid [kW] =", float(power_to_fluid))
```

```
power_required = power_to_fluid*8760
print("Power to fluid yearly [kwH] =", float(power_required))

power_after_eff = power_required/0.39
cost_annually_single_pipe = power_after_eff*0.075
print("The annual cost to pump in single 14in pipe [$]",␣
 ↪cost_annually_single_pipe)
```

```
Power to fluid [kW] = 1656.4595032801844
Power to fluid yearly [kwH] = 14510585.248734416
The annual cost to pump in single 14in pipe [$] 27541103.0833253/pi**2
```

5.0.1 Part B

Since the system is parallel, the pressure drop must be the same across the nodes, therefore, Ws must be the same.

PROBLEM 4B:

12" Pipe → 0.3048 m
16" pipe → 0.4064 m

$$\frac{1}{D^2} = \frac{1}{D_1^2} + \frac{1}{D_2^2} + \frac{1}{D_3^2}$$

$$D_e = \frac{1}{0.3048^2} + \frac{1}{0.3556^2} + \frac{1}{0.4064^2} = 1.28 m^{-2}$$

$D_e = 0.880 m$

$$h_f = 0.02 \times \frac{4828}{0.880} \times \frac{5.24^2}{2(9.81)} = 12.9 m$$

$Q = 1.608e6$ ft³/day => 18.611 ft³/s
$l = 15,840 ft$
$\varepsilon = 0.000 15 ft$
$\mu = 4.60 \times 10^{-4}$ lbm/ft·s
$\rho = 55.5$ lbm/ft³

$$A_{12} = \frac{\pi}{4}(0.9965 ft)^2 = 0.78 ft^2$$

$$A_{14} = \frac{\pi}{4}(1.0937)^2 = 0.939 ft^2$$

$$A_{16} = \frac{\pi}{4}(1.250)^2 = 1.23 ft^2$$

Guess

$$\frac{A_{12}}{A_{sum}}(Q) = \frac{0.78}{2.949}(18.611) = 4.923 ft^3/s = Q_{12}$$

$$\frac{A_{14}}{A_{sum}}(Q) = \frac{0.939}{2.949}(18.611) = 5.926 ft^3/s = Q_{14}$$

$$\frac{A_{16}}{A_{sum}}(Q) = \frac{1.23}{2.949}(18.611) = 7.762 = Q_{16}$$

$V_{12} = \frac{Q_{12}}{A_{12}} = 6.312$ Problem 4B Alt658

$V_{14} = Q_{14}$ 6.311

TABLE 4B Cont   Alt 8

	Q_{12}	Q_{14}	Q_{16}	V_{12}	V_{14}	V_{16}	Re_{12}
1	4.923	5.926	7.762	6.312	6.311	6.311	758,891 →
2	4.623	5.926	8.062	5.927	↓	6.55	712,602

→	Re_{14}	Re_{16}	f_{12}	f_{14}	f_{16}	h_{12}	h_{f14}
1	832,782	951,794	0.0143	0.014	0.0137	140	125
2	↓	987,837	0.0145		0.0145	125	125

	h_{f16}
1	107
2	122

$$\text{POWER} = Q \rho h_f = (18.6)(55.5)(125 + 125 + 122) = 384242.7 \frac{\text{ft·lbf}}{s}$$

$$\text{POWER} = 384,242 \frac{\text{ft·lbf}}{s} \times \frac{1}{737.56} = 520.96 \text{ KW} \times (8760 \text{ hr})$$

$$= 4563638.18 \text{ KWh} / 0.39$$

with

$$= 11701636.4 \text{ KWh} \times \frac{\$0.075}{\text{KWh}} = \boxed{\$877,622.73}$$

5.1 Problem 5

[871]:

```
# easier to read these as sympy symbols
gamma = sp.symbols(r'\gamma')

Ws = sp.symbols('W_s')

hf = sp.symbols('h_f')

g = sp.symbols('g')

P_in = sp.symbols('P_{in}')
P_out = sp.symbols('P_{out}')

V_in = sp.symbols('V_{in}')
V_out = sp.symbols('V_{out}')

z_in = sp.symbols('z_{in}')
z_out = sp.symbols('z_{out}')


# defining my values
L_num = 7000
D_num = 60
ep_num = 0.01
h = 1250
rho_num = 1.934 #converted to slug/ft^3
mu_num = 2.043e-5 #also divided by 32.2 for conversion

n_turbine = 0.92
n_pump = 0.82

z_in = h
```

Since the pressures are at atmospheric conditions, P__in, P__out = 0 The resevoirs are big, so we can say the velocity Vin, Vout = 0 Zout = datum = 0

[872]:

```
headeq_1 = sp.Eq(Ws, g*z_in - hf)
headeq_1
```

[872]: $W_s = 1250g - h_f$

Now, hf = (K__major + K__ minor)*(V^2/2)

[873]:

```
f_laminar_sym = sp.symbols('f_{laminar}')
K_minor = sp.symbols('K_{minor}')
Q_sym = sp.symbols('Q_{sym}')
```

```python
# I calculated ft on my calculator using epsilon = 0.01 and D = 60, it is␣
↪faster just to input into python manually
Ft = 0.013259
K_minor_1 = 1.5 + 498*Ft

headeq_2 = headeq_1.subs(hf, 8*(Q_sym**2)*(f_laminar_sym*(L_num/D_num) +␣
↪K_minor_1)/(sp.pi**2*D_num**4) )
headeq_2
```

[873]:

$$W_s = -\frac{Q_{sym}^2\left(116.666666666667 f_{laminar} + 8.102982\right)}{1620000\pi^2} + 1250g$$

Equation looks strange, but it should be Ws = …. K_minor = 0.5 + 1 + 6(8)Ft + 14(30)Ft + 10(3)Ft

and Ft is now known, so we know K_minor simplifies to:

[874]:
```python
# define these symbols
Re_num = sp.symbols('Re_{num}')



#Reynolds was a function of one unknown, Q_sym

Re_num = 4*rho_num*Q_sym/(sp.pi*D_num*mu_num)



# I put the reynolds eq as a function of Q into it so now my whole head␣
↪function is just dependent on Q

f_laminar_eq =  (0.3086)*sp.log(10)**2/(sp.log(6.9/Re_num + (ep_num/(D_num*3.
↪7))**1.11))**2

# sympy doesnt have log 10 so i had to use log rules



# now we can substitute this f_laminar into my head equation and make it all a␣
↪function of Q
# everything subbed into the equation now:
```

[875]:
```python
headeq_3 = headeq_2.subs(f_laminar_sym, f_laminar_eq)

# headeq_4 = headeq_2.subs(K_minor, K_minor_1)
# headeq_4
#
# headeq_5 = headeq_4.subs(K_major, K_major_1)
# headeq_5
```

[876]:
```python
from sympy import solve
head_4 = headeq_3.subs(Q_sym, 50000)
head_4

solution = solve(head_4)
print(solution)
```

```
[{W_s: 1250.0*g - 1509.0448389156}]
```

[877]:
```python
head_4 = headeq_3.subs(Q_sym, 150000)
head_4

solution = solve(head_4)
print(solution)
```

```
[{W_s: 1250.0*g - 13580.208169733}]
```

[878]:
```python
head_4 = headeq_3.subs(Q_sym, 175000)
head_4

solution = solve(head_4)
print(solution)
```

```
[{W_s: 1250.0*g - 18484.0558748449}]
```

[879]:
```python
head_4 = headeq_3.subs(Q_sym, 250000)
head_4

solution = solve(head_4)
print(solution)
```

```
[{W_s: 1250.0*g - 37722.1354837542}]
```

Couldn't figure out how to get it to plot, so I did a few values manually:

1. 50,000 cfs => 38,741
2. 150,000 cfs => 26,670
3. 175,000 cfs =>
4. 250,000 cfs

At 150,000 cfs, I finished the rest of the calculations by hand to find the power of the fluid then multiplied that by the turbine efficiency. Essentially, I was able to make the head loss a function of Q, and iterate different values from 50,000 cfs to 250,000 cfs. Power can then be calculated using power = rho$Q$(Ws(Q)). I plotted this against flowrates.

- ws(150,000) is Ws as a function of 150,000

  Ws(150,000) * rho * 150,000 / 550 = 14,067,212

  Then 14,067,212 * 0.89 = my maximum power

  MAX Power in [hp] = 12,941,835

5.2 Part B

The max power occurs at approximately 150,000 cfs

[880]:
```
# Ws_ = sp.lambdify(Q_sym, headeq_3.rhs, modules='numpy')
```

[881]:
```
# vol_flow_rates = np.linspace(50000, 250000, 100000)
# power_of_fluid = rho*Ws_(vol_flow_rates)*vol_flow_rates/550
# power_shaft = n_turbine*power_of_fluid

# fig, ax = plt.subplots()
# ax.plot(vol_flow_rates, power_of_fluid, label='Fluid Power', ls='--')
# ax.plot(vol_flow_rates, power_shaft, label='Shaft Power')
# ax.set_xlabel(r'Flow Rate, $Q$ ($\frac{ft^3}{s}$)')
# ax.set_ylabel(r'Power ($hp$)')
# ax.set_title('Turbine')
# ax.legend()
# plt.show()
```

[882]:
```
# max_power_gen = np.max(power_of_fluid)
# print(f'The maximum power is {max_power_gen} horse power.')
```

[883]:
```
# jupyter nbconvert --to pdf my_notebook.ipynb
# jupyter nbconvert --to pdf '.\ESD - Assignment 1 - Alt658.ipynb'
```