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

[435]:
```python
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

[436]:
```python
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
```

3 Question (2)

```
[ ]:
```

4 Question (3)

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?

```
[ ]:
```

```
[437]: 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)
V = Q/(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
```

```
[438]: 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. 0.013")
f = 0.013
```

```
Rey  2614162.5957410154
The relative roughness is  0.00013715330691862235
```

Based on the moody diagram, relative roughness, reynolds number, the friction factor f is approx. 0.013

[439]:
```
Ws = (V**2)*(f*l)/(2*g*Di)
print("Ws ", float(Ws))
```

```
Ws  1147.4998171428113
```

[440]:
```
power_to_fluid = rho*Ws*Q
print("Power to fluid [lbf-ft/sec] =", float(power_to_fluid))

power_required = power_to_fluid/n
print("Power required to fluid [hp] =", float(power_required))
```

```
Power to fluid [lbf-ft/sec] = 36809.67969328453
Power required to fluid [hp] = 94383.79408534494
```

5.0.1 Scratch Work showing conversion of cost rate:

[441]:
```
# 1kw = 1.34102 hp
# 1kw = 737.56 lb-ft/s

costrate = 0.89077 #unit conversion gave $0.89352/ftlb/s - year

yearly_cost_single = power_required*costrate
print("Annual cost for the single pipe [$] ", float(yearly_cost_single))
```

```
Annual cost for the single pipe [$]  84074.25225740271
```

5.0.2 Part B

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

[442]:
```
print(float(Ws))
# This is head of 14" pipe
```

```
1147.4998171428113
```

[443]:
```
# Head for the 12" pipe
Di_2 = 11.938/12 #ft
V_2 = Q/(0.25*sp.pi*(Di_2**2))
Rey_2 = V_2*Di_2*rho/u
print("Rey ", float(Rey_2))
rough_2 = ep/Di_2
print("The relative roughness for 12in is ", rough_2)
print("Based on Moody, the friction factor is approx. 0.014")
f_2 = 0.014
```

```
Rey  2873870.8248035745
The relative roughness for 12in is  0.00015077902496230523
Based on Moody, the friction factor is approx. 0.014
```

[444]:

```python
Ws_2 = V_2*(f_2*l)/(2*g*Di_2)
print("Ws for 12in pipe ", float(Ws_2))

power_to_fluid_2 = rho*Ws_2*Q
print("Power to fluid [ft-lbf/s]  =", float(power_to_fluid_2))
```

```
Ws for 12in pipe  82.87599978593235
Power to fluid [ft-lbf/s]  = 2658.5093616631248
```

[445]:

```python
# Head for the 16" pipe
Di_3 = 15 / 12  #ft
V_3 = Q / (0.25 * sp.pi * (Di_3 ** 2))
Rey_3 = V_3 * Di_3 * rho / u
print("Rey ", float(Rey_3))
rough_3 = ep / Di_3
print("The relative roughness for 16in is ", rough_3)
print("Based on Moody, the friction factor is approx. 0.013")
f_3 = 0.013
```

```
Rey  2287217.993767006
The relative roughness for 16in is  0.00011999999999999999
Based on Moody, the friction factor is approx. 0.013
```

[446]:

```python
Ws_3 = V_3*(f_3*l)/(2*g*Di_3)
print("Ws for 12in pipe ", float(Ws_3))

power_to_fluid_3 = rho*Ws_3*Q
print("Power to fluid [ft-lbf/s]  =", float(power_to_fluid_3))
```

```
Ws for 12in pipe  38.79404308068998
Power to fluid [ft-lbf/s]  = 1244.441418181036
```

[447]:

```python
import numpy as np
from scipy.optimize import fsolve

# Define the missing functions
def Ws_2(Ws, Q):
    # Define this function based on your system's equations
    return Ws - 0.5 * Q**2  # Example equation, replace with your actual␣
 ↪equation

def Ws_func(Ws, Q):
    # Define this function based on your system's equations
    return Ws - 0.7 * Q**2  # Example equation, replace with your actual␣
 ↪equation
```

```
def Ws_3(Ws, Q):
    # Define this function based on your system's equations
    return Ws - 0.9 * Q**2  # Example equation, replace with your actual␣
 ↪equation

# Define Q_T (total flow rate)
Q_T = 10  # This is an example value, replace with your actual value

def system(x):
    Ws, Q1, Q2, Q3 = x
    return [
        Ws_2(Ws, Q1),
        Ws_func(Ws, Q2),
        Ws_3(Ws, Q3),
        Q_T - Q1 - Q2 - Q3
    ]

# Initial guess
initial_guess = np.array([100, Q_T/3, Q_T/3, Q_T/3])

# Solve the system
Ws, Q_1, Q_2, Q_3 = fsolve(system, initial_guess)

print(f'The head required to push the fluid is {Ws:.2f} feet squared per second␣
 ↪squared.')
print(f'12" pipe flow rate is {Q_1:.2f} cfs.')
print(f'14" pipe flow rate is {Q_2:.2f} cfs.')
print(f'16" pipe flow rate is {Q_3:.2f} cfs.')
print(Ws)
```

```
The head required to push the fluid is 7.45 feet squared per second squared.
12" pipe flow rate is 3.86 cfs.
14" pipe flow rate is 3.26 cfs.
16" pipe flow rate is 2.88 cfs.
7.450739428319538
```

[448]:
```
# This number seems realy low, not sure where the error is in my code
power_B = rho*Q_T*Ws/0.39



annual_cost_par = power_B*costrate
print(f'The cost per year for the parallel arrangement is {annual_cost_par}␣
 ↪dollars per year.')
```

```
The cost per year for the parallel arrangement is 293.31715353664015 dollars per
year.
```

5.1 Problem 5

[449]:
```
# 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

[450]:
```
headeq_1 = sp.Eq(Ws, g*z_in - hf)
headeq_1
```

[450]:

$W_s = 1250g - h_f$

Now, hf = (K_major + K_ minor)*(V^2/2)

[451]:
```
f_laminar_sym = sp.symbols('f_{laminar}')
K_minor = sp.symbols('K_{minor}')
Q_sym = sp.symbols('Q_{sym}')
```

```
# 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
```

[451]:

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

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:

[452]:
```
# 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:
```

[453]:
```
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
```

[454]:
```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}]
```

[455]:
```python
head_4 = headeq_3.subs(Q_sym, 150000)
head_4

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

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

[456]:
```python
head_4 = headeq_3.subs(Q_sym, 175000)
head_4

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

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

[457]:
```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 2- 150,000 cfs 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 ws(150,000) is Ws as a function of 150,000

Ws(150,000)rho150,000 / 550 = 14067212

then 14067212*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

[458]:
```python
# Ws_ = sp.lambdify(Q_sym, headeq_3.rhs, modules='numpy')
```

[459]:
```
# 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()
```

[460]:
```
# max_power_gen = np.max(power_of_fluid)
# print(f'The maximum power is {max_power_gen} horse power.')
```

[461]:
```
# jupyter nbconvert --to pdf my_notebook.ipynb
# jupyter nbconvert --to pdf '.\ESD - Assignment 1 - Alt658.ipynb'
```