Determination of Pump Efficiency and Frictional Losses in Copper Piping System

Lucas Culverhouse
April 26, 2023
ECH 145B Section A02

Executive Summary

Through the experiments performed it was determined that piping system configuration can have a significant impact on the performance of the pumping system in terms of efficiency and total flow rate. Several piping system configurations were measured with varying pipe lengths, sets of fittings and relative outlet heights. It was found that significant flow rate losses can be expected as outlet height relative to the pump height increases. It was also found that frictional losses lead to flow rate loss, which scales with longer piping systems.

Theory & Background

First we will discuss the Reynolds number and how it relates to piping systems. The Reynolds number is a dimensionless number that relates to the character of the flow situation. This tells you if the flow is laminar or turbulent. This matters because a laminar flow and turbulent flow have very different characteristics in terms of mass and heat transfer. The Reynolds number can be calculated as:

$$Re = \frac{uL}{\nu} \; Eq.\ 1$$

Where L is the characteristic length, which is the pipe diameter for a cylindrical pipe geometry. The condition determining if flow is turbulent is:

$$Re < 10^5$$

That is, below 10^5 the flow will be laminar and above it will be turbulent.

Piping systems are critical for fluid transport in chemical engineering applications. Understanding how the layout of the piping system impacts how it behaves is important when designing such systems. One simple way to understand the behavior of a piping system is using the following equation:

$$\Delta(\frac{<v>^2}{2g}) + \frac{\Delta P}{\rho g} + \Delta z = \frac{W}{\rho g Q} - \frac{E}{\rho g Q} \quad Eq. 2$$

Each term in this governing equation is in terms of “head”, which has units of length. Breaking down the terms we have:

$\Delta(\frac{<v>^2}{2g})$ — Velocity head

$\frac{\Delta P}{\rho g} + \Delta z$ — Piezometric head

$\frac{W}{\rho g Q}$ — Power supplied by pump

$\frac{E}{\rho g Q}$ — Dissipation head loss

Of these we neglect velocity head and frictional losses for the rest of the analysis. Under these assumptions we can put Eq. 2 into terms of Q:

$$Q = \frac{\epsilon E}{\rho g}\left[\frac{1}{\frac{\Delta P}{\rho g}+\Delta z}\right] Eq.\ 3$$

This allows us to frame Q, the volumetric flow rate, as a function of the piezometric head. This proves useful because it allows us to analyze how the change in outlet height relative to the pump affects the final flow rate, and thus velocity, that gets seen at the outlet. Here, ε refers to the efficiency of the pump being used to move the water in the system.

Hidden in eq. 1 there are two so-called “frictional head loss” factors that will be further discussed. These are the major head loss:

$$h_f = f\frac{Lv^2}{D2g}\ Eq.\ 4$$

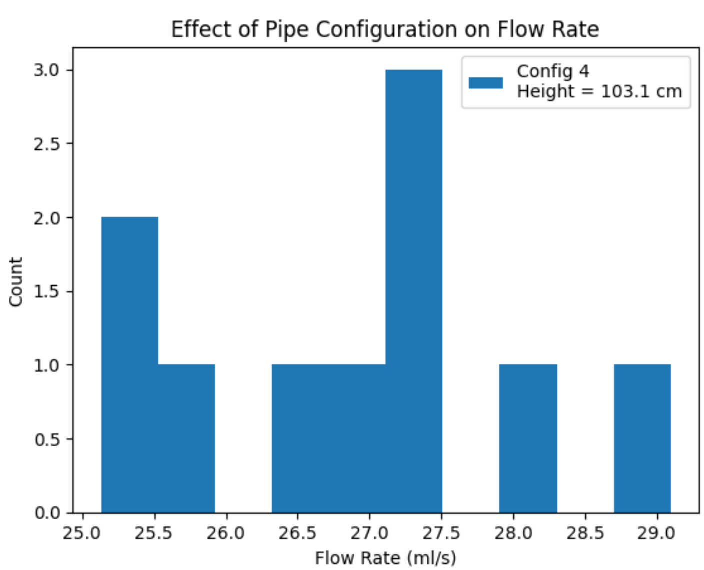
Where f is the friction factor that depends on the Reynold’s number and
And the minor head loss:

$$h_m = K\frac{v^2}{2g}\ Eq.\ 5$$

Where *K* is the friction factor that is based on the fitting geometry. These factors are tabulated for many kinds of fittings and can be used to further analyze our problem. Using these two equations we can account for the frictional losses, where before we ignored them.

Results

Three separate experiments were performed, which each measured the flow rate at the exit of different piping and tubing configurations. Below are the results of a selected trial of two of those experiments



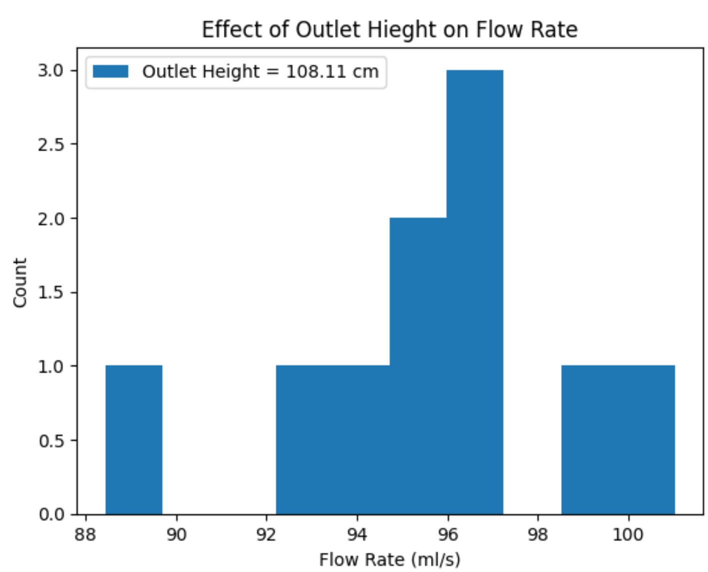


Figure 1. Selected Trial of Pipe Configuration and Outlet Height Experiments. Pipe configuration 4 was the longest configuration, going through 3 long pipes, 2 short pipes, 3 elbow

connectors and 3 ball valves. For the outlet height experiment a plastic tube with a length of 72 in and a diameter of 0.51 in was used for each trial.

For both the outlet height and pipe configuration experiments flow rate was plotted versus the inverse of the piezometric head in order to find the pump efficiency.

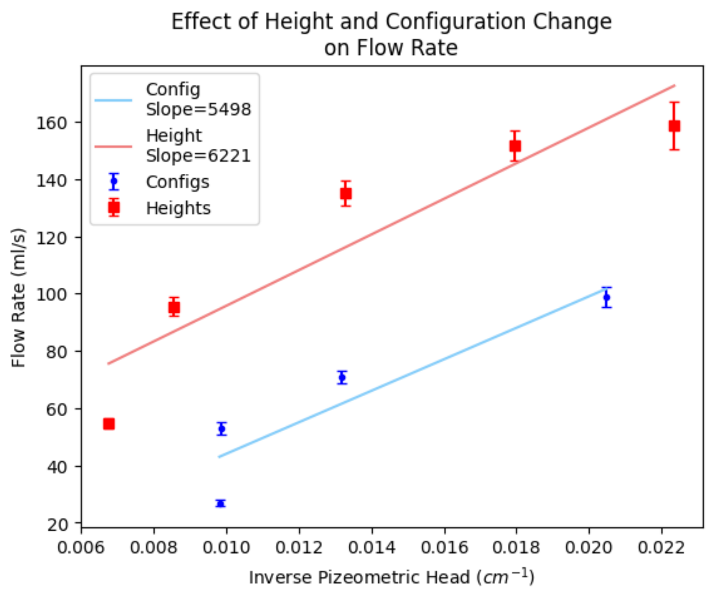


Figure 2. Effects of Inverse Piezometric Head on Measured Flow Rate for Select Experiments. Pipe configurations 1, 2 ,3 and 4 are shown with the pump inlet placed below, meaning Δz is positive. For the outlet height experiment a plastic tube with a length of 72 in and a diameter of 0.51 in was used for each trial.

From this plot, and using eq. 3 we can determine the pump efficiency to be 50.7 ± 1%

From this pump efficiency, and a calculation of the frictional losses, we can plot our measured flow rates, versus the flow rate predicted by theory

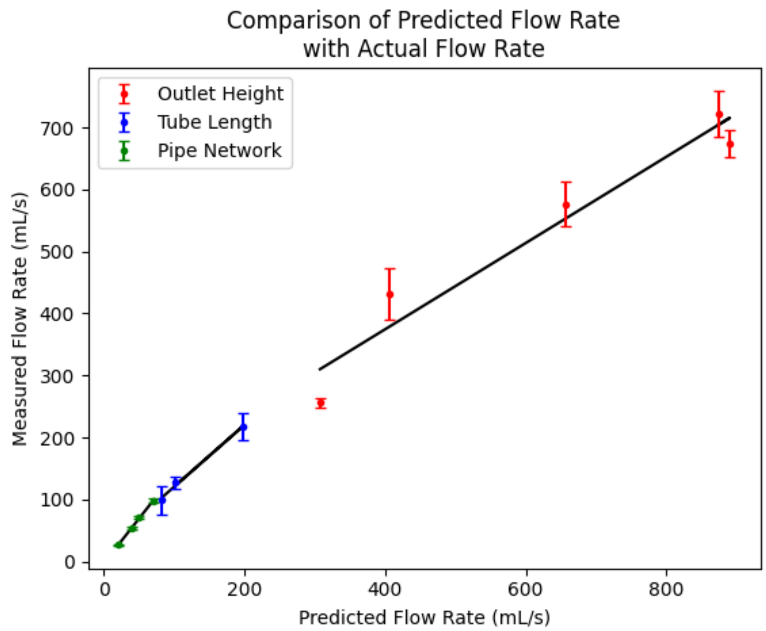


Figure 3. Deviation of Predicted Flow Rate From Measured Flow Rate. Each experiment is represented here, with the total flow rate of the outlet height experiment being the most varied.

Discussion

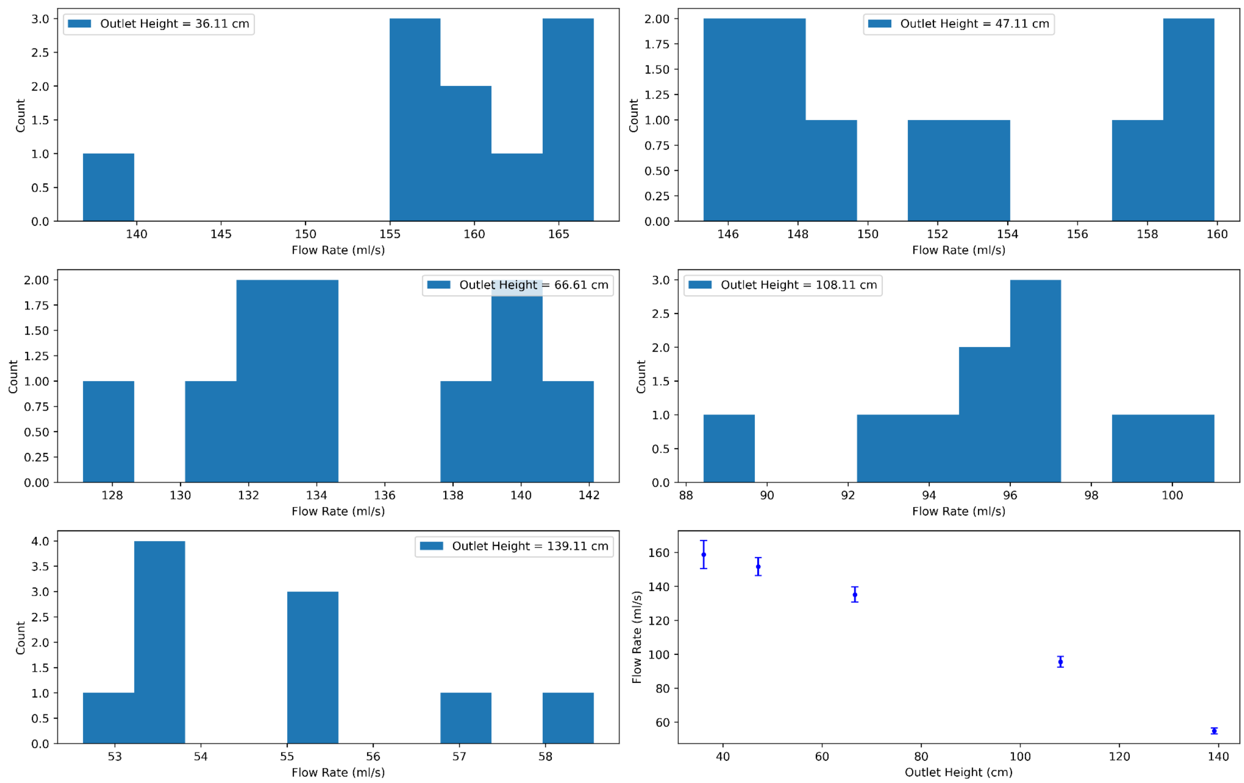
As we can see from figure 1 and the presentation of error bars in the other two figures, the error in the flow rate measurement was quite significant. This is likely due to the measurement method of a hand timer and a separate person placing the collection cylinder under the water stream when directed. This introduces significant possibility for mis-timings in the data collection process. We can also see from figure 2 that the effect of the piezometric head on the total flow rate is about the same for both experiments. This would imply that the contribution of the other factors present, frictional losses, are not that impactful on the flow rate produced. This provides a justification for ignoring the frictional losses, as it shows the piezometric head can be considered more significant that those other factors. We can see that our results do deviate from the predicted theory. This can be explained by the simplifications made in the calculations of the theoretical values, as well as the measurement errors obtained.

Conclusions and Recommendations

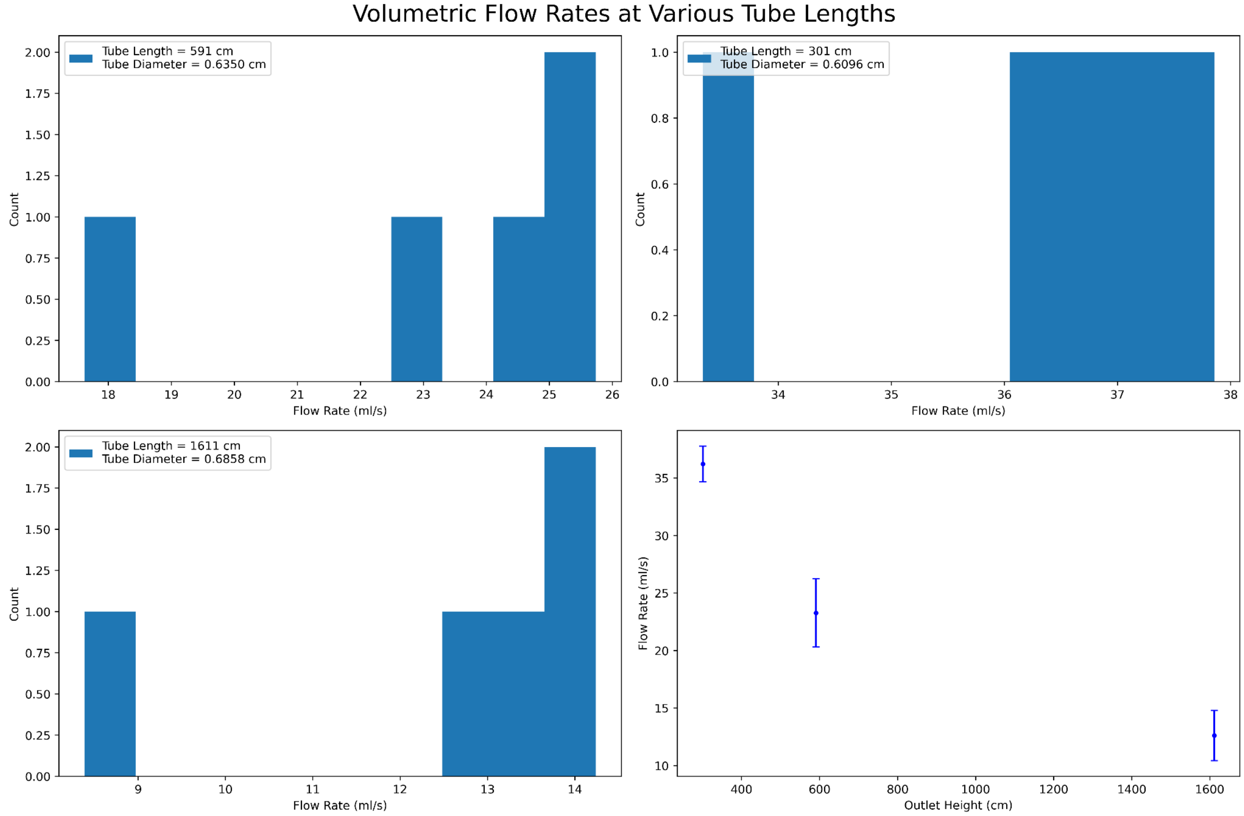
Piezometric head appears to be the most significant for these types of piping systems. As a first approximation, the simplified form of eq. 3 can be used to create solid analysis of such systems.

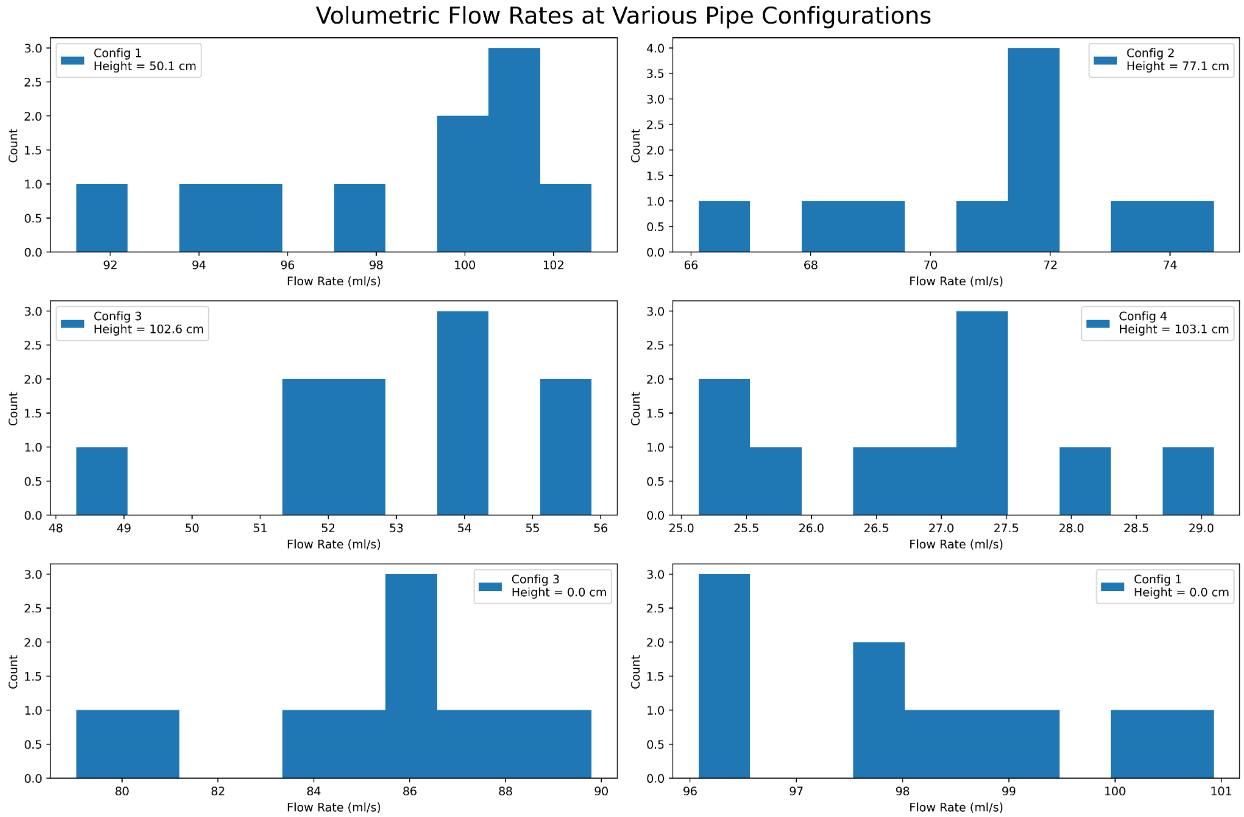
Appendix

Volumetric Flow Rates at Various Outlet Heights



Volumetric Flow Rates at Various Tube Lengths




Volumetric Flow Rates at Various Pipe Configurations
Config 1
Height = 50.1 cm
Config 2
Height = 77.1 cm
Config 3
Height = 102.6 cm
Config 4
Height = 103.1 cm
Config 3
Height = 0.0 cm
Config 1
Height = 0.0 cm
Count
Flow Rate (ml/s)

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy

init_tube_data = np.loadtxt("./Data/1.1_initial_tube.csv", delimiter=",", skiprows=
flow_rate = np.zeros((50, 2))
flow_rate[:, 0] = init_tube_data[:, 0] - 3.5 * 2.54
flow_rate[:, 1] = init_tube_data[:, 2] / init_tube_data[:, 1]
fig,axs = plt.subplots(3,2)
fig.set_figheight(10)
fig.set_figwidth(15)

avg_flow_rate_h = np.zeros((5,2))
std_flow_rate_h = np.zeros((5,2))
for i in range(5):
    avg_flow_rate_h[i, 0] = flow_rate[i*10:i*10+10, 0].mean()
    avg_flow_rate_h[i, 1] = flow_rate[i*10:i*10+10, 1].mean()
    std_flow_rate_h[i, 0] = flow_rate[i*10:i*10+10, 0].std()
    std_flow_rate_h[i, 1] = flow_rate[i*10:i*10+10, 1].std()

    ax = axs.flat[i]
    if i == 3:
        ax.hist(height_hist:=flow_rate[i*10:i*10+10, 1], label=f"Outlet Height = {f
    else:
        ax.hist(flow_rate[i*10:i*10+10, 1], label=f"Outlet Height = {flow_rate[i*10
    ax.set_ylabel("Count")
    ax.set_xlabel("Flow Rate (ml/s)")
    ax.legend()

axs[2,1].errorbar(avg_flow_rate_h[:,0], avg_flow_rate_h[:,1], std_flow_rate_h[:,1],
axs[2,1].set_ylabel("Flow Rate (ml/s)")
axs[2,1].set_xlabel("Outlet Height (cm)")
font = {"size": 20}
fig.suptitle("Volumetric Flow Rates at Various Outlet Heights", **font)
fig.tight_layout()
fig.set_dpi(400)
plt.show()

copper_tube_data = np.loadtxt("./Data/1.1_copper_tube.csv", delimiter=",", skiprows
flow_rate = np.zeros((15, 3))
flow_rate[:, 0] = copper_tube_data[:, 0] - 3.5 * 2.54
flow_rate[:, 1] = copper_tube_data[:, 1] * 2.54
flow_rate[:, 2] = copper_tube_data[:, 3] / copper_tube_data[:, 2]
fig, axs = plt.subplots(2,2)
fig.set_figwidth(15)
fig.set_figheight(10)
avg_flow_rate = np.zeros((3,3))
std_flow_rate = np.zeros((3,3))
for i in range(3):
    avg_flow_rate[i, 0] = flow_rate[i*5:i*5+5, 0].mean()
    avg_flow_rate[i, 1] = flow_rate[i*5:i*5+5, 1].mean()
    avg_flow_rate[i, 2] = flow_rate[i*5:i*5+5, 2].mean()
    std_flow_rate[i, 0] = flow_rate[i*5:i*5+5, 0].std()
    std_flow_rate[i, 1] = flow_rate[i*5:i*5+5, 1].std()
```

```
    std_flow_rate[i, 2] = flow_rate[i*5:i*5+5, 2].std()

    ax = axs.flat[i]
    ax.hist(flow_rate[i*5:i*5+5, 2], label=f"Tube Length = {flow_rate[i*5, 0]:.0f}
    ax.set_ylabel("Count")
    ax.set_xlabel("Flow Rate (ml/s)")
    ax.legend()

axs[1,1].errorbar(avg_flow_rate[:,0], avg_flow_rate[:,2], std_flow_rate[:,2], fmt="
axs[1,1].set_ylabel("Flow Rate (ml/s)")
axs[1,1].set_xlabel("Outlet Height (cm)")
font = {"size": 20}
fig.suptitle("Volumetric Flow Rates at Various Tube Lengths", **font)
fig.tight_layout()
fig.set_dpi(400)
plt.show()

diff_height_data = np.loadtxt("./Data/1.2_diff_height.csv", delimiter=",", skiprows
same_height_data = np.loadtxt("./Data/1.2_same_height.csv", delimiter=",", skiprows
config_data = np.append(diff_height_data, same_height_data, axis=0)
flow_rate = config_data[:, 3] / config_data[:, 2]
config = config_data[:, 0]
height = config_data[:, 1]
height[:40] -= 3.5 * 2.54

fig, axs = plt.subplots(3,2)
fig.set_figwidth(15)
fig.set_figheight(10)
for i in range(6):
    ax = axs.flat[i]
    if i == 3:
        ax.hist(pipe_hist:=flow_rate[i*10:i*10+10], label=f"Config {config[i*10]:.0
    else:
        ax.hist(flow_rate[i*10:i*10+10], label=f"Config {config[i*10]:.0f}\nHeight
    ax.set_ylabel("Count")
    ax.set_xlabel("Flow Rate (ml/s)")
    ax.legend()
font = {"size": 20}
fig.suptitle("Volumetric Flow Rates at Various Pipe Configurations", **font)
fig.tight_layout()
fig.set_dpi(400)
plt.show()
```

In [ ]:

```
plt.hist(height_hist, label="Outlet Height = 108.11 cm")
plt.xlabel("Flow Rate (ml/s)")
plt.ylabel("Count")
plt.title("Effect of Outlet Hiegth on Flow Rate")
plt.legend()
plt.show()

plt.hist(pipe_hist, label="Config 4\nHeight = 103.1 cm")
plt.xlabel("Flow Rate (ml/s)")
plt.ylabel("Count")
plt.title("Effect of Pipe Configuration on Flow Rate")
```

```
plt.legend()
plt.show()
```

In [ ]:

```
p_water = 997
visc_water = 0.89
g = 9.81

flow_rates = config_data[:, 3] / config_data[:, 2]
heights = config_data[:, 1] - 3.5 * 2.54
avg_flow_rate = np.array([np.mean(flow_rates[i * 10 : i * 10 + 10]) for i in range(
std_flow_rate = np.array([np.std(flow_rates[i * 10 : i * 10 + 10]) for i in range(6
liquid_head = np.ones(flow_rates.shape) * (6.5 - 3.5) * 2.54

pz_head = heights + liquid_head
inv_pz_head = 1 / pz_head

linreg = scipy.stats.linregress(inv_pz_head[:-20:10], avg_flow_rate[:-2])
slope = linreg.slope
intercept = linreg.intercept
eff1 = (
    (p_water * g / config_data[:-20, 4])
    * (slope * inv_pz_head[:-20] + intercept)
    / inv_pz_head[:-20]
).mean()
eff1_err = (
    (p_water * g / config_data[:-20, 4])
    * (slope * inv_pz_head[:-20] + intercept)
    / inv_pz_head[:-20]
).std()

# Pipe Configs
plt.errorbar(
    inv_pz_head[:-20:10],
    avg_flow_rate[:-2],
    std_flow_rate[:-2],
    fmt="b.",
    label="Configs",
    capsize=3,
)
plt.plot(
    inv_pz_head[:-20],
    slope * inv_pz_head[:-20] + intercept,
    color="lightskyblue",
    label=f"Config\nSlope={slope:.0f}",
)
plt.xlabel("Inverse Pizeometric Head $(cm^{-1})$")
plt.ylabel("Flow Rate (ml/s)")
plt.legend()

# Different Hieghts
flow_rates = init_tube_data[:, 2] / init_tube_data[:, 1]
heights = init_tube_data[:, 0] - 3.5 * 2.54
avg_flow_rate = avg_flow_rate_h[:, 1]
std_flow_rate = std_flow_rate_h[:, 1]
liquid_head = np.ones(flow_rates.shape) * (6.9 - 3.5) * 2.54
```

```python
pz_head = heights + liquid_head
inv_pz_head = 1 / pz_head

linreg = scipy.stats.linregress(inv_pz_head[::10], avg_flow_rate)
slope = linreg.slope
intercept = linreg.intercept
eff2 = (
    (p_water * g / init_tube_data[:, 3])
    * (slope * inv_pz_head + intercept)
    / inv_pz_head
).mean()

plt.errorbar(
    inv_pz_head[::10],
    avg_flow_rate,
    std_flow_rate,
    fmt="rs",
    label="Heights",
    capsize=3,
)
plt.plot(
    inv_pz_head[:],
    slope * inv_pz_head[:] + intercept,
    color="lightcoral",
    label=f"Height\nSlope={slope:.0f}",
)
plt.xlabel("Inverse Pizeometric Head $(cm^{-1})$")
plt.ylabel("Flow Rate (ml/s)")
plt.title("Effect of Height and Configuration Change\non Flow Rate")
plt.legend()

plt.show()


print(eff1*1e-7)
print(eff1_err*1e-7)
print(eff2*1e-7)
```

In [ ]:

```python
v_water = 1.0034 * (1e-3)**2
flow_rates = config_data[:, 3] / config_data[:, 2]
velocity = (flow_rates * 1e-3) / (np.pi * 6e-3**2 / 4)
avg_velocity = np.array([velocity[i*10:i*10+10].mean() for i in range(6)])
reynolds = avg_velocity * 6e-3 / v_water
print("CONFIGS")
print(config_data[::10, 0])
print(reynolds)
print()

flow_rates = init_tube_data[:, 2] / init_tube_data[:, 1]
velocity = (flow_rates * 1e-3) / (np.pi * (0.51*0.0254)**2 / 4)
avg_velocity = np.array([velocity[i*10:i*10+10].mean() for i in range(5)])
reynolds = avg_velocity * (0.51*0.0254) / v_water
print("HEIGHTS")
print(init_tube_data[::10, 0])
print(reynolds)
print()
```

```
flow_rates = copper_tube_data[:, 3] / copper_tube_data[:, 2]
velocity = (flow_rates * 1e-3) / (np.pi * ((copper_tube_data[:, 2]*0.0254)**2 / 4))
avg_velocity = np.array([velocity[i*5:i*5+5].mean() for i in range(3)])
reynolds = avg_velocity * (copper_tube_data[::5, 2]*0.0254)/ v_water
print("LENGTHS")
print(reynolds)
```

In [ ]:
```
p_water = 997
mu_water = 0.89
short_copper = 20.5e-2
long_copper = 60.5e-2
pipe_D = 0.52 * 0.0254
pipe_e = 0.0015e-3
g = 9.81

flow_rate = (diff_height_data[:, 3] / diff_height_data[:, 2]) * 1e-3
print(flow_rate)
avg_flow_rate = np.array([flow_rate[i*10:i*10+10].mean() for i in range(4)])
print(avg_flow_rate)
std_flow_rate = np.array([flow_rate[i*10:i*10+10].std() for i in range(4)])

def colebrook_white(f, Re, D, epsilon):
    cw = -2 * np.log10(epsilon / (3.7 * D) + 2.51 / (Re * np.sqrt(f))) - 1 / np.sqr
    return cw

def fitting(k,v,g):
    loss = k * v**2/(2*g)
    return loss

i = 3
Q = avg_flow_rate[i]
Q_err = std_flow_rate[i]

vel = Q / (pipe_D**2 *np.pi / 4)
Re = p_water * pipe_D * vel / (mu_water)

ff_sol = scipy.optimize.root(colebrook_white, 0.01, args=(Re, pipe_D, pipe_e))
friction_factor = ff_sol.x
print(friction_factor)
```