Ryan Herrmann

Lab 2

BMED 430-03

Introduction

The purpose of this lab was to use temperature and viscosity changes in water and use interpolation to estimate the viscosity of water at specific temperatures. The lab was also to investigate the different methods in python to do the operation. This included linear interpolation, spline interpolation, and curve-fitting to find the values of water.

Numerical Method

The methods we used were for loops for the creation of the tables as well as a definition function for the curve fit. The numerical method was an interpolation in python as well as numpy functions for curve fits. The output would be figures from scipy and the tables would be from a dictionary in pandas.

Pseudo-Code

Import numpy as np

Import matplotlib.pyplot as plt

From scipy.interpolate import CubicSpline

From scipy.optimize import curve_fit

Import pandas as pd

#define constants

sigfigs = 4

Temp = [12.0,24.0,38.0,64.0] #target temperatures

#read data from excel

excel_df = pd.read_excel('Lab2/viscosity_data.xlsx', sheet_name = 'data_file')

#copy data into arrays

tempD = excel_df.iloc[:,0].values

visd = excel_df.iloc[:,1].values

#perfrom interpolations

- Standard interpolation
- Cubic spline interpolation

#Curve fit

- Define curve fit function

```
def fit_func(x,a,b):
    fit=a*np.log(x) + b
    return fit
```

#Set up for plots

- Establish x data for plotting via linspace
- Cubic spline to create y data
- Loop through data to initialize lists for plotting
  - Count
  - Viscosity interpolation
  - Viscosity through cubic spline
  - Viscosity through curve-fit
  - List of given data

#Create plots

- Subplot of three plots
  - Title
  - X-axis
  - Y-axis
  - Plot data
- Save final figure
- Set up Data Frames for csv file
- Create csv file

Output

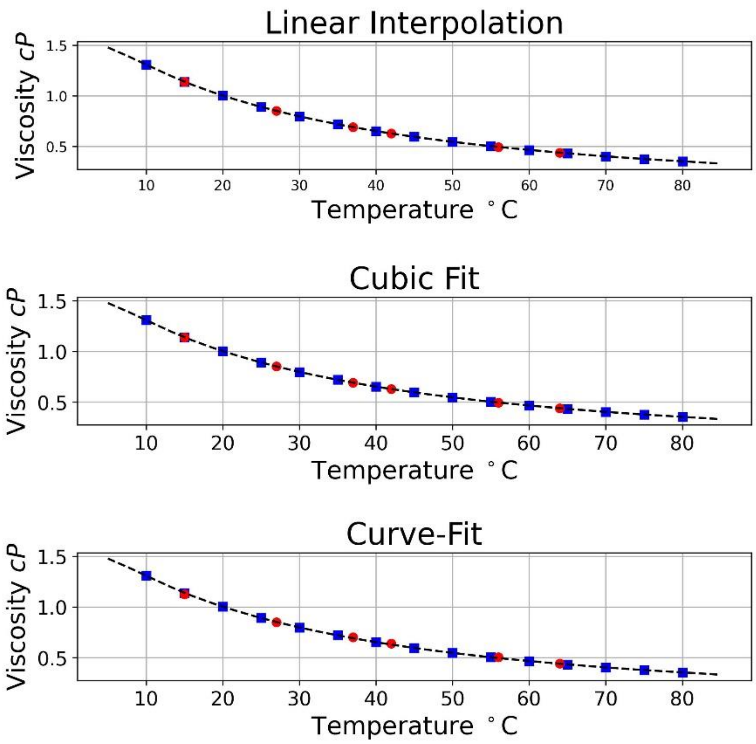

Figure 1: Subplot of interpolation graphs for viscosity through linear interpolation, cubic fit, and curve fit

Table 1: Results of viscosities at specific temperatures for the three methods in python including the exact numbers.

Temperature ˚C	Linear (cP)	Cubic Spline (cP)	Curve Fit (cP)	Exact (cP)
15	1.139	1.139	1.126	1.139
27	0.8538	0.8517	0.8486	0.852
37	0.6932	0.692	0.7002	0.692
42	0.6302	0.6291	0.6405	0.629
56	0.4966	0.4962	0.505	0.496
64	0.4406	0.4404	0.4421	0.44

Discussion

The functions for the viscosities worked. The linear interpolation and the cubic spline were both more accurate than the curve fit, but the curve fit was still a good estimation. This lab showed that python works for basing interpolations as well as using the integrated methods in python to estimate values.

Appendix

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

```
from scipy.interpolate import CubicSpline
from scipy.optimize import curve_fit
import pandas as pd

#
# Read in excel_file
#
excel_df = pd.read_excel('Lab2/viscosity_data.xlsx', sheet_name ='data_file')
#excel_df

tempD = excel_df.iloc[:,0].values
visd = excel_df.iloc[:,1].values

print(tempD)
print(visd)
#target temps
temps = [15.0,27.0,37.0,42.0,56.0,64.0] #deg C
L_exact = [1.139,0.852,0.692,0.629,0.496,0.440] #cP

sigfigs = 4

#linear interpolation
vis1_temp = np.interp(temps,tempD,visd)
vis1 = vis1_temp*1000.0
print(vis1)
print(L_exact)

#Spline
f=CubicSpline(tempD,visd,bc_type='natural')
vis2=f(temps)*1000.0
print(vis2)

#Curvefit
def fit_func(x,a,b):
    fit=a*np.log(x) + b
    return fit

#results from curvefitting
params = curve_fit(fit_func,tempD,visd)
a_fit = params[0][0]
b_fit = params[0][1]

#load lists
L_vis1 = [] #linear interpolation
L_vis2 = [] #cubic spine
```

```
L_vis3 = [] #curve fit
L_vis3r = [] #curve fit unformatted
L_temp = [] #temperature formatted

for i in temps:
    v1 = np.interp(i,tempD,visd)*1000.0
    v2 = f(i)*1000.0
    v3 = fit_func(i,a_fit,b_fit)*1000.0

    L_vis1.append('%.*g' % (sigfigs,v1))
    L_vis2.append('%.*g' % (sigfigs,v2))
    L_vis3.append('%.*g' % (sigfigs,v3))
    L_temp.append('%.*g' % (sigfigs,i))
    L_vis3r.append(v3) #for plotting curve-fit results

vis_dictionary = {'Temperature $^\\circ$%C':L_temp, 'Linear $cP':L_vis1, 'Cubic
Spline $cP': L_vis2, 'Curve Fit $cP': L_vis3, 'Exact $cP':L_exact}
print(vis_dictionary)
df1 = pd.DataFrame(vis_dictionary)
df1.set_index('Temperature $^\\circ$%C',inplace=True)
#df1

xnew = np.linspace(5,85,30)
ynew = f(xnew)*1000.0

#New plot lines
fig = plt.figure(figsize=(10,8))
plt.subplot(3,1,1)
plt.rcParams.update({'font.size':14})
plt.plot(tempD,visd*1000,'bs')
plt.plot(temps,vis1,'ro')
plt.plot(xnew,ynew,'--k')
plt.title("Linear Interpolation",fontsize = 22)
plt.ylabel('Viscosity $cP$',fontsize = 18)
plt.xlabel('Temperature $^\\circ$C', fontsize = 18)
plt.grid(True)


plt.subplot(3,1,2)
plt.rcParams.update({'font.size':14})
plt.plot(tempD,visd*1000,'bs')
plt.plot(temps,vis2,'ro')
plt.plot(xnew,ynew,'--k')
plt.title("Cubic Fit",fontsize = 22)
plt.ylabel('Viscosity $cP$',fontsize = 18)
```

```
plt.xlabel('Temperature $^\\circ$C', fontsize = 18)
plt.grid(True)


plt.subplot(3,1,3)
plt.rcParams.update({'font.size':14})
plt.plot(tempD,visd*1000,'bs')
plt.plot(temps,L_vis3r,'ro')
plt.plot(xnew,ynew,'--k')
plt.title("Curve-Fit",fontsize = 22)
plt.ylabel('Viscosity $cP$',fontsize = 18)
plt.xlabel('Temperature $^\\circ$C', fontsize = 18)
plt.grid(True)

plt.subplots_adjust(bottom = 0.1, right = 0.8, top = 0.9, hspace = 1)

fig.savefig('Lab2/interpolationGraphs.jpeg',dpi = 300,bbox_inches = 'tight')

L_dfs = [excel_df,df1]
with open('Lab2/interpolationCSV.csv','w',newline='') as f:
    for df in L_dfs:
        df.to_csv(f)
        f.write("\n")
```