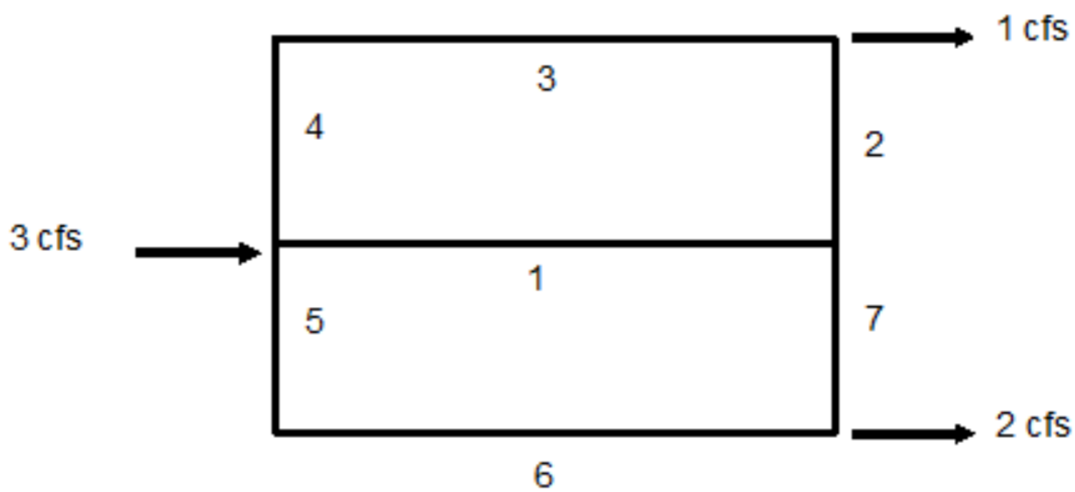
Kirchhoff Example

February 12, 2022

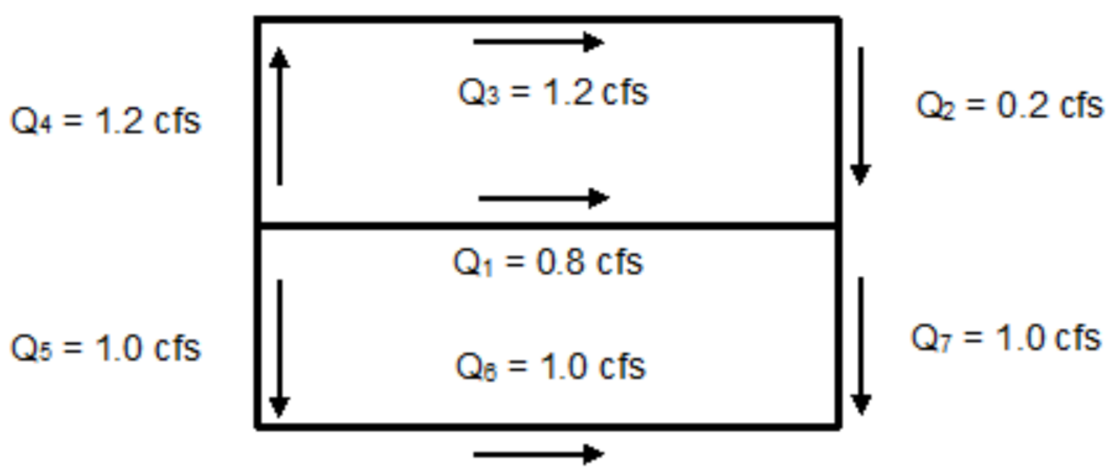
[1]:
```
from msu_esd import Pipe
from scipy.optimize import fsolve
import numpy as np
```

1 Example 1.11



Pipe	L (ft)	D (in)	K	C	ϵ (ft)
1	2000	12	0	0	0.00015
2	2000	8	0	0	0.00015
3	3000	6	0	0	0.00015
4	4000	6	0	0	0.00015
5	1000	8	0	0	0.00015
6	3000	8	0	0	0.00015
7	2000	8	0	0	0.00015

The guess values are,



Define pipe objects first.

[2]:
```python
epsilon = 0.00015
rho = 1.94
mu = 3.104e-5

pipe1 = Pipe(1, 2000, epsilon, rho, mu)
pipe2 = Pipe(8/12, 2000, epsilon, rho, mu)
pipe3 = Pipe(6/12, 3000, epsilon, rho, mu)
pipe4 = Pipe(6/12, 4000, epsilon, rho, mu)
pipe5 = Pipe(8/12, 1000, epsilon, rho, mu)
pipe6 = Pipe(8/12, 3000, epsilon, rho, mu)
pipe7 = Pipe(8/12, 2000, epsilon, rho, mu)
```

1.1 No Additional Devices

Set up a system of equations. In order to utilize `fsovle`, the function needs to return an array of values that are supposed to be zero. In other words, all the equations need to be set equal to zero.

[3]:
```python
def no_devices(x):
    Q1, Q2, Q3, Q4, Q5, Q6, Q7 = x
    return [
        Q1 + Q4 + Q5 - 3,
        Q1 + Q2 - Q7,
        Q6 + Q7 - 2,
        1 + Q2 - Q3,
        Q5 - Q6,
        pipe4.h(Q4) + pipe3.h(Q3) + pipe2.h(Q2) - pipe1.h(Q1),
        pipe1.h(Q1) + pipe7.h(Q7) - pipe6.h(Q6) - pipe5.h(Q5)
    ]

solution = fsolve(no_devices, np.array([0.8, 0.2, 1.2, 1.2, 1, 1, 1]))
solution
```

[3]:
```
array([ 1.86619223, -0.76215372,  0.23784628,  0.23784628,  0.89596149,
        0.89596149,  1.10403851])
```

1.2 Heat Exchanger in Line 1

If the loss of the heat exchanger is $50{Q_1}^2$, then

[4]:
```python
def heat_exchanger(x):
    Q1, Q2, Q3, Q4, Q5, Q6, Q7 = x
    return [
        Q1 + Q4 + Q5 - 3,
        Q1 + Q2 - Q7,
        Q6 + Q7 - 2,
        1 + Q2 - Q3,
        Q5 - Q6,
        pipe4.h(Q4) + pipe3.h(Q3) + pipe2.h(Q2) - pipe1.h(Q1) - 50*Q1*abs(Q1),
```

```
        pipe1.h(Q1) + pipe7.h(Q7) - pipe6.h(Q6) - pipe5.h(Q5) + 50*Q1*abs(Q1)
    ]

solution = fsolve(heat_exchanger, np.array([0.8, 0.2, 1.2, 1.2, 1, 1, 1]))
solution
```

[4]:
```
array([ 0.80977802, -0.436375  ,  0.563625  ,  0.563625  ,  1.62659699,
        1.62659699,  0.37340301])
```

1.3 Add a Pump in Line 1

If the pump adds 203.5 ft to the system,

[5]:
```
def heat_exchanger_with_pump(x):
    Q1, Q2, Q3, Q4, Q5, Q6, Q7 = x
    return [
        Q1 + Q4 + Q5 - 3,
        Q1 + Q2 - Q7,
        Q6 + Q7 - 2,
        1 + Q2 - Q3,
        Q5 - Q6,
        pipe4.h(Q4) + pipe3.h(Q3) + pipe2.h(Q2) - pipe1.h(Q1) - 50*Q1*abs(Q1) +␣
 ↪203.5,
        pipe1.h(Q1) + pipe7.h(Q7) - pipe6.h(Q6) - pipe5.h(Q5) + 50*Q1*abs(Q1) -␣
 ↪203.5
    ]

solution = fsolve(heat_exchanger_with_pump, np.array([0.8, 0.2, 1.2, 1.2, 1, 1,␣
 ↪1]))
solution
```

[5]:
```
array([ 2.00002057, -0.81294739,  0.18705261,  0.18705261,  0.81292682,
        0.81292682,  1.18707318])
```

1.4 Large Pump in Line 6

If we remove the previous devices and add only the pump to line 6 with a value of 1000 ft,

[6]:
```
def pump_6(x):
    Q1, Q2, Q3, Q4, Q5, Q6, Q7 = x
    return [
        Q1 + Q4 + Q5 - 3,
        Q1 + Q2 - Q7,
        Q6 + Q7 - 2,
        1 + Q2 - Q3,
        Q5 - Q6,
        pipe4.h(Q4) + pipe3.h(Q3) + pipe2.h(Q2) - pipe1.h(Q1),
        pipe1.h(Q1) + pipe7.h(Q7) - pipe6.h(Q6) - pipe5.h(Q5) + 1000
```

```
    ]

solution = fsolve(pump_6, np.array([0.8, 0.2, 1.2, 1.2, 1, 1, 1]))
solution
```

[6]:
```
array([-4.92433211, -1.24960309, -0.24960309, -0.24960309,  8.1739352 ,
        8.1739352 , -6.1739352 ])
```