Lab 2: Joule-Thomson Coefficient

Max Huggins – UCA Department of Chemistry

February 9, 2021

Abstract

The Joule-Thomson Coefficients for CO_2, N_2, and He were found by plotting temperature as a function of pressure. This was done by pressurizing a vessel with each gas and varying the pressure while measuring the temperature. A linear regression was used to then determine the slopes of each function and therefore determine the coefficients. The values found were (respectively): (1.11, 0.801, -0.058)K/atm corresponding to percent errors of (0.31, 196.63, 6.23)%.

1 Procedure

1.1 Overview

The procedure for this lab was adapted from [1]. The procedure is straightforward, but caution must be taken when connecting fittings and transporting the various gasses.

1.2 Details

A gas tank was connected to an apparatus similar to the one described in the original source. The variation of pressure was allowed by the tank's regulator and the collection of temperature data was taken from a type K thermocouple attached to a lab bench multimeter. It was given that the thermocouple's conversion is $39\mu V$ per degree celsius. The pressure range typically varied from 20-80psi in increments of 5psi.

2 Results and Analysis

All of my analysis was completed in Python using open-source software. Because of this I will not go too much into depth about the methods used and the prescriptions applied for each I will focus mainly on the data and the results. All of my scripting is attached in the appendix and I have tried to make my variable naming such that it is readable. When compared to the values given by [1], it can be seen that the nitrogen curve has had problems. I am unsure of the problem here, but there are many factors that could have led to this. The first is that the electrical noise the multimeter picked up was magnitudes larger than the desired data values. Also, the CO_2 gas was done before N_2 so it is possible the lines still had some CO_2 gas in them, increasing the coefficient. This would also explain the linearity still shown in the plot despite the massive error.

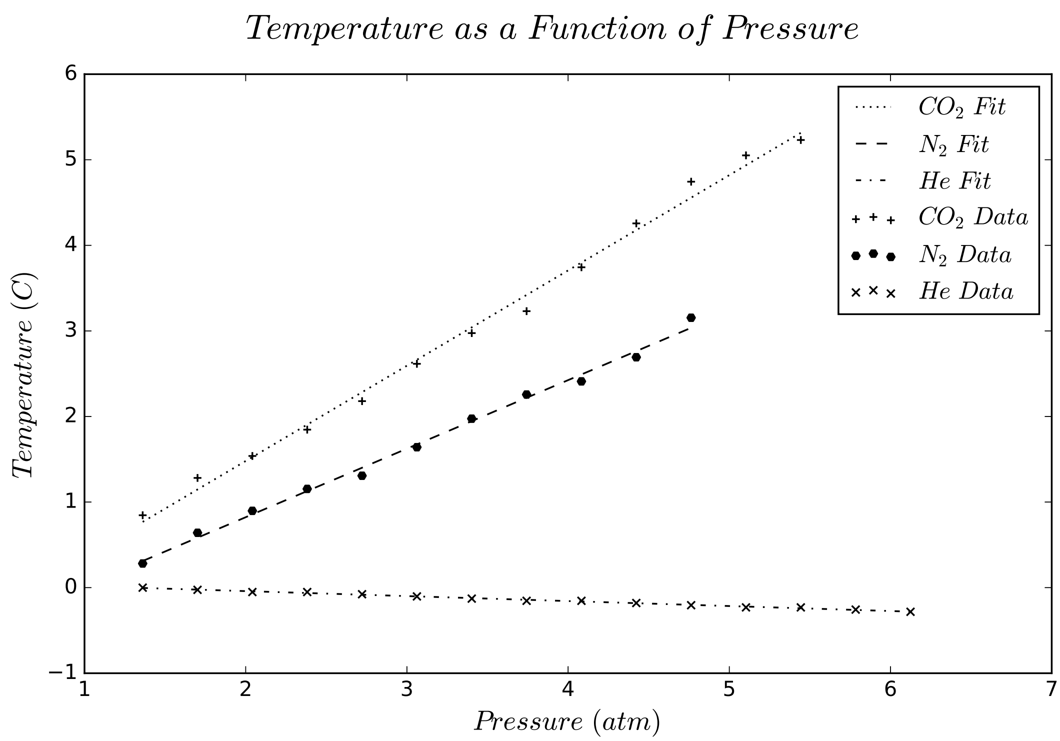

Figure 1: Plot of the temperature at various pressures for each gas.

3 Conclusion

This experiment had large uncertainties associated with it. Since much of this error was electrical noise, it is not easily characterized and it not accounted for in an error analysis here. However, with multiple data points, some values resembling the expected literature values were determined. Ways to improve on this setup are mostly ones that minimize electrical noise of the apparatus. The entire setup was essentially connected to one large antenna and the thermocouple producing peak data voltages on the scale of microvolts was mostly unreadable with all of the electromagnetic noise being introduced to the data line. A proper voltage amplifier and oscilloscope to log data would be more suitable for this experiment. This would allow for not only better electrical connections, but averages for voltage output of each pressure to be determined; further reducing errors. This in turn would also allow for error analysis to be done in a systematic way as reliance on the tolerance of electrical components is highly regulated and easily acquired.

4 Appendix

Here I have included the script written for this lab. I have also included the output of the code in the form of a photo from the kernel.

```python
# -*- coding: utf-8 -*-
"""
Created on Tue Feb 18 19:56:56 2020

@author: maxhu
"""
#=========================Importing Modules========================================#
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
plt.style.use('classic')
#=========================Defining Constants=======================================#
"""
39 uV/C
Where data is in mV
=======> 39e-3 V/C
"""
VoltageToCelsiusConversion = 39e-3
PsiToAtm = 0.068046
LiteratureValues = [1.11, .27, -.062] #CO2, N2, He respectively\
Gasses = ['CO2', 'N2', 'He']
#============================Importing Data========================================#
data = pd.read_excel (
         r'C:\Users\maxhu\Documents\PChem\PchemII\M1\Joule_Thompson\M1_Joule_thompson_Data.xlsx'
         , sheet_name='CO2')

df = pd.DataFrame(data, columns= ['psi'])
hold = df.values.tolist()
CO2Pressure  = np.array([val for sublist in hold for val in sublist])
CO2Pressure = CO2Pressure * PsiToAtm

df = pd.DataFrame(data, columns= ['mV'])
hold = df.values.tolist()
CO2Voltage  = np.array([val for sublist in hold for val in sublist])
CO2Temperature = CO2Voltage / VoltageToCelsiusConversion
#==================================================================#
data = pd.read_excel (
         r'C:\Users\maxhu\Documents\PChem\PchemII\M1\Joule_Thompson\M1_Joule_thompson_Data.xlsx'
         , sheet_name='N2')

df = pd.DataFrame(data, columns= ['psi'])
hold = df.values.tolist()
N2Pressure  = np.array([val for sublist in hold for val in sublist])
N2Pressure = N2Pressure * PsiToAtm

df = pd.DataFrame(data, columns= ['mV'])
```

```python
hold = df.values.tolist()
N2Voltage  = np.array([val for sublist in hold for val in sublist])
N2Temperature = N2Voltage / VoltageToCelsiusConversion
#==================================================================#
data = pd.read_excel (
        r'C:\Users\maxhu\Documents\PChem\PchemII\M1\Joule_Thompson\M1_Joule_thompson_Data.xlsx'
        , sheet_name='He')

df = pd.DataFrame(data, columns= ['psi'])
hold = df.values.tolist()
HePressure  = np.array([val for sublist in hold for val in sublist])
HePressure = HePressure * PsiToAtm

df = pd.DataFrame(data, columns= ['mV'])
hold = df.values.tolist()
HeVoltage  = np.array([val for sublist in hold for val in sublist])
HeTemperature = HeVoltage / VoltageToCelsiusConversion
#=====================My Linear Regression Function==============================#
def LinearRegression(x,y, GasName):
    m, b, r_value, p_value, std_err = stats.linregress(
            x,y)
    slope=round(m,6)
    if GasName == 'CO2':
        error = round((abs(LiteratureValues[0] - m) / LiteratureValues[0]) * 100, 2)
    if GasName == 'N2':
        error = round((abs(LiteratureValues[1] - m) / LiteratureValues[1]) * 100, 2)
    if GasName == 'He':
        error = round(abs((LiteratureValues[2] - m) / LiteratureValues[2]) * 100, 2)
    print('The slope of {} is {} K/atm with an error of {}%'.format(
            GasName, slope, error))
    return m*x + b
#==============================Plotting==========================================#
size = 20
size_config = .8
fig = plt.figure(1,figsize=(10,6))
my_fig = fig.add_subplot(111)
fig.suptitle('$Temperature\ as\ a\ Function\ of\ Pressure$', fontsize=size)
plt.xlabel('$Pressure\ (atm)$', fontsize=size_config*size)
plt.ylabel('$Temperature\ (C)$', fontsize=size_config*size)
#==================================================================#
plt.plot(CO2Pressure, LinearRegression(CO2Pressure,-CO2Temperature, 'CO2'),
         color='black', label='$CO_2\ Fit$', linestyle='dotted')
plt.scatter(CO2Pressure,-CO2Temperature,color='black', label='$CO_2\ Data$',
            marker ='+')
#==================================================================#
plt.plot(N2Pressure, LinearRegression(N2Pressure,-N2Temperature, 'N2'),
         color='black', label='$N_2\ Fit$', linestyle='dashed')
plt.scatter(N2Pressure,-N2Temperature,color='black', label='$N_2\ Data$',
            marker ='H')
#==================================================================#
```

```
plt.plot(HePressure, LinearRegression(HePressure,-HeTemperature, 'He'),
         color='black', label='$He\ Fit$', linestyle='dashdot')
plt.scatter(HePressure,-HeTemperature,color='black', label='$He\ Data$',
            marker ='x')
#==========================================================#
plt.legend(loc='best')
plt.savefig('Joule_Thompson.eps')
#============================================================================#
```

```
The slope of CO2 is 1.113479 K/atm with an error of 0.31%
The slope of N2 is 0.800912 K/atm with an error of 196.63%
The slope of He is -0.058138 K/atm with an error of 6.23%
```

Figure 2: Output for the python script.

References

[1] Arthur M Halpern and Saeed Gozashti. An improved apparatus for the measurement of the joule-thomson coefficient of gases. *Journal of Chemical Education*, 63(11):1001, 1986.