Ryan Herrmann

Lab 6a

BMED 430

Introduction

The purpose of this lab was to use python to solve for the 1D heat equation for temperature of a bar that is insulated and exposed on both ends. The first part of the lab was solved via linsolve built in function.

Numerical Methods

The numerical method used was the built in linsolver.

Pseudo Code

- Import required packages
- Define constants and input data
    - Sigfigs
    - Length of rod
    - Points on the rod
- Create the matrix that will define the points on the rod
- Use numpy.linalg.solve
- Export to dataframe
- Use df to csv
- Plot the resulting graph
- Save fig

Output

The graph for the heat versus length is shown in Figure 1.

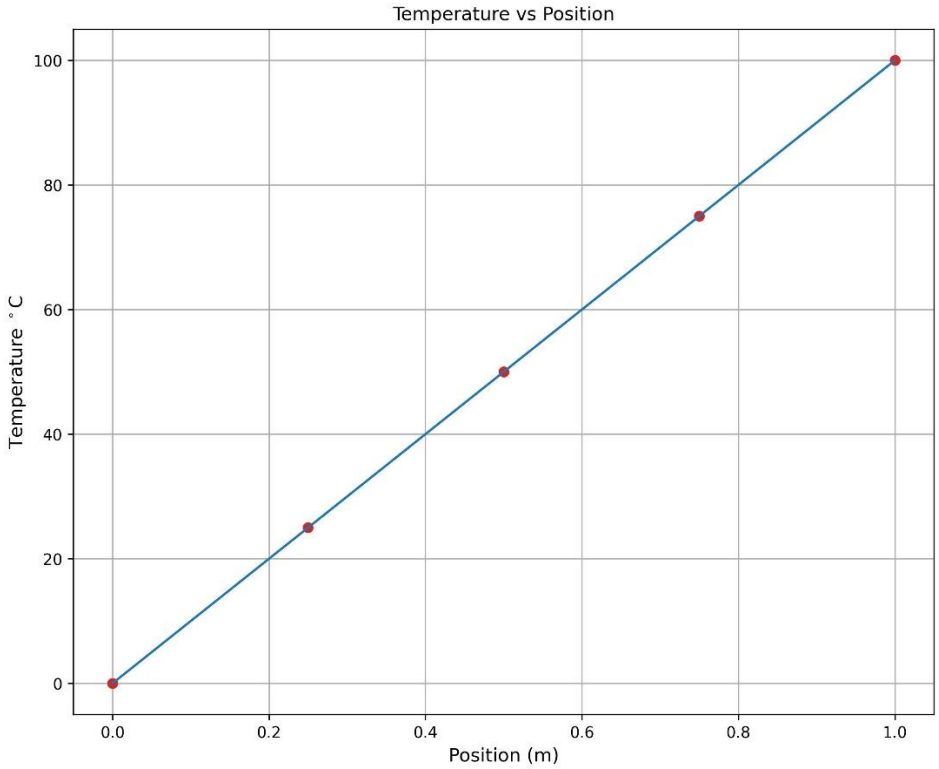

Figure 1: Temperature of the rod based on position of the points: the graph of the rod temp when the side was exposed
Table 1 shows the points and the temp at the time.

Table 1: Position on the rod and temperature associated with it

Position (m)	Temperature (C)
0	0
0.25	25
0.5	50
0.75	75
1	100

Discussion

This proved that the built in solver for python works, and the linsolve can be used to solve for heat linearly.

Appendix

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

```
#setup m matrix finite diff metrix
#setup c matrix solution matrix
L = 1 #m
sigfigs = 4
n = 3
n1 = n+1
n2 = n1 + 1
dx = L/n1

mMat = np.zeros((n2,n2))
cMat = np.zeros((n2))

mMat[n1,n1] = 1
mMat[0,0] = 1
cMat[-1] = 100.00

LoHS = 0.0
L_xp = [0]
L_xpf = ['%.*g'%(sigfigs,LoHS)]

#append format so that I dont have to keep writing the same thing over and over
def L_xpfAppend(n):
    L_xpf.append('%.*g' % (sigfigs,n))

for i in range(1,n1):
    L_xp.append(i*dx)
    L_xpfAppend(i*dx)

    mMat[i,i] = -2
    mMat[i, i-1] = 1
    mMat[i, i+1] = 1
    cMat[i] = 0

#test print matrix
print(mMat)
print(cMat)

L_xp.append(L)
L_xpfAppend(L)

solve = np.linalg.solve(mMat,cMat)

#test print
print(solve)
print(L_xp)
```

```
fig = plt.figure(figsize = (10,8))
plt.plot(L_xp, solve, 'o', color = "tab:red")
plt.plot(L_xp, solve, '-', color = "tab:blue")
plt.title('Temperature vs Position', fontsize = 12)
plt.ylabel('Temperature $^\circ$C', fontsize = 12)
plt.xlabel('Position (m)', fontsize = 12)
plt.grid(True)
plt.show()
fig.savefig('Heat_Transfer/ResultFig.jpeg',dpi = 300,bbox_inches = 'tight')

results = {'Position (m)': L_xpf, 'Temperature (C)': solve}
df1 = pandas.DataFrame(results)
print(df1)

with open('Heat_Transfer/TempTable.csv','w',newline='') as f:
    df1.to_csv(f)
    f.write("\n")
```