

Homework 1

Kyle Wodehouse

September 5, 2024

1. Definitions (pages are for my own reference)

- (a) **closed system** – A system where mass cannot enter or leave the system. (pg. 3)
- (b) **open system** – A system where mass can enter or leave the system (cross system boundary).
- (c) **isolated system** – a system closed to both heat and energy; a change in the surroundings cannot change the system. (pg. 3)
- (d) **steady-state system** – a system where things may be flowing in or out, but the system properties do not change with time. (pg. 3)
- (e) **adiabatic system** – a system where there are no heat flows in or out; system is thermally isolated from surroundings. (pg. 3)
- (f) **state** – thermodynamic properties of a system as characterized by density, refractive index, composition, pressure, temperature, etc. (pg. 4)
- (g) **phase** – state of agglomeration of matter in a system. (pg. 4)
- (h) **equilibrium state** – state where there are no mass or energy flows in or out of the system, the rate of all chemical reactions is zero, all concentration gradients are zero, and the work between the system and the surroundings is zero. (pg. 8)
- (i) **absolute pressure** – pressure scale where zero is the lowest theoretically possible pressure. (pg. 12)
- (j) **temperature** – property where if two systems have the same temperature value, they are in thermal equilibrium. (pg. 12)
- (k) **ideal gas** – a gas at a density so low that intermolecular interactions are negligible. (pg. 13)
- (l) **thermometric property** – a property in which each value corresponds to a unique temperature. (pg. 14)
- (m) **heat** – transfer of energy as a result of only a temperature difference. (pg. 15)
- (n) **work** – energy transfer by any mechanism involving mechanical motion of, or across, the system boundary. (pg. 15)
- (o) **intensive variable** – properties independent of the size of the system (pressure, temperature). (pg. 19)
- (p) **extensive variable** – properties dependent on the size of the system (mass, volume, total energy). (pg. 19)
- (q) **energy** – The capacity to do work, or the capacity to raise the temperature of a substance.

homework 1; van der Waals

author: k.wodehouse

purpose: calculate molar volume of a gas using the van der Waals equation

2 a.i.

the program code is finding the zeros of the $f(V)$ equation to find the volume given all the other parameters. It's solving numerically because the equation is not easily solvable by hand and possibly impossible to isolate the volume with algebra.

$$\left(P + \frac{a}{V^2}\right)(V - b) = R \cdot T$$
$$f(V) = R \cdot T - \left(P + \frac{a}{V^2}\right)(V - b)$$

code from lobo's video:

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.optimize as opt

# constants for van der waals equation

a = 0.3634 # Pa m^6/mol^2
b = 4.236e-5 # m^3/mol

# other params

R = 8.314 # J/(mol K) ; ideal gas constant
P = 5e5 # Pa ; pressure
T = 300 # K ; temperature

# van der waals equation

def vdW(V):
    return (R * T) - ((P + a / V**2) * (V - b))

volume = opt.fsolve(vdW, R*T/P)[0]

print(f'van der Waals volume: {volume:.5f} mol') # m^3/mol
```

van der Waals volume: 0.00488 mol

Comparing van der Waals with the ideal gas

```
In [ ]: ideal = R * T / P
print(f'ideal gas volume: {ideal:.6f} mol') # m^3/mol

percent_diff = 100 * (ideal - volume) / volume
print(f'percent difference: {percent_diff:.2f}%')

ideal gas volume: 0.004988 mol
percent difference: 2.15%
```

2. a. ii.

first i'll setup another van der Waals function that also takes Temperature as an input

```
In [ ]: def vdWT(V, T):
    return (R * T) - ((P + a / V**2) * (V - b))

temperatures = np.array([150, 200, 250, 300, 350])
vdw_volumes = [opt.fsolve(vdWT, R*T/P, args=(T))[0] for T in temperatures]
ideals = R * temperatures / P
diff = abs(ideals - vdw_volumes)

df = pd.DataFrame({'T (K)': temperatures,
                  'V (m^3/mol)': vdw_volumes,
                  'Ideal Gas': ideals,
                  'Difference (m^3/mol)': diff})
df.set_index('T (K)', inplace=True)
df
```

```
Out[ ]:      V (m^3/mol)  Ideal Gas  Difference (m^3/mol)
T (K)
150      0.002215  0.002494      0.000280
200      0.003140  0.003326      0.000186
250      0.004020  0.004157      0.000137
300      0.004883  0.004988      0.000105
350      0.005736  0.005820      0.000083
```

results for the 150K are somewhat different (~1%) though the rest are quite similar. then as temperature rises the ideal gas equation gets much more similar to the van der Waals equation. i.e. as temperature increases the difference decreases.

homework 1; 2b

author: k.wodehouse

First I'll copy over the original plot for reference.

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt

tmax = 10 # Total length (in time) of the simulation

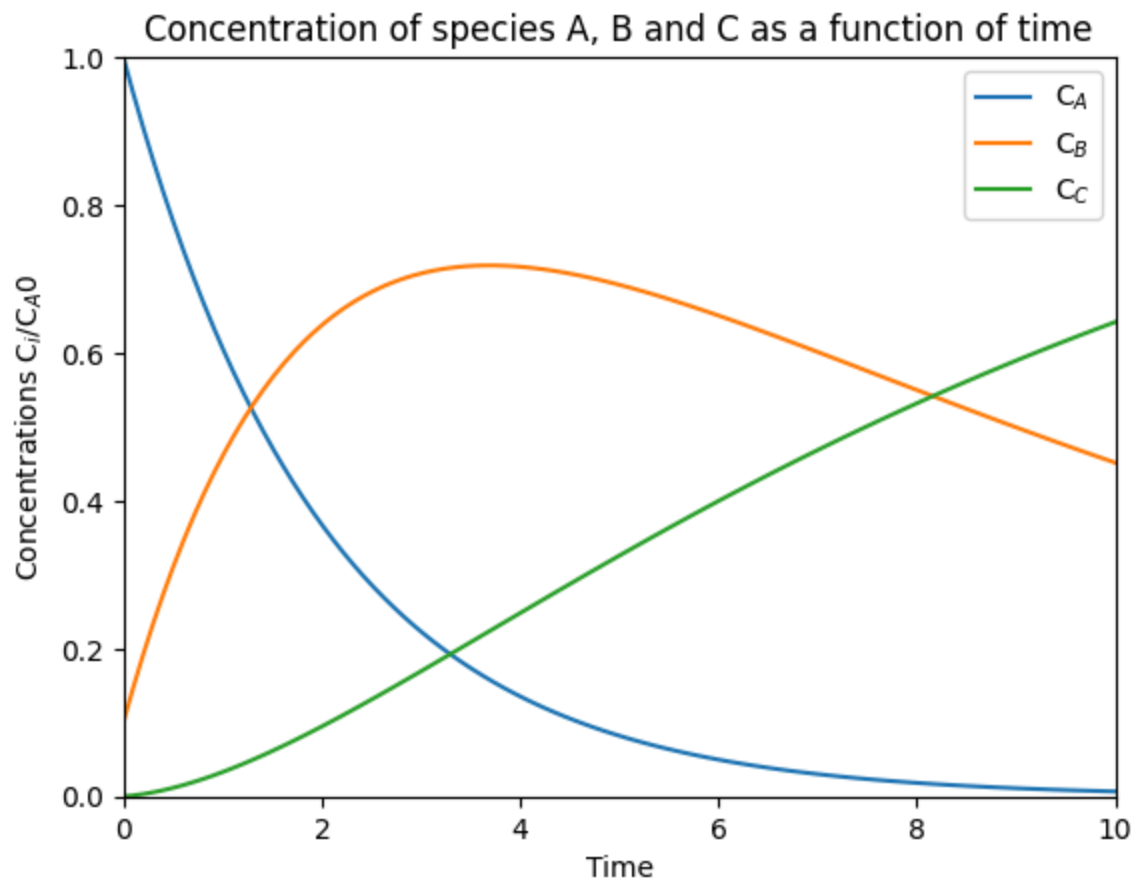
### Initial Conditions ###

CA0 = 1. # mol/L
CB0 = 0.1 # mol/L

k1 = 0.5 # Rate constants (units of inverse time)
k2 = 0.11
```

```
In [ ]: t= np.linspace(0,tmax,100)
CA = CA0 * np.exp(-k1 * t)
CB = k1*CA0 / (k2 - k1) * (np.exp(-k1*t) - np.exp(-k2*t)) + CB0 * np.exp(-k2*t)
CC = CA0 + CB0 - CA - CB
```

```
In [ ]: fig, ax = plt.subplots()
plt.title('Concentration of species A, B and C as a function of time')
ax.plot(t,CA, label='C$_A$')
ax.plot(t,CB, label='C$_B$')
ax.plot(t,CC, label='C$_C$')
ax.set(xlabel='Time',ylabel='Concentrations C$_i$/C$_{A0}$')
ax.legend(loc='best')
ax.set_xlim(t[0],t[-1])
ax.set_ylim(0,1)
plt.show()
```



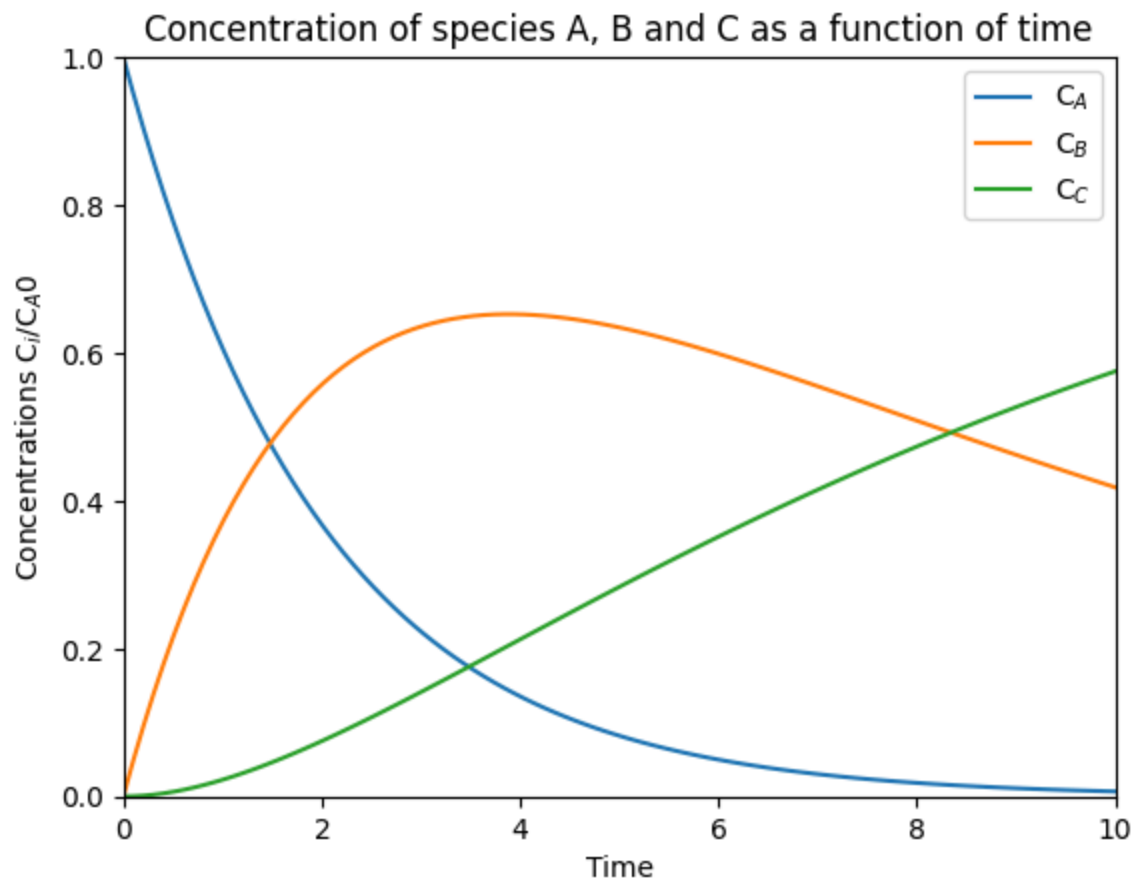
2 b. i.

Changing initial concentration of B to 0

```
In [ ]: CB0 = 0

CB = k1*CA0 / (k2 - k1) * (np.exp(-k1*t) - np.exp(-k2*t)) + CB0 * np.exp(-k2*t)
CC = CA0 + CB0 - CA - CB

fig, ax = plt.subplots()
plt.title('Concentration of species A, B and C as a function of time')
ax.plot(t, CA, label='C$_A$')
ax.plot(t, CB, label='C$_B$')
ax.plot(t, CC, label='C$_C$')
ax.set_xlabel='Time', ylabel='Concentrations C$_i$/C$_{A0}$')
ax.legend(loc='best')
ax.set_xlim(t[0], t[-1])
ax.set_ylim(0, 1)
plt.show()
```



Affected the plot by moving down the y-int of the orange curve (the C_B curve)

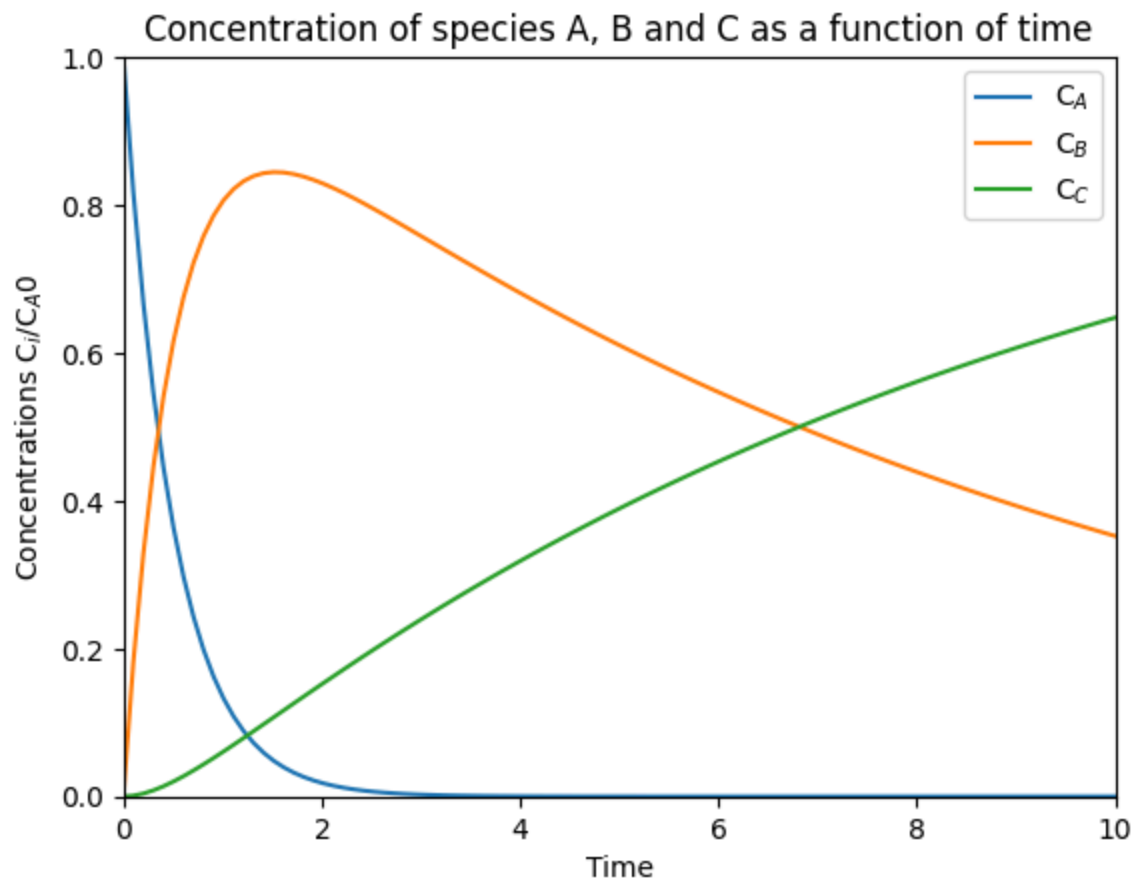
2 b. ii.

Increasing K_1

```
In [ ]: CB0 = 0
         k1 = 2

         t= np.linspace(0,tmax,100)
         CA = CA0 * np.exp(-k1 * t)
         CB = k1*CA0 / (k2 - k1) * (np.exp(-k1*t) - np.exp(-k2*t)) + CB0 * np.exp(-k2*t)
         CC = CA0 + CB0 - CA - CB

         fig, ax = plt.subplots()
         plt.title('Concentration of species A, B and C as a function of time')
         ax.plot(t,CA, label='C$_A$')
         ax.plot(t,CB, label='C$_B$')
         ax.plot(t,CC, label='C$_C$')
         ax.set(xlabel='Time',ylabel='Concentrations C$_i$/C$_A0$')
         ax.legend(loc='best')
         ax.set_xlim(t[0],t[-1])
         ax.set_ylim(0,1)
         plt.show()
```



increasing K_1 sped up the reaction significantly. This is because $A \rightarrow B$ faster, so the concentration of B increases much faster than before.

van der Waals pt. 2

author: k.wodehouse

```
In [ ]: import numpy as np
import scipy.optimize as opt
import matplotlib.pyplot as plt

a = 0.3634
b = 4.236e-5
R = 8.314
P = 5e5

def vdWT(V, T):
    return (R * T) - ((P + (a / V**2)) * (V - b))
```

question 2 c.i.

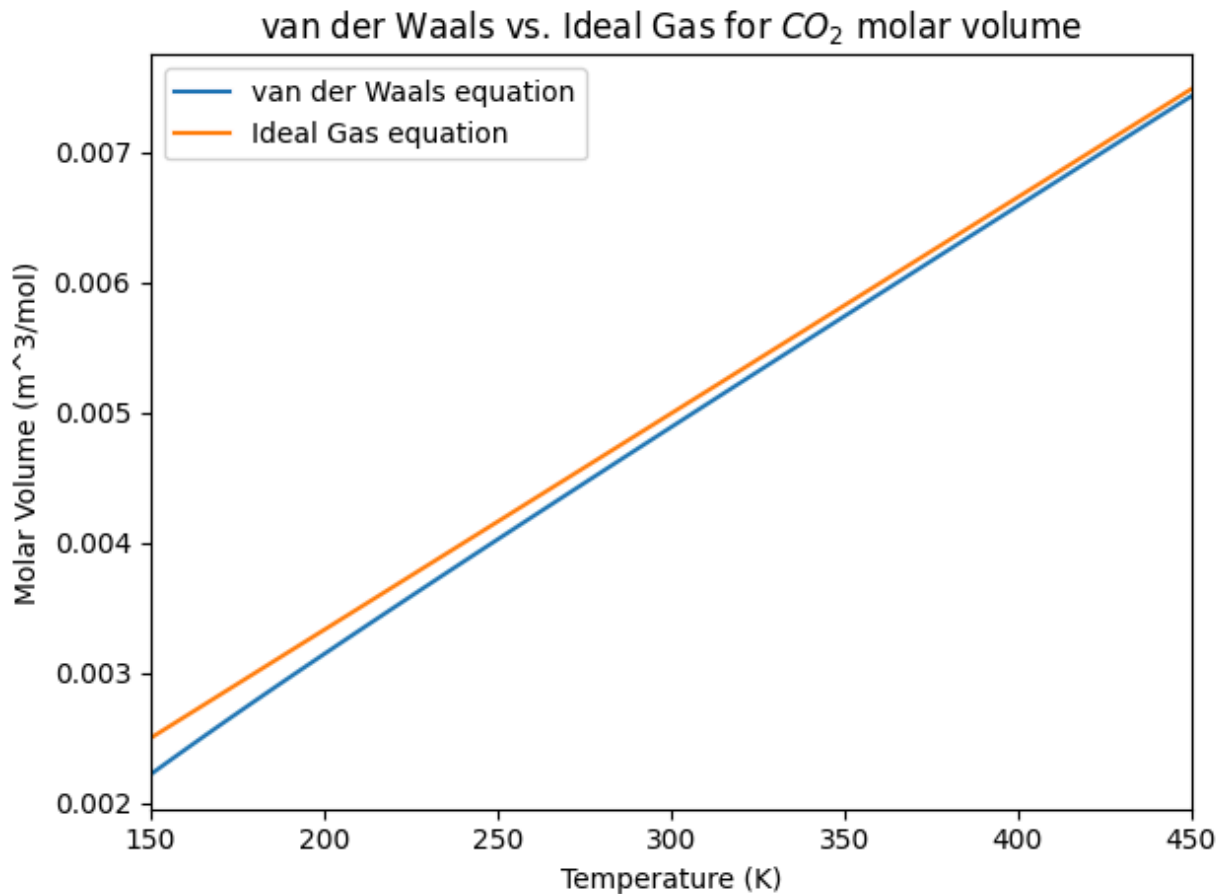
```
In [ ]: temperatures = np.linspace(150,450,100)

vdw_volumes = np.array([opt.fsolve(vdWT, R*T/P, args=(T))[0] for T in temperatures])

ideals = R * temperatures / P

plt.plot(temperatures, vdw_volumes, label='van der Waals equation')
plt.plot(temperatures, ideals, label='Ideal Gas equation')
plt.xlim(150,450)
plt.xlabel('Temperature (K)')
plt.ylabel('Molar Volume (m^3/mol)')
plt.title('van der Waals vs. Ideal Gas for $CO_2$ molar volume')
plt.tight_layout()
plt.legend()
```

```
Out[ ]: <matplotlib.legend.Legend at 0x15a623350>
```

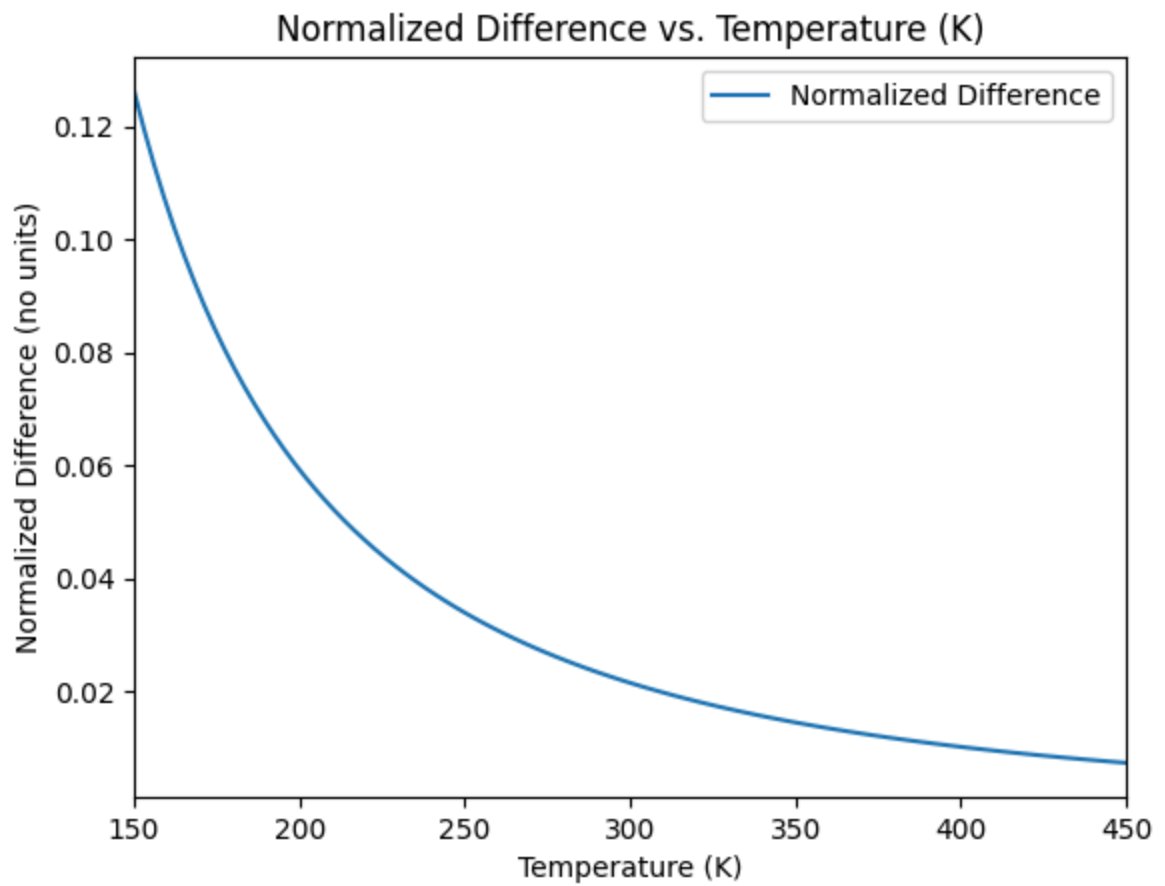
the trend is that as temperature increases so does molar volume for both ideal and van der Waals. the values are overall quite similar, especially in the second half of the temperature range.

2 c. ii. plotting normalized difference

```
In [ ]: normalized_difference = abs(ideals - vdw_volumes) / vdw_volumes

plt.plot(temperatures, normalized_difference, label='Normalized Difference')
plt.xlim(150,450)
plt.xlabel('Temperature (K)')
plt.ylabel('Normalized Difference (no units)')
plt.title('Normalized Difference vs. Temperature (K)')
plt.legend()
```

```
Out[ ]: <matplotlib.legend.Legend at 0x15a78ab10>
```



the trend is that as temperature increases the normalized difference decreases. this might be exponential or could also be proportional to reciprocal temperature.

Homework 1, Problem 3

author: k.wodehouse

I'll start off by bringing in all the data as a dataframe for ease

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = {
    'T (C)': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90],
    'Volume of 1 gram of H2O (cm3)': [1.0001329, 1.0000733, 1.0000321, 1.00000,
    'Volume of 1 gram of Hg (cm3)': [0.0735560, 0.0735694, 0.0735828, 0.073596]
}

df = pd.DataFrame(data).set_index('T (C)')
df
```

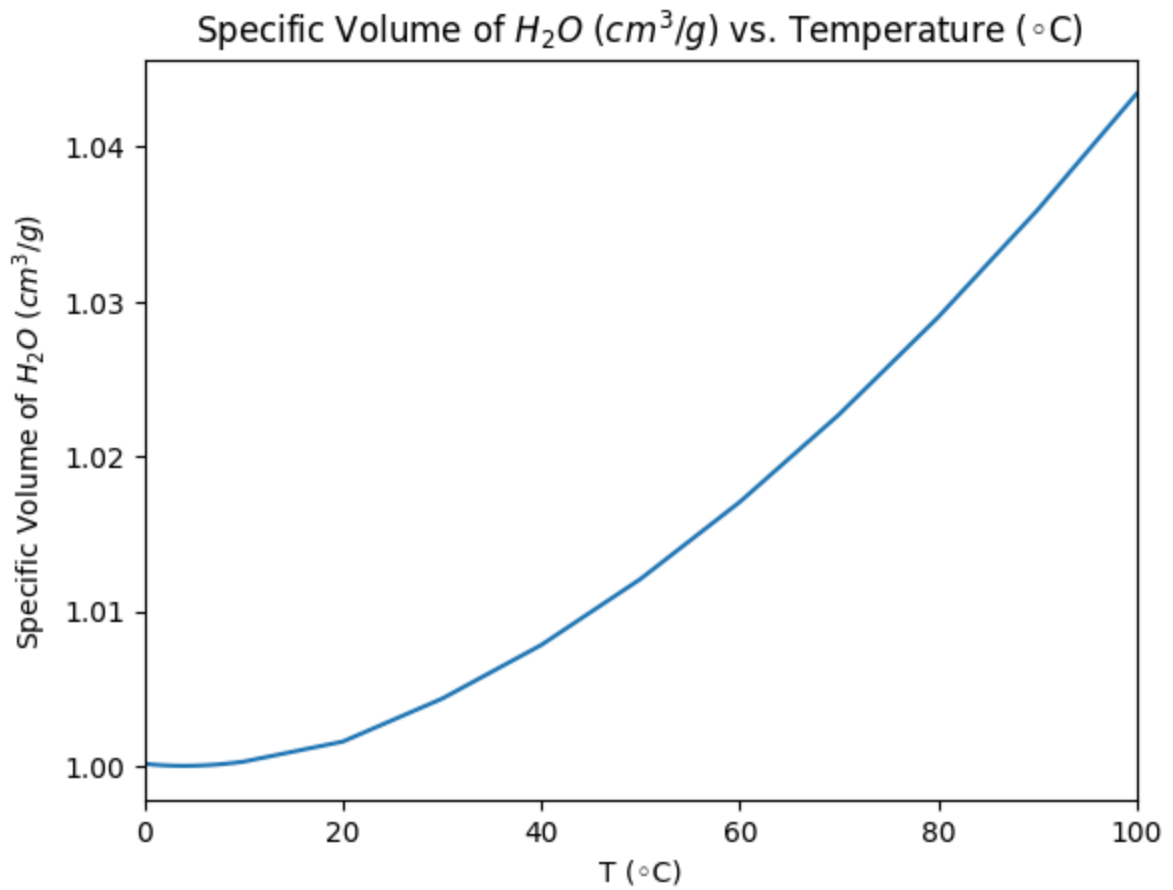
Out[]: Volume of 1 gram of H2O (cm3) Volume of 1 gram of Hg (cm3)

T (C)		
0	1.000133	0.073556
1	1.000073	0.073569
2	1.000032	0.073583
3	1.000008	0.073596
4	1.000000	0.073609
5	1.000008	0.073623
6	1.000032	0.073636
7	1.000070	0.073650
8	1.000124	0.073663
9	1.000191	0.073676
10	1.000272	0.073689
20	1.001568	0.073823
30	1.004341	0.073957
40	1.007811	0.074091
50	1.012074	0.074225
60	1.017046	0.074359
70	1.022694	0.074494
80	1.028987	0.074628
90	1.035904	0.074763
100	1.043427	0.074898

Graphing the specific volume of H_2O as a function of temperature

```
In [ ]: plt.plot(df['Volume of 1 gram of H2O (cm3)'], label='H2O')
plt.xlim(0, 100)
plt.xlabel('T (°C)')
plt.ylabel('Specific Volume of $H_{20}$ $(cm^3/g)$')
plt.title('Specific Volume of $H_{20}$ $(cm^3 / g)$ vs. Temperature (°C)')
```

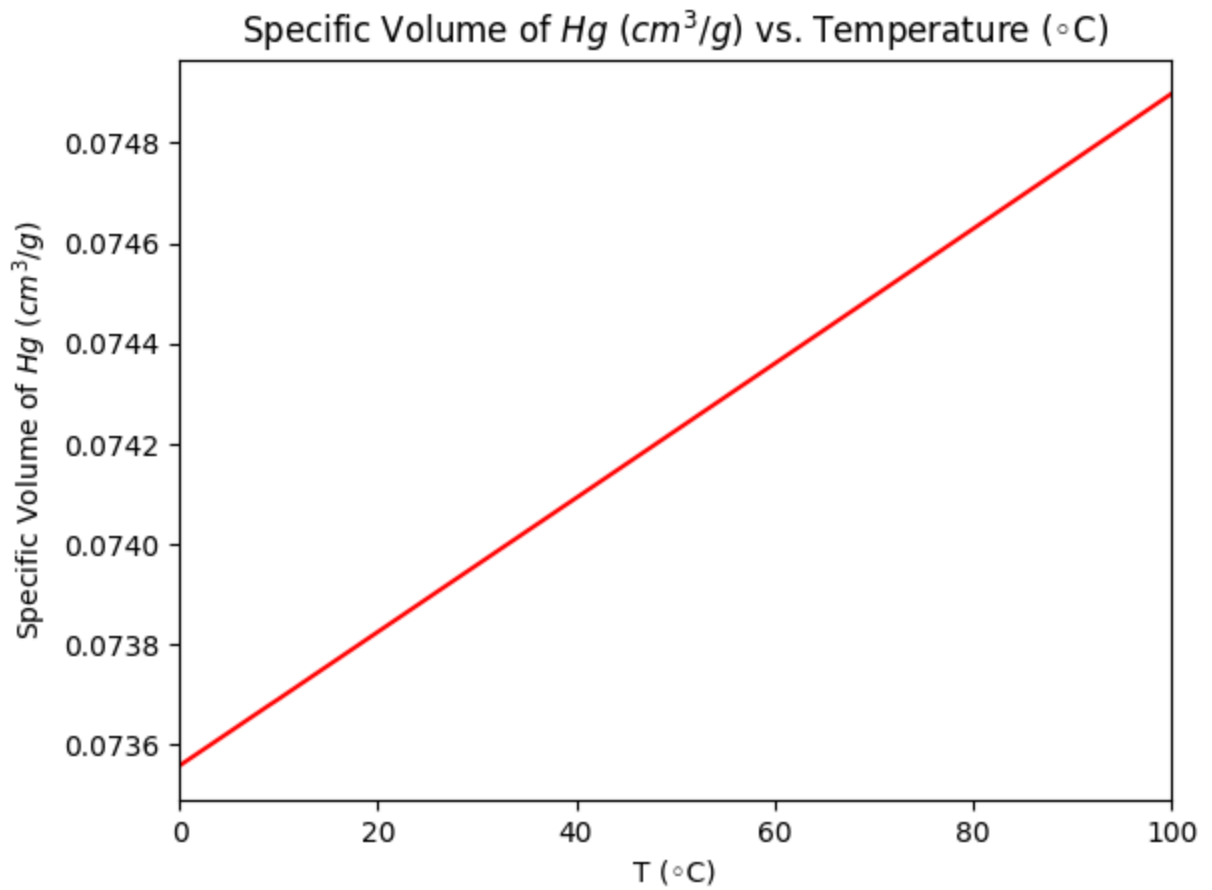
Out[]: Text(0.5, 1.0, 'Specific Volume of \$H_{20}\$ \$(cm^3 / g)\$ vs. Temperature (°C)')



doing the same for mercury

```
In [ ]: plt.plot(df['Volume of 1 gram of Hg (cm3)'], label='Hg', color='red')
plt.xlim(0, 100)
plt.xlabel('T ( $^{\circ}C$ )')
plt.ylabel('Specific Volume of $Hg$ ( $cm^3/g$ )$')
plt.title('Specific Volume of $Hg$ ( $cm^3 / g$ ) vs. Temperature ( $^{\circ}C$ )')
```

```
Out[ ]: Text(0.5, 1.0, 'Specific Volume of $Hg$ ( $cm^3 / g$ ) vs. Temperature ( $^{\circ}C$ )')
```

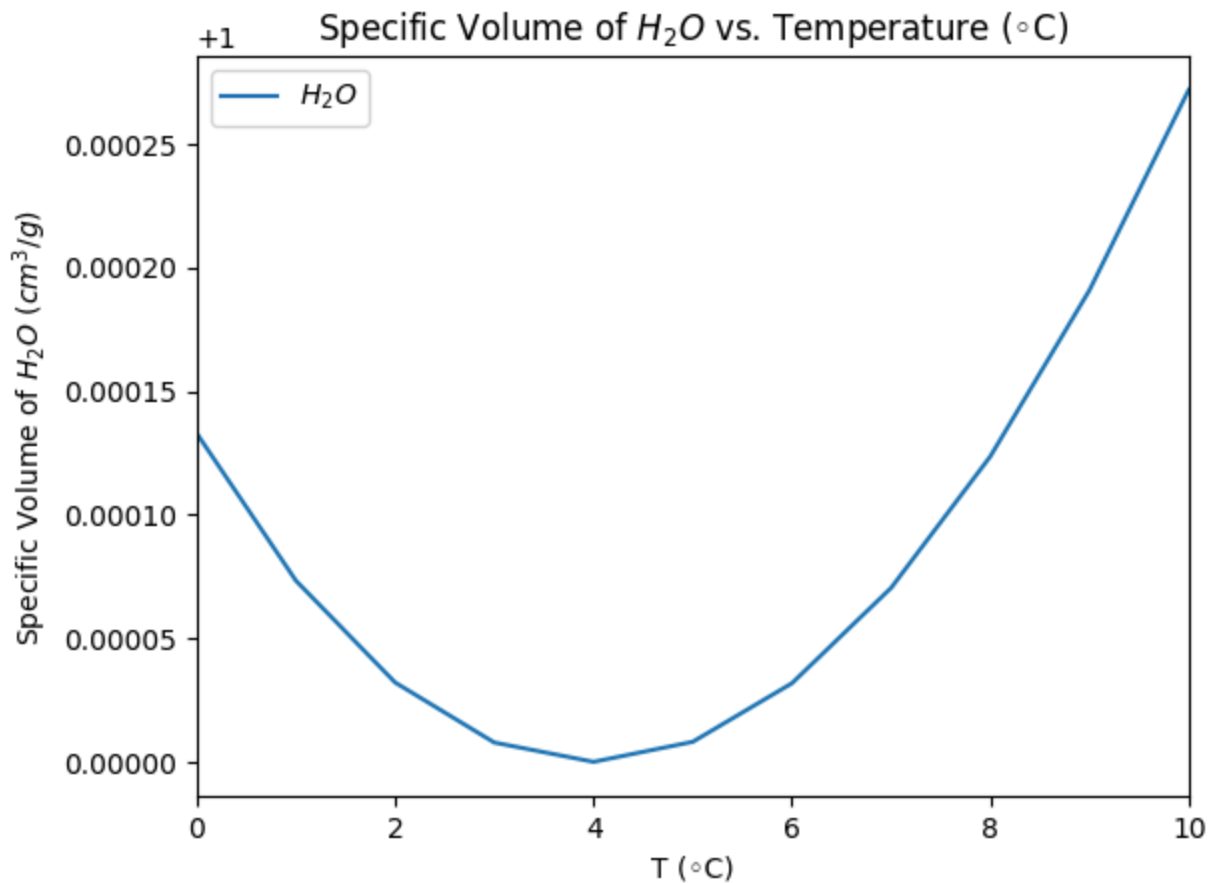


a

visualizing the specific volume of water from $T=0^{\circ}C$ to $T=10^{\circ}C$

```
In [ ]: plt.plot(df['Volume of 1 gram of H2O (cm3)'][:11], label='$H_2O$')
plt.xlim(0, 10)
plt.xlabel('T (°C)')
plt.ylabel('Specific Volume of $H_2O$ $(cm^3/g)$')
plt.title('Specific Volume of $H_2O$ vs. Temperature (°C)')
plt.legend()
```

```
Out[ ]: <matplotlib.legend.Legend at 0x13abd8a10>
```



water wouldn't be very good on this range of temperatures since it's hard to tell 0-2 apart from 6-8 and also water as a whole (from the earlier visualization) is quite nonlinear. since it's not strictly increasing/decreasing the temperature could be rising but the thermometer level could be changing directions. this is not what you want a thermometer to do!

b

first simulate one that's only calibrated at 0°C and 100°C

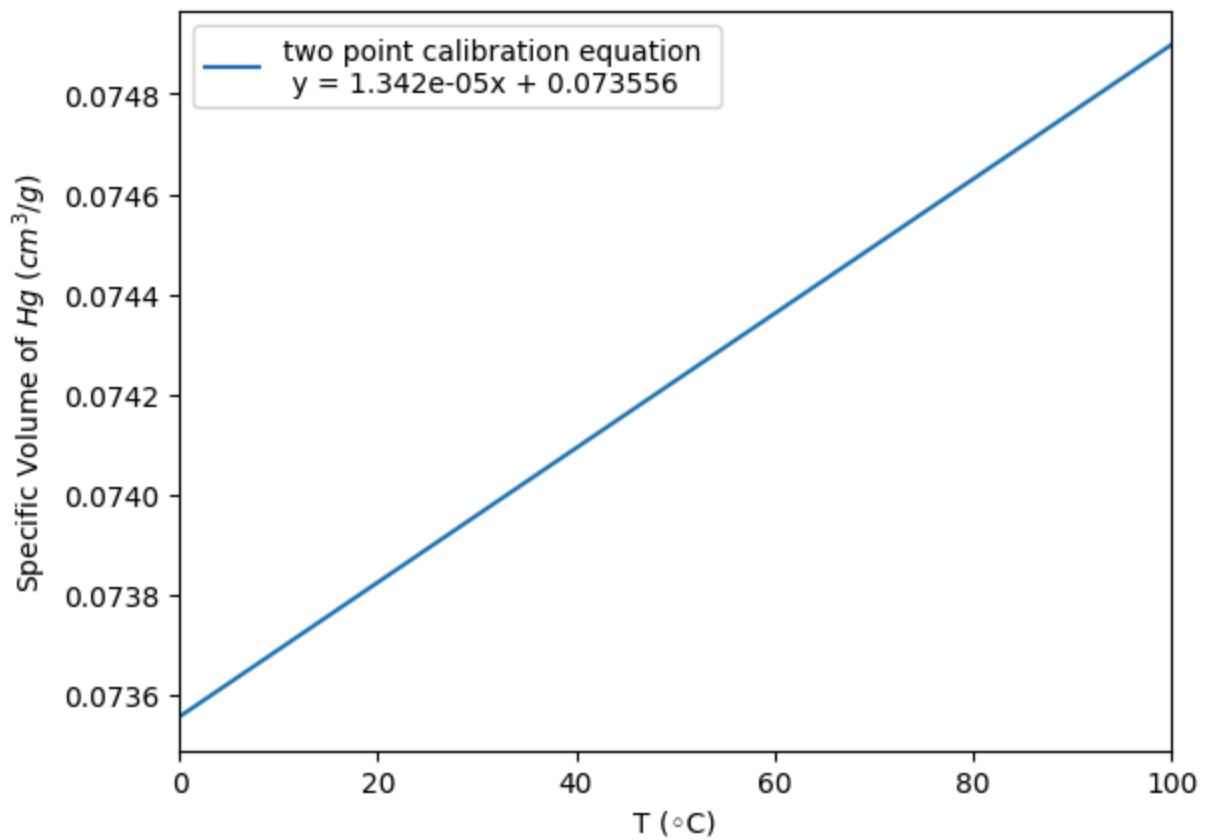
```
In [ ]: volumes = [df['Volume of 1 gram of Hg (cm3)'][0], df['Volume of 1 gram of Hg (cm3)'][100]]
temperatures = [0, 100]

m, b = np.polyfit(temperatures, volumes, 1)
print(f'equation: y = {m:.3e}x + {b:.6f}')

plt.plot(temperatures, m*np.array(temperatures) + b, label=f'two point calibration')
plt.xlim(0, 100)
plt.xlabel('T (°C)')
plt.ylabel('Specific Volume of $Hg$ $(\text{cm}^3/\text{g})$')
plt.legend()
```

```
equation: y = 1.342e-05x + 0.073556
```

```
Out[ ]: <matplotlib.legend.Legend at 0x13ad3ec90>
```



The two point calibration gives us

$$\hat{V} = 1.342 \cdot 10^{-5} \cdot T + 0.073556$$

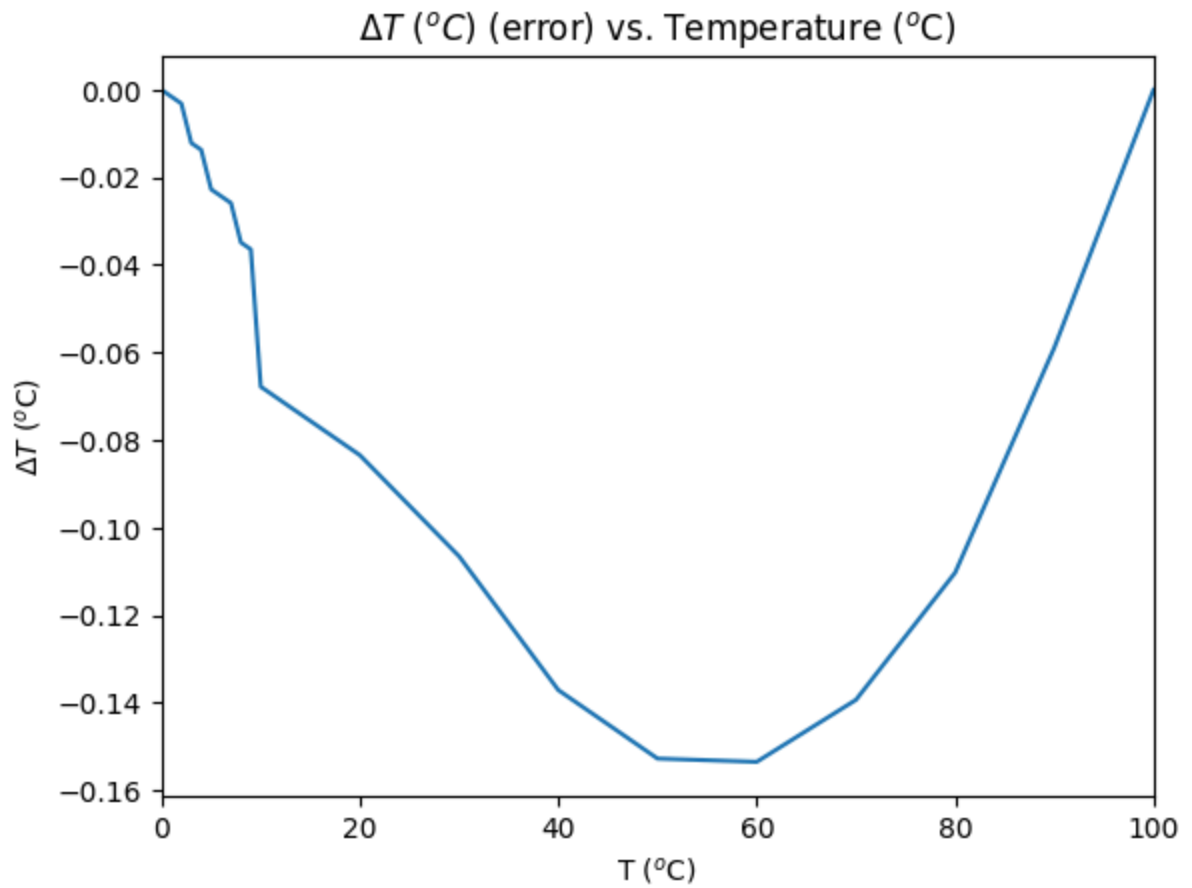
Which means

$$T_{meas} = \frac{\hat{V} - 0.073556}{1.3402 \cdot 10^{-5}}$$

so we go through each \bar{V} in the table and calculate what temp that correlates to on the two point calibration, subtract the actual temperature from it, and then put together a nice little graph

```
In [ ]: volumes = df['Volume of 1 gram of Hg (cm3)']
temperatures = df.index
measured_temps = (volumes - b) / m
delta_T = measured_temps - temperatures

plt.plot(temperatures, delta_T, label='Measured Temperatures')
plt.xlim(0, 100)
plt.xlabel('T (°C)')
plt.ylabel('$\Delta T$ (°C)')
plt.title('$\Delta T$ (°C) (error) vs. Temperature (°C)');
```

C

The change in height of the mercury is proportional to how much the volume increases and inversely proportional to the cross sectional area of the tube/capillary. To put this in math:

$$\Delta h_{\text{mercury}} = \frac{\Delta V}{A}$$

So if you want to maximize the thermometers response to changes in temperatures you'd minimize the A , which is the purpose of using the capillary tube, and maximize the ΔV , which is the purpose of the large bulb since having more mass of Hg increases the ΔV

4. Math Practice

a.

$$\begin{aligned}
 \int_{V_1}^{V_2} P \, dV &= RT \int_{V_1}^{V_2} \frac{1}{V} \, dV \\
 &= RT [\ln(V)]_{V_1}^{V_2} \\
 &= RT [\ln(V_2) - \ln(V_1)] \\
 &= RT \ln\left(\frac{V_2}{V_1}\right)
 \end{aligned}$$

b.

First we can describe volume as a function of x where x represents the distance compressed, L_1 represents the length of the cylinders, and A represents the cross-sectional area of the cylinder

$$V = A(L_1 - x)$$

Then taking the derivative with respect to x on each side, we get

$$\frac{dV}{dx} = -A$$

Which yields the relationship

$$\frac{-dV}{A} = dx$$

Returning to the work definition we can change to dV and apply the relationship $P = \frac{F}{A}$

$$W = \int_{x_1}^{x_2} F \, dx = - \int_{x_1}^{x_2} \frac{F}{A} \, dV = - \int_{x_1}^{x_2} P \, dV$$

Now to adjust the bounds, we can use the relationship between x and V . When $x = x_1$, $V = A(L_1 + 0) = V_1$ and when $x = x_2$, $V = A(L_1 - (x - x_1)) = V_2$. So we can substitute these values in for the bounds of the integral. We can also pull pressure out because this form of work is done against/at a constant pressure.

$$W = - \int_{V_1}^{V_2} P \, dV = -P \int_{V_1}^{V_2} dV = -P(V_2 - V_1)$$