Project 1

March 1, 2022

Gabe Morris

[1]:

```python
# Imports
import warnings

import numpy as np
import pandas as pd
from IPython.display import display, Latex
from msu_esd import Pipe, hardy_cross
from scipy.optimize import fsolve

warnings.filterwarnings('ignore', category=FutureWarning)  # Some random warning with pandas
```

Contents

ME 6333   Project 1   Gabe Morris
gnm54

1 Introduction

A central chiller system for the Orlando International Airport is to be investigated. The airport consists of 4 concourses (Airside 1, 2, 3, and 4) shown in Figure 1. The main terminal building is not included in the chiller system. The working fluid of this system is Therminol D-12TM, a common heat transfer fluid.

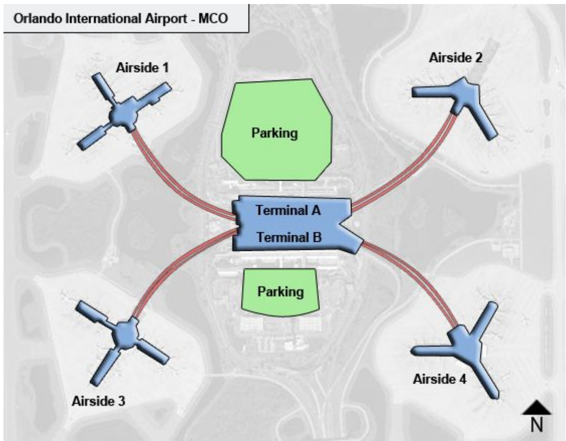

Figure 1: Concourse Layout

In addition, the following specifications must be met:

- Provide valves to isolate each concourse, pump, and supply and return lines.
- Minimize the pipe length by placing lines in the same tunnel when feasible.
- Provide 50 feet of pipe in each mechanical room (one per concourse).
- Use a "Z" network (shown in Figure 3).
- Avoid pipe velocities much in excess of 9 feet per second.
- Access tunnels are 20 feet below the entrance to each concourse, mechanical rooms are 20 feet above the surface.
- The 2 air handling units in concourse 4 are to be placed in parallel.
- Avoid running lines under the main terminal building.
- Flow rates across the air handling units must not exceed 3.5% of the minimum required.

2 Given

The energy requirements across the air handling units are,

Table 1: AHU Energy Requirements

Concourse	Tons	K_{AHU}
1	480	4.5
2	480	4.5
3	480	4.5
4 (each)	225	10

where 1 ton is $12,000\,\frac{Btu}{hr}$. The head loss through the air handling units in each terminal is $K_{AHU}Q^2$ where Q is the flow rate in cubic feet per second. The units for K_{AHU} are $\frac{ft\,lbf}{lbm}$. The head loss across the chiller is taken to be $0.1Q^2$. The Therminol D-12 exits the chiller $25^{\circ}F$ cooler than it enters. The lines are well insulated.

The fluid properties are to be taken at $50^{\circ}F$ because this is the average working temperature of therminol. The temperature is also assumed to be constant. The following properties were found:

$\rho = 48.05\frac{lbm}{ft^3}$
$c_p = 0.491\frac{Btu}{lbm\,^{\circ}F}$
$\mu = 3.715\frac{lbm}{ft\,hr}$

The piping material used for this system will be galvanized steel, a common piping material for chiller systems. The absolute roughness of galvanized steel is 0.0005’. Galvanized steel is also useful for combating corrosion.

3 Solution

The first step is to obtain a physical mapping of the airport. A schematic of the airport was given to determine a rough idea of the lengths for each line. The location of the mechanical rooms were assumed to be somewhere in the center of each airside (shown in Figure 2).

3.1 Physical Mapping

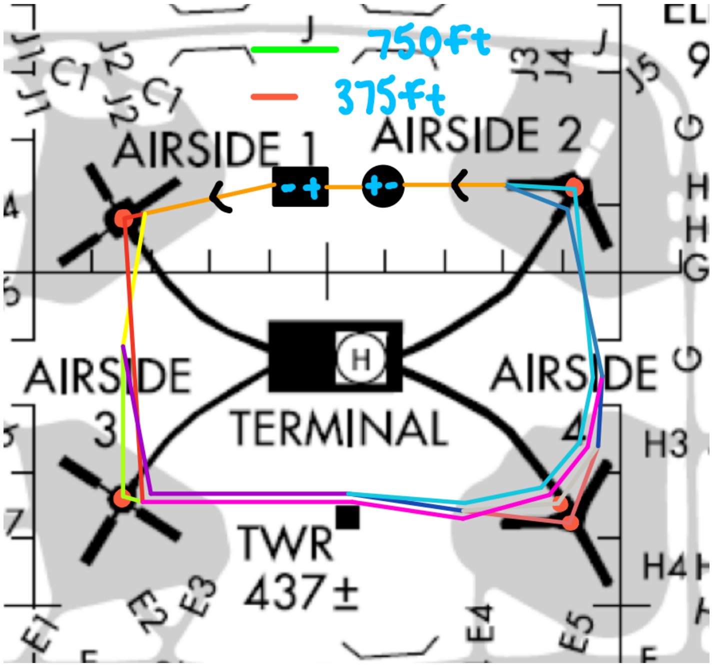

Figure 2: Physical Map of the Chiller System

The lateral distance between pipes shown above is slightly exaggerated to make it clear of the individual pipes. It is especially exaggerated around airside 4 (the parallel arrangement).

3.1.1 Pipe Lengths

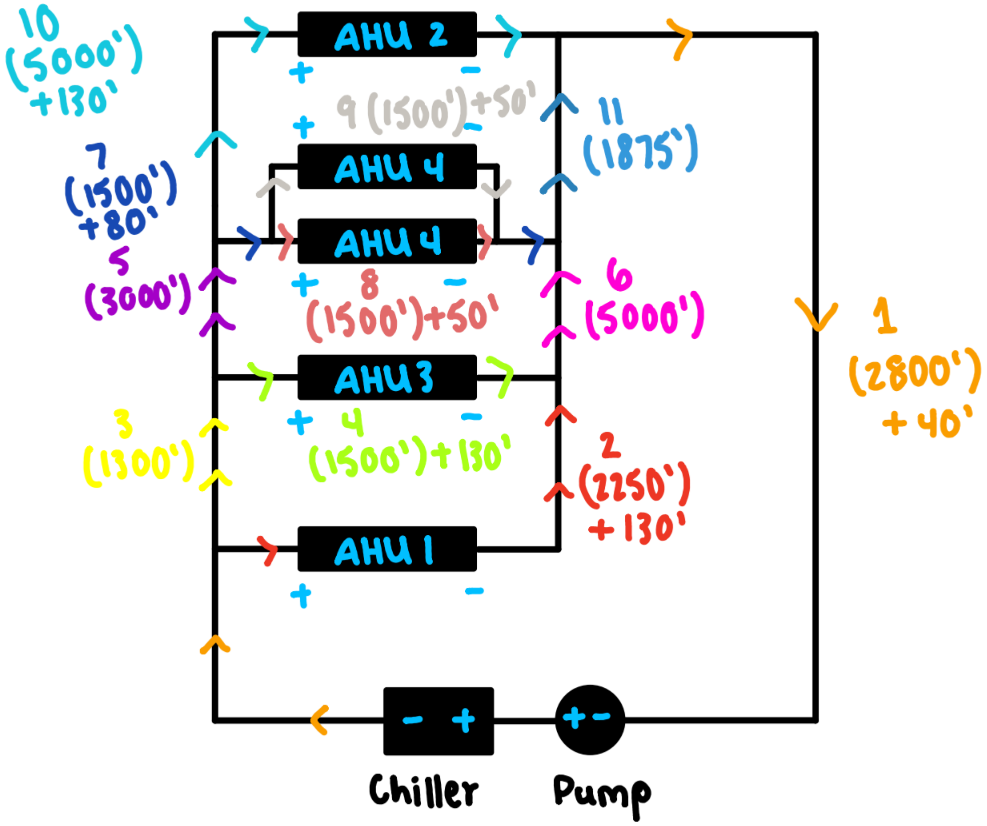

Figure 3: Z-Network Schematic of Lengths

Figure 3 shows the lengths associated with each line. The original length is shown in parentheses, and the added lengths are shown if there is a change in elevation or if the line is in a mechanical room (additional 50 feet). Here is a summary of the lengths for each of the 11 lines:

- Pipe 1: Original length is 2800'. Assumed to be at ground level, so add the piping that goes 20' up and 20' down to the tunnels. Total length is **2840'**
- Pipe 2: Original length is 2250'. This line runs through a mechanical room that is 40' up from the tunnels, then an additional 50' in the mechanical room, then 40' back down to the tunnels. The total length is **2380'**
- Pipe 3: There is no elevation change. The total length is **1300'**
- Pipe 4: The original length is 1500'. This line runs through a mechanical room, so an additional 130' is added to account for the elevation change and additional length within the room. The total length is **1630'**.
- Pipe 5: There is no elevation change. The total length is **3000'**.
- Pipe 6: There is no elevation change. The total length is **5000'**.
- Pipe 7: The original length on the map is 1500'. The line runs up to the mechanical room, but does not run through the mechanical room. Pipe 7 splits up prior to the parallel arrangement in airside 4, then meets up with lines 8 and 9 and returns to the tunnel. Although it is two different pipes, the flow rate is the same, so it will be analysed as if it were one pipe. The total length is **1580'**.
- Pipe 8: The original length is 1500'. It runs through the mechanical room, adding 50'. The

total length is **1550'**.

- Pipe 9: The original length is 1500'. It runs through the mechanical room, adding 50'. The total length is **1550'**.
- Pipe 10: The original length is 5000'. It runs through a mechanical room and changes elevation when doing so. The total length is **5130'**.
- Pipe 11: There is no elevation change. The total length is **1875'**.

3.1.2 Pipe Fittings and Valve Requirements

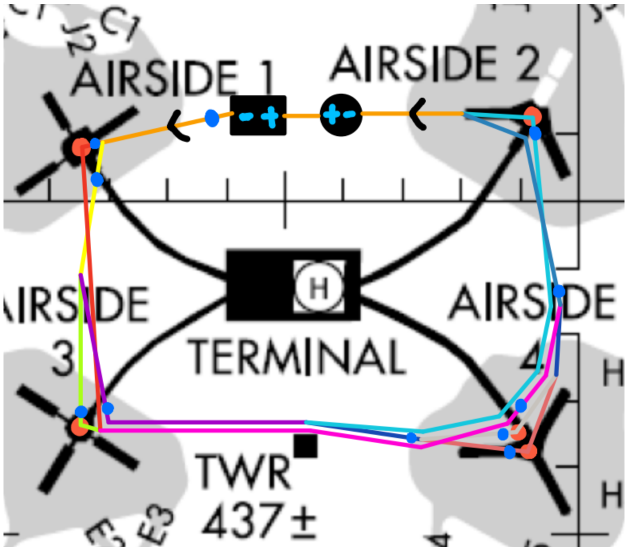

Figure 4: Physical Map of Valve Placements

Figure 4 shows the exaggerated valve placements (the blue dots). The specifications require that the flow is able to be completely shutoff valve; therefore, one shutoff valve will be placed in each line.

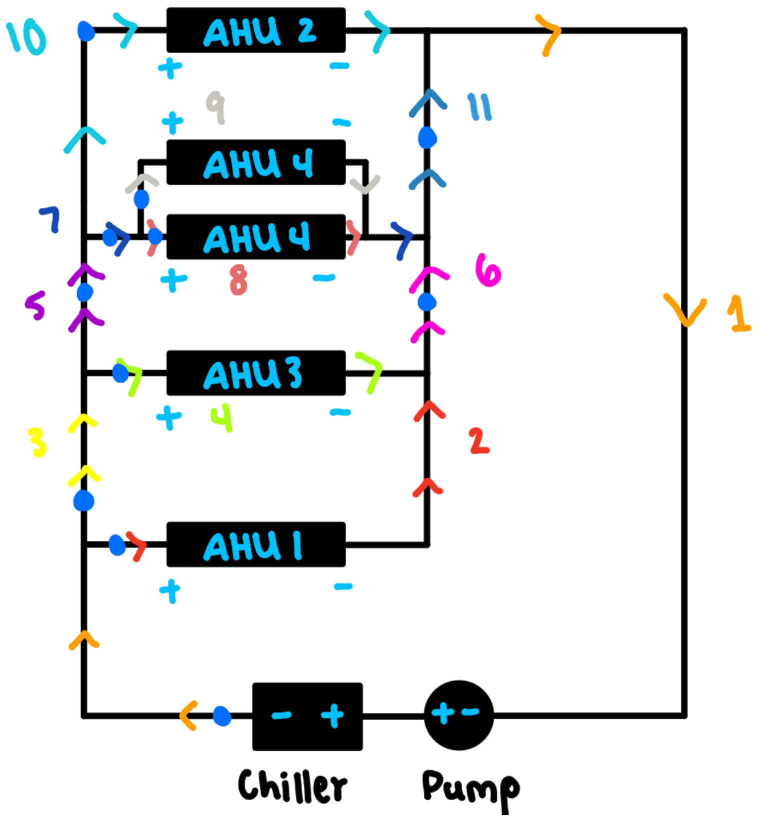

Figure 5: Schematic of Valve Placements

Figure 5 shows a clearer network of the pipe placements. The valves will be gate valves, since it is the most common type of shutoff valve. The loss per each gate valve is $K = 8f_T$ (each line will have $C = 8$ because this is the only loss considered that gets multiplied by f_T). The fittings used for this network will be standard tee (for splitting the flow) and elbow fittings (for changing direction). There is a lot of approximation when it comes to the fittings, but the major losses and losses across the air handling units for this case should triumph over the loss produced by the fittings, since the lengths of each pipe are thousands of feet. All the pipes will have a loss due to the entrance and exit for each individual pipe ($K = 1.78$). The loss due to the tee connections will be ignored (only 8 tee connections total), but losses due to elbows will be considered due to the larger quantity ($K_{elbow} = 0.75$). The number of elbows is estimated based on a change in direction looking down on the top plane (the point of view in the above figures) and the change in direction perpendicular to the top view (coming out or into the page from the point of view in the above figures). Here is a description for the minor losses in each pipe (each fitting is visualized in Figure 4):

- Pipe 1: It is possible with the mapping shown in Figure 2 and 4 for there to be no elbows because there is no change in direction on the top plane or an orthogonal plane coming out of the page. $K_1 = 1.78$
- Pipe 2: There is one elbow to change direction toward airside 3 and 4 elbows to change direction coming into and out of the page. $K_2 = 1.78 + 5(0.75) = 5.53$
- Pipe 3: No changes in direction. $K_3 = 1.78$
- Pipe 4: Contains an elbow to change direction toward airside 4 and four more due to the elevation change. $K_4 = 1.78 + 5(0.75) = 5.53$
- Pipe 5: There are 2 elbows to change direction toward airside 4. $K_5 = 1.78 + 2(0.75) = 3.28$

- Pipe 6: It has 2 elbows to change direction along the top plane. $K_6 = 1.78 + 2(0.75) = 3.28$
- Pipe 7: Contains no additional elbows. $K_7 = 1.78$
- Pipe 8: Contains 5 elbows. $K_8 = 1.78 + 5(0.75) = 5.53$
- Pipe 9: Contains 7 elbows $K_9 = 1.78 + 7(0.75) = 7.03$
- Pipe 10: Contains four elbows for changing direction on the top plane and an additional four for elevation change. $K_{10} = 1.78 + 8(0.75) = 7.78$
- Pipe 11: Contains 1 elbow. $K_{11} = 1.78 + 0.75 = 2.53$

3.2 Boundary Conditions

The minimum flow rate across each air handling unit may be found using this relationship,

$$q = \dot{m} c_p \Delta T$$
$$\dot{m} = \rho Q$$

where q is the energy in tons given in Table 1. Keeping an eye on the units, Q may be solved.

$$Q_{min} = \frac{q}{c_p \Delta T \rho} \rightarrow Q_{min} = \frac{10q}{3 c_p \Delta T \rho} \frac{ft^3}{s}$$

The above expression is true for the units of c_p and ρ defined in the Given section and with q in tons. The upper bound of the flow rate is going to be $3.5\% {\cdot} Q_{min} + Q_{min}$.

[2]:

```
# Finding the upper and lower bounds
q = np.array([480, 480, 225, 225, 480])  # In numerical order according to␣
 ↪Figure 5
c_p, rho, del_T = 0.491, 48.05, 25  # Units are described earlier

Q_min = (10*q)/(3*c_p*del_T*rho)
Q_min
```

[2]:

```
array([2.71272075, 2.71272075, 1.27158785, 1.27158785, 2.71272075])
```

[3]:

```
Q_max = Q_min + Q_min*0.035
Q_max
```

[3]:

```
array([2.80766598, 2.80766598, 1.31609343, 1.31609343, 2.80766598])
```

The boundary conditions are,

$2.713 \leq Q_2 < 2.808$
$2.713 \leq Q_4 < 2.808$
$1.272 \leq Q_8 < 1.316$
$1.272 \leq Q_9 < 1.316$
$2.713 \leq Q_{10} < 2.808$

where Q is in $\frac{ft^3}{s}$. The only other boundary condition is that the velocity of the fluid in all the pipes has to be less than $9\frac{ft}{s}$.

3.3 Diameter Selections

The minimum diameter for the pipes containing the air handling units may be found using,

$$Q = VA = V\frac{\pi}{4}D^2$$
$$D^2 = \frac{4Q}{V\pi}$$
$$D_{min} = \sqrt{\frac{4Q_{min}}{V_{max}\pi}}$$

where Q_{min} was solved above, and the maximum velocity is $9\frac{ft}{s}$.

[4]:

```
# Calculating the minimum diameters
D_min = (4*Q_min/(9*np.pi))**0.5
D_min  # In ft
```

[4]:

```
array([0.61949292, 0.61949292, 0.42413781, 0.42413781, 0.61949292])
```

Schedule 40 piping will be used for this analysis because it is commonly used for chiller systems and encompasses inner diameters greater than the values shown above. A chart for schedule 40 piping can be found here.

[5]:

```
# Getting a table of values to choose from
nps = np.array([5, 6, 8, 10, 12, 14, 16, 18, 20])  # Nominal Pipe Size (in)
od = np.array([5.563, 6.625, 8.625, 10.75, 12.75, 14, 16, 18, 20])  # Outer␣
 ↪Diameter (in)
t = np.array([0.258, 0.28, 0.322, 0.365, 0.406, 0.438, 0.5, 0.562, 0.594])  #␣
 ↪Thickness (in)
id_ = od - 2*t  # Inner Diameter (in)
id_ft = id_/12

df = pd.DataFrame({'NPS': nps, 'Outer Diameter (in)': od, 'Thickness (in)': t,␣
 ↪'Inner Diameter (in)': id_, 'Inner Diameter (ft)': id_ft})
# df
display(Latex(df.to_latex(index=False)))
```

NPS	Outer Diameter (in)	Thickness (in)	Inner Diameter (in)	Inner Diameter (ft)
5	5.563	0.258	5.047	0.420583
6	6.625	0.280	6.065	0.505417
8	8.625	0.322	7.981	0.665083
10	10.750	0.365	10.020	0.835000
12	12.750	0.406	11.938	0.994833
14	14.000	0.438	13.124	1.093667
16	16.000	0.500	15.000	1.250000
18	18.000	0.562	16.876	1.406333
20	20.000	0.594	18.812	1.567667

There are four diameters that need to be considered: the main line, the supply lines, the lines across air handling units 1, 2, and 3, and the lines across air handling units in airside 4. The main pipeline (line 1) needs to be the largest diameter because it will have the largest flow rate. Increasing the diameter will help to decrease the velocity. The lines with the air handling units need to be smaller, and air handling units in airside 4 can have smaller diameters than the lines that run through the other units because its flow rate doesn't have to be as high (as seen in the Boundary Conditions section).

The diameter choices are,

- 20 NPS for the main line
- 12 NPS for the supply and return lines
- 10 NPS for AHU 1, 2, and 3
- 8 NPS for AHU 4 (both lines)

Table 2 is a summary for all the decisions made thus far.

Table 2: Summary of Piping

Pipe	L (ft)	D (in)	# Valves	# Elbows	K	C
1	2840	18.812	1	0	1.78	8
2	2380	10.02	1	5	5.53	8
3	1300	11.938	1	0	1.78	8
4	1630	10.02	1	5	5.53	8
5	3000	11.938	1	2	3.28	8
6	5000	11.938	1	2	3.28	8
7	1580	11.938	1	0	1.78	8
8	1550	7.981	1	5	5.53	8
9	1550	7.981	1	7	7.03	8
10	5130	10.02	1	8	7.78	8
11	1875	11.938	1	1	2.53	8

Material: Galvanized Steel (Schedule 40)
Absolute Roughness: $\epsilon = 0.0005\,ft$
Working Fluid: Therminol D-12
Density: $\rho = 48.05\frac{lbm}{ft^3}$
Heat Capacity: $c_p = 0.491\frac{Btu}{lbm^\circ F}$
Viscosity: $\mu = 3.715\frac{lbm}{ft\,hr}$

Assumptions:

- Working fluid is constant temperature
- Loss is ignored for tee connections but valid for entrance and exit and for elbow connections. Although the system appears to be closed, each individual pipe experiences a loss due to fluid entering and exiting.
- The elbow connections for the layout on a large scale is considered. The elbow connections within a mechanical room is unknown and neglected.
- The estimated loss for fittings is approximated and may or may not be accurate, but most of the loss is major, meaning that it's ok to be off with the fittings.

Totals:

- Number of Tees: 8
- Number of Elbows: 35
- Pipe Length: 27,835 ft

3.4 Kirchhoff Setup

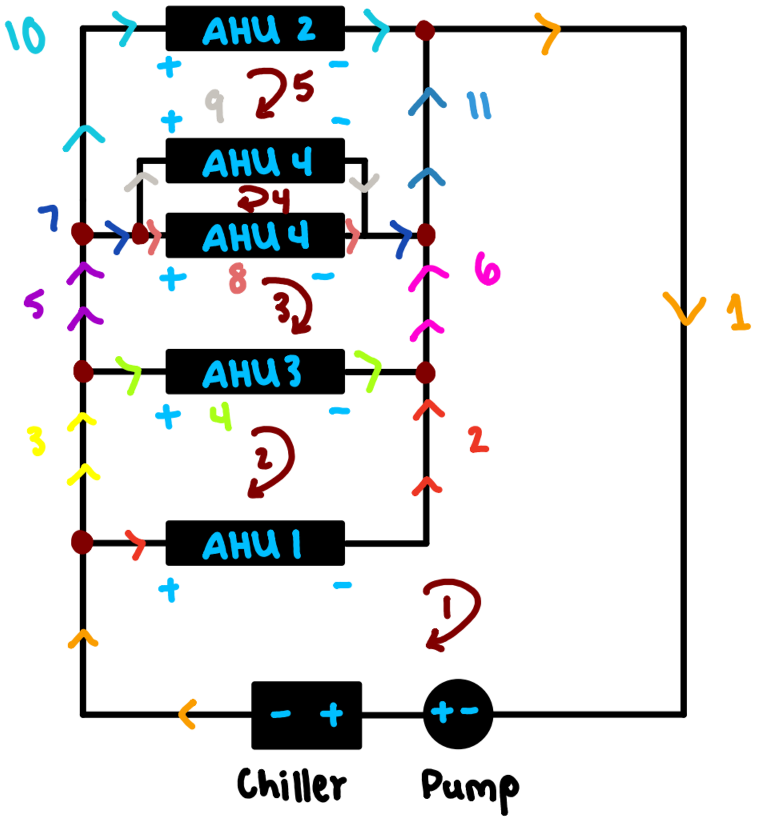

Figure 6: Kirchhoff Network

The relationship for the losses across the air handling units and the chiller result in units of feet upon converting *lbm* to *slugs* and dividing by g. This means that the rest of the terms in the system also need to be expressed in feet (make sure everything is divided by g for the loop equations). From Figure 6, the following system may be formed:

$$\begin{cases} Q_1 = Q_3 + Q_2 \\ Q_3 = Q_4 + Q_5 \\ Q_6 = Q_2 + Q_4 \\ Q_5 = Q_7 + Q_{10} \\ Q_7 = Q_8 + Q_9 \\ Q_{11} = Q_6 + Q_7 \\ h_2 + K_1 Q_2|Q_2| + h_6 + h_{11} + h_1 - W_s + 0.1Q_1|Q_1| = 0 \\ h_4 + K_3 Q_4|Q_4| - h_2 - K_1 Q_2|Q_2| + h_3 = 0 \\ h_7 + h_8 + K_4 Q_8|Q_8| - h_6 - h_4 - K_3 Q_4|Q_4| + h_5 = 0 \\ h_9 + K_4 Q_9|Q_9| - h_8 - K_4 Q_8|Q_8| = 0 \\ h_{10} + K_2 Q_{10}|Q_{10}| - h_{11} - h_7 - h_9 - K_4 Q_9|Q_9| = 0 \end{cases}$$

The $Q_i|Q_i|$ is very important because it allows for the losses to always work again the flow if a negative flow rate gets calculated. For this system, the flow rates should all be positive, but solvers may calculate a negative value before converging to the final result and having Q^2 may result in an invalid solution. This same concept should be applied to the h_i functions. In the code, everywhere a `.h()` is seen, just know that this is what is being returned:

$$h = \frac{8Q|Q|}{g\pi^2 D^4}(\frac{L}{D}f + K + C \cdot f_T)$$

3.5 Unbalanced Solution

With no corrections to the system, this is the result.

[6]:

```
# Define known constants
K1, K2, K3, K4 = 4.5, 4.5, 4.5, 10  # Loss coefficients

rho = 48.05/32.174  # In slugs per cubic feet
mu = 3.715/(3600*32.174)  # In slugs per (ft s) or lbf*s per ft squared
epsilon = 0.0005  # In ft

D20, D12, D10, D8 = np.array([18.812, 11.938, 10.02, 7.981])/12  # Diameters in␣
 ↪ft

Ws = 380  # in ft

def unbalanced(x):
    Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11 = x
    # all expressions need to be set to zero
    return [
        Q1 - Q2 - Q3,
        Q6 - Q2 - Q4,
        Q3 - Q4 - Q5,
        Q5 - Q10 - Q7,
        Q7 - Q8 - Q9,
        Q11 - Q6 - Q7,
        K1*Q2*abs(Q2) + p2.h(Q2) + p6.h(Q6) + p11.h(Q11) + p1.h(Q1) - Ws + 0.
 ↪1*Q1*abs(Q1),
        p4.h(Q4) + K3*Q4*abs(Q4) - p2.h(Q2) - K1*Q2*abs(Q2) + p3.h(Q3),
        p7.h(Q7) + p8.h(Q8) + K4*Q8*abs(Q8) - p6.h(Q6) - p4.h(Q4) -␣
 ↪K3*Q4*abs(Q4) + p5.h(Q5),
        p9.h(Q9) + K4*Q9*abs(Q9) - p8.h(Q8) - K4*Q8*abs(Q8),
        p10.h(Q10) + K2*Q10*abs(Q10) - p11.h(Q11) - p9.h(Q9) - K4*Q9*abs(Q9) -␣
 ↪p7.h(Q7)
    ]

p1 = Pipe(D20, 2840, epsilon, rho, mu, K=1.78, C=8)
p2 = Pipe(D10, 2380, epsilon, rho, mu, K=5.53, C=8)
p3 = Pipe(D12, 1300, epsilon, rho, mu, K=1.78, C=8)
p4 = Pipe(D10, 1630, epsilon, rho, mu, K=5.53, C=8)
p5 = Pipe(D12, 3000, epsilon, rho, mu, K=3.28, C=8)
p6 = Pipe(D12, 5000, epsilon, rho, mu, K=3.28, C=8)
p7 = Pipe(D12, 1580, epsilon, rho, mu, K=1.78, C=8)
```

```
p8 = Pipe(D8, 1550, epsilon, rho, mu, K=5.53, C=8)
p9 = Pipe(D8, 1550, epsilon, rho, mu, K=7.03, C=8)
p10 = Pipe(D10, 5130, epsilon, rho, mu, K=7.78, C=8)
p11 = Pipe(D12, 1875, epsilon, rho, mu, K=2.53, C=8)

pipes = [p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11]

Q_guess = np.array([5, 1, 4, 2, 2, 3, 1, 3, -2, 1, 4])  # This does satisfy␣
  ↪mass conservation
unbalanced_sol = fsolve(unbalanced, Q_guess)
unbalanced_sol
```

[6]:

```
array([14.52599372,  4.00530898, 10.52068474,  2.74871641,  7.77196833,
        6.75402539,  3.77988801,  1.89543065,  1.88445736,  3.99208032,
       10.5339134 ])
```

To capture the repetitive process of testing to see if the flow rates lie within range of the boundary conditions, this function was created.

[7]:

```
# Making a test function
# Returns a dataframe

def test(pipes_, flow_rates):
    lines = range(1, len(pipes_) + 1)
    Q_values, V_values = [], []
    AHU_1_3, AHU_4 = [], []
    V_less = []

    for pipe_, Q_, i in zip(pipes_, flow_rates, lines):
        Q_values.append(Q_)
        V_ = pipe_.V(Q_)
        V_values.append(V_)

        if V_ < 9:
            V_less.append('Yes')
        else:
            V_less.append('No')

        if i in [2, 4, 10]:
            if 2.713 <= Q_ < 2.808:
                AHU_1_3.append('Yes')
            else:
                AHU_1_3.append('No')
        else:
            AHU_1_3.append('-')

        if i in [8, 9]:
            if 1.272 <= Q_ < 1.316:
                AHU_4.append('Yes')
            else:
                AHU_4.append('No')
        else:
            AHU_4.append('-')

    return pd.DataFrame({'Pipe': lines, 'Q': Q_values, 'V': V_values, 'V<9':␣
 ↪V_less, '2.713<=Q<2.808': AHU_1_3, r'1.272<=Q<1.316': AHU_4})

unbalanced_test = test(pipes, unbalanced_sol)
# unbalanced_test
display(Latex(unbalanced_test.to_latex(index=False)))
```

Pipe	Q	V	V<9	2.713<=Q<2.808	1.272<=Q<1.316
1	14.525994	6.699353	Yes	-	-
2	4.005309	8.102744	Yes	No	-
3	10.520685	6.965156	Yes	-	-
4	2.748716	8.639212	Yes	Yes	-
5	7.771968	8.649708	Yes	-	-
6	6.754025	6.879611	Yes	-	-
7	3.779888	5.034698	Yes	-	-
8	1.895431	5.538109	Yes	-	No
9	1.884457	8.716428	Yes	-	No
10	3.992080	8.110157	Yes	No	-
11	10.533913	8.005548	Yes	-	-

The unbalanced solution shown above is the case where all the flow rates are either within range of the boundary conditions or above the range of the boundary conditions. For this system, this occurs with a main pump head of 380 ft. Line 4 appears to be what is causing the main issue. The balanced solution with only adding losses is to be considered.

3.6 Balanced Solution with Turn Down Valves

Adding turn down valves to the lines that exceed the boundary conditions is considered. Line 4 will not receive any turn down valves because that line is the limiting factor for the whole system. Also, leaving that line free from turn down valves will prevent bucking of the pump. The losses of gate valves are considered:

- $\frac{3}{4}$ open: $K = 0.9$
- $\frac{1}{2}$ open: $K = 4.5$
- $\frac{1}{4}$ open: $K = 21$

[8]:

```
# Getting the balanced solution using turn down valves
# Define new pipe objects with easy to manipulate the loss coefficients

k1_ = 0*0.9 + 0*4.5 + 0*24
k2_ = 0*0.9 + 0*4.5 + 16*24
k3_ = 1*0.9 + 0*4.5 + 0*24
```

```
k4_ = 0*0.9 + 0*4.5 + 12*24
k5_ = 0*0.9 + 0*4.5 + 2*24
k6_ = 0*0.9 + 0*4.5 + 0*24
k7_ = 0*0.9 + 0*4.5 + 0*24
k8_ = 0*0.9 + 0*4.5 + 25*24
k9_ = 0*0.9 + 0*4.5 + 25*24
k10_ = 0*0.9 + 0*4.5 + 15*24
k11_ = 0*0.9 + 0*4.5 + 0*24

p1_ = Pipe(D20, 2840, epsilon, rho, mu, K=1.78 + k1_, C=8)
p2_ = Pipe(D10, 2380, epsilon, rho, mu, K=5.53 + k2_, C=8)
p3_ = Pipe(D12, 1300, epsilon, rho, mu, K=1.78 + k3_, C=8)
p4_ = Pipe(D10, 1630, epsilon, rho, mu, K=5.53 + k4_, C=8)
p5_ = Pipe(D12, 3000, epsilon, rho, mu, K=3.28 + k5_, C=8)
p6_ = Pipe(D12, 5000, epsilon, rho, mu, K=3.28 + k6_, C=8)
p7_ = Pipe(D12, 1580, epsilon, rho, mu, K=1.78 + k7_, C=8)
p8_ = Pipe(D8, 1550, epsilon, rho, mu, K=5.53 + k8_, C=8)
p9_ = Pipe(D8, 1550, epsilon, rho, mu, K=7.03 + k9_, C=8)
p10_ = Pipe(D10, 5130, epsilon, rho, mu, K=7.78 + k10_, C=8)
p11_ = Pipe(D12, 1875, epsilon, rho, mu, K=2.53 + k11_, C=8)

new_pipes = [p1_, p2_, p3_, p4_, p5_, p6_, p7_, p8_, p9_, p10_, p11_]

Ws_ = 380

def balanced_turn_down(x):
    Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11 = x
    # all expressions need to be set to zero
    return [
        Q1 - Q2 - Q3,
        Q6 - Q2 - Q4,
        Q3 - Q4 - Q5,
        Q5 - Q10 - Q7,
        Q7 - Q8 - Q9,
        Q11 - Q6 - Q7,
        K1*Q2*abs(Q2) + p2_.h(Q2) + p6_.h(Q6) + p11_.h(Q11) + p1_.h(Q1) - Ws_ + 0.1*Q1*abs(Q1),
        p4_.h(Q4) + K3*Q4*abs(Q4) - p2_.h(Q2) - K1*Q2*abs(Q2) + p3_.h(Q3),
        p7_.h(Q7) + p8_.h(Q8) + K4*Q8*abs(Q8) - p6_.h(Q6) - p4_.h(Q4) - K3*Q4*abs(Q4) + p5_.h(Q5),
        p9_.h(Q9) + K4*Q9*abs(Q9) - p8_.h(Q8) - K4*Q8*abs(Q8),
        p10_.h(Q10) + K2*Q10*abs(Q10) - p11_.h(Q11) - p9_.h(Q9) - K4*Q9*abs(Q9) - p7_.h(Q7)
    ]

balanced_turn_down_sol = fsolve(balanced_turn_down, Q_guess)
balanced_turn_down_test = test(new_pipes, balanced_turn_down_sol)
```

```
# balanced_turn_down_test
display(Latex(balanced_turn_down_test.to_latex(index=False)))
```

Pipe	Q	V	V<9	2.713<=Q<2.808	1.272<=Q<1.316
1	10.979644	7.877801	Yes	-	-
2	2.791550	6.250675	Yes	Yes	-
3	8.188094	10.643675	No	-	-
4	2.801328	5.408279	Yes	Yes	-
5	5.386766	6.774984	Yes	-	-
6	5.592878	7.259036	Yes	-	-
7	2.611444	6.589383	Yes	-	-
8	1.306392	8.643382	Yes	-	Yes
9	1.305051	7.795462	Yes	-	Yes
10	2.775323	6.245239	Yes	Yes	-
11	8.204321	10.664769	No	-	-

The above solution has the following description:

- 16 additional 1/4 open gate valves on line 2 ($K_2 = 389.53$)
- 1 additional 3/4 open gate valve on line 3 ($K_3 = 2.68$)
- 12 additional 1/4 open gate valves on line 4 ($K_4 = 293.53$)
- 2 additional 1/4 open gate valves on line 5 ($K_5 = 51.28$)
- 25 additional 1/4 open gate valves on lines 8 and 9 ($K_8 = 605.53$ and $K_9 = 607.03$)
- 15 additional 1/4 open gate valves on line 10 ($K_{10} = 367.78$)

Even with these ridiculous loss additions, not all lines have velocities under 9 cfs. Because the additional loss required to reach a balance using turn down valves exceeds 50 in some pipes, this option will not be considered, nor attempted to further correct. Also, in order to gain more control over this type of analysis, turn down valves were added to all four lines (could not be avoided). This is not a good design because bucking of the pump would occur. Balancing using booster pumps is to be investigated.

3.7 Balanced Solution Using Booster Pumps

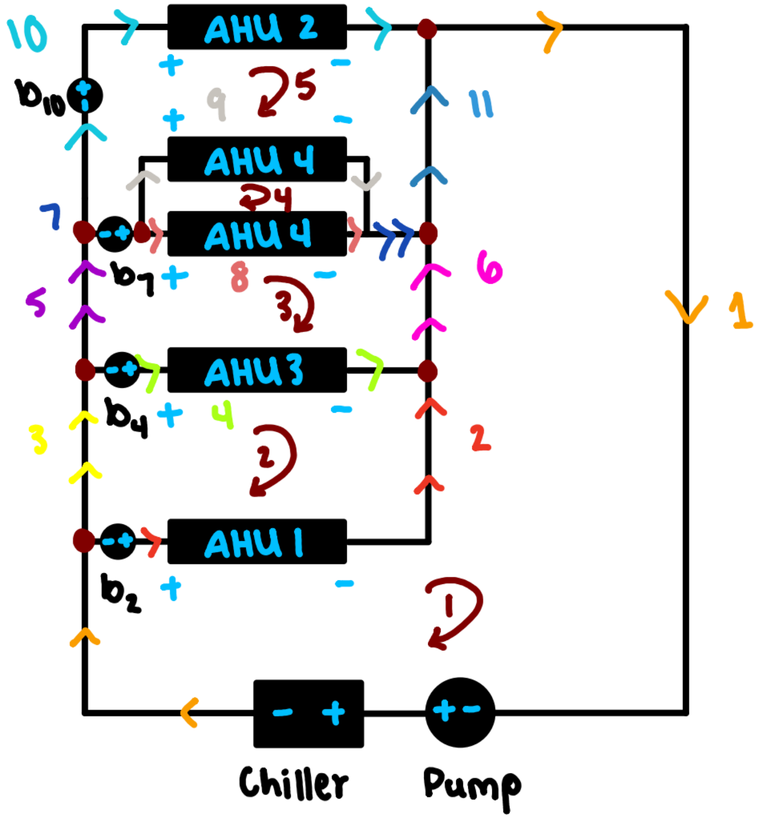

Figure 7: Booster Balance

The balanced solution will add booster pumps to line 2, 4, 7, and 10. This changes the system of equations to,

$$\begin{cases} Q_1 = Q_3 + Q_2 \\ Q_3 = Q_4 + Q_5 \\ Q_6 = Q_2 + Q_4 \\ Q_5 = Q_7 + Q_{10} \\ Q_7 = Q_8 + Q_9 \\ Q_{11} = Q_6 + Q_7 \\ h_2 + K_1 Q_2 |Q_2| + h_6 + h_{11} + h_1 - 150 + 0.1 Q_1 |Q_1| - b_2 = 0 \\ h_4 + K_3 Q_4 |Q_4| - h_2 - K_1 Q_2 |Q_2| + h_3 + b_2 - b_4 = 0 \\ h_7 + h_8 + K_4 Q_8 |Q_8| - h_6 - h_4 - K_3 Q_4 |Q_4| + h_5 + b_4 - b_7 = 0 \\ h_9 + K_4 Q_9 |Q_9| - h_8 - K_4 Q_8 |Q_8| = 0 \\ h_{10} + K_2 Q_{10} |Q_{10}| - h_{11} - h_7 - h_9 - K_4 Q_9 |Q_9| + b_7 - b_{10} = 0 \end{cases}$$

[9]:
```
# Balancing using booster pumps

Ws__ = 145
```

```python
def balanced(x, b2, b4, b7, b10):
    Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11 = x
    # all expressions need to be set to zero
    return [
        Q1 - Q2 - Q3,
        Q6 - Q2 - Q4,
        Q3 - Q4 - Q5,
        Q5 - Q10 - Q7,
        Q7 - Q8 - Q9,
        Q11 - Q6 - Q7,
        K1*Q2*abs(Q2) + p2.h(Q2) + p6.h(Q6) + p11.h(Q11) + p1.h(Q1) - Ws__ + 0.1*Q1*abs(Q1) - b2,
        p4.h(Q4) + K3*Q4*abs(Q4) - p2.h(Q2) - K1*Q2*abs(Q2) + p3.h(Q3) + b2 - b4,
        p7.h(Q7) + p8.h(Q8) + K4*Q8*abs(Q8) - p6.h(Q6) - p4.h(Q4) - K3*Q4*abs(Q4) + p5.h(Q5) + b4 - b7,
        p9.h(Q9) + K4*Q9*abs(Q9) - p8.h(Q8) - K4*Q8*abs(Q8),
        p10.h(Q10) + K2*Q10*abs(Q10) - p11.h(Q11) - p9.h(Q9) - K4*Q9*abs(Q9) - p7.h(Q7) + b7 - b10
    ]

b = 67.65, 101.95, 55.7, 44.9
balanced_solution = fsolve(balanced, Q_guess, args=(*b, ))
balanced_booster_test = test(pipes, balanced_solution)
# balanced_booster_test
display(Latex(balanced_booster_test.to_latex(index=False)))
```

Pipe	Q	V	V<9	2.713<=Q<2.808	1.272<=Q<1.316
1	10.692539	5.732650	Yes	-	-
2	2.713009	7.451870	Yes	Yes	-
3	7.979530	6.007410	Yes	-	-
4	2.713247	7.666423	Yes	Yes	-
5	5.266283	6.254602	Yes	-	-
6	5.426256	6.043785	Yes	-	-
7	2.552338	6.691457	Yes	-	-
8	1.279845	7.525854	Yes	-	Yes
9	1.272493	5.704194	Yes	-	Yes
10	2.713944	5.054272	Yes	Yes	-
11	7.978595	5.784762	Yes	-	-

The results above show that the system is balanced at $b_2 = 67.65\,ft$, $b_4 = 101.95\,ft$, $b_7 = 55.7\,ft$, and $b_{10} = 44.9\,ft$ and with a dramatic decrease in the pump head at 145 ft.

3.8 Power Consumption

The power added to the fluid may be calculated using,

$$power = \rho Q W_s$$

W_s needs to be in units of $\frac{ft^2}{s^2}$, so it is necessary to multiply by $g\,\frac{ft}{s^2}$, but if ρ is in $\frac{lbm}{ft^3}$, then it won't be necessary to multiply by g. For the power consumed using the main pump only with a head of 380 feet is,

```
[10]: # power for main pump only
      # using the flow rate in line 1 that was calculated above
      power_A = 48.05*10.979644*380
      power_A/550   # In hp
```

```
[10]: 364.5042178109091
```

The power for the booster pump option is the summation of all the pump powers for each line.

```
[11]: # Getting the power for the booster pump option
      power_B = 48.05*(10.692539*145 + 2.713009*67.65 + 2.713247*101.95 + 2.552338*55.
        ↪7 + 2.713944*44.9)
      power_B/550
```

```
[11]: 198.71645867724547
```

4 Verification

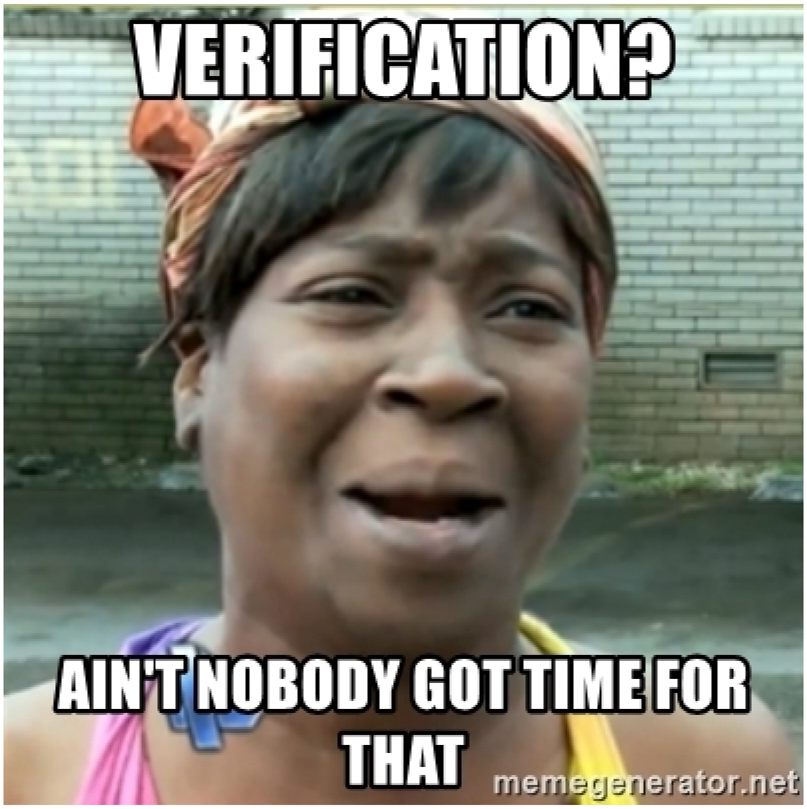

Figure 8: Me Right Now

The math for the booster pump analysis can be confirmed with a hardy cross solution.

[12]:

```
# Getting the hardy cross solution

# Connection matrix
N = np.transpose([
    [1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1],
    [0, -1, 1, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, -1, 1, -1, 1, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, -1, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, -1, 0, -1, 1, -1]
])

# Losses/gains in each line
b2_, b4_, b7_, b10_ = b
h = [
    lambda Q: 0.1*Q*abs(Q) - 145,
    lambda Q: K1*Q*abs(Q) - b2_,
    lambda Q: 0,
    lambda Q: K3*Q*abs(Q) - b4_,
    lambda Q: 0,
    lambda Q: 0,
    lambda Q: -b7_,
    lambda Q: K4*Q*abs(Q),
    lambda Q: K4*Q*abs(Q),
    lambda Q: K2*Q*abs(Q) - b10_,
    lambda Q: 0
]

dh = [
    lambda Q: 2*0.1*abs(Q),
    lambda Q: 2*K1*abs(Q),
    lambda Q: 0,
    lambda Q: 2*K3*abs(Q),
    lambda Q: 0,
    lambda Q: 0,
    lambda Q: 0,
    lambda Q: 2*K4*abs(Q),
    lambda Q: 2*K4*abs(Q),
    lambda Q: 2*K2*abs(Q),
    lambda Q: 0
]

hardy_solution = hardy_cross(pipes, Q_guess, N, h=h, dh=dh)
hardy_solution
```

[12]:

```
array([10.69225486,  2.71301168,  7.97924318,  2.71323258,  5.2660106 ,
        5.42624426,  2.55222046,  1.27982131,  1.27239916,  2.71379014,
        7.97846473])
```

The Hardy Cross solution is the same as the Kirchhoff solution, but the tolerance of the Hardy Cross solution is greater than the numerical solver.

5 Results and Discussion

The recommended choice for this piping network is the option of adding booster pumps. There are some downsides for adding booster pumps, such as additional maintenance and initial cost. However, adding booster pumps allows for more control over the system, and consumes far less power than the alternative choice of overloading the main pump. It is better to keep adding power until it's just enough to reach the minimum requirements than to overshoot.

The option for choosing the balanced system with turn down valves consumed so much power that it should be completely neglected. Also notice that in order to achieve a range that is remotely close to the boundary conditions, bucking of the pump had to have occurred because the analysis required that there be turn down valves in all lines. Furthermore, a power consumption analysis will show that the booster pump choice is superior ($power_{turndown} = 365\,hp$ and $power_{booster} = 200\,hp$):

$\frac{365-200}{200}(100) = 82.5$

The power required from the turn down valve analysis is 82.5% more than the power required for the booster pump analysis.