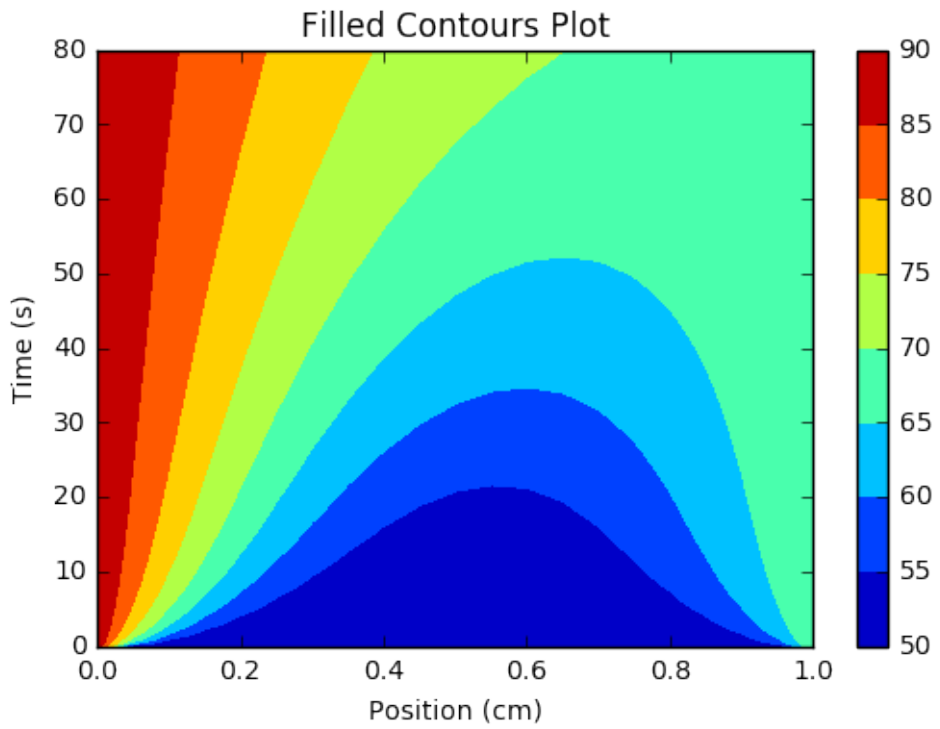
Visualization

February 23, 2017

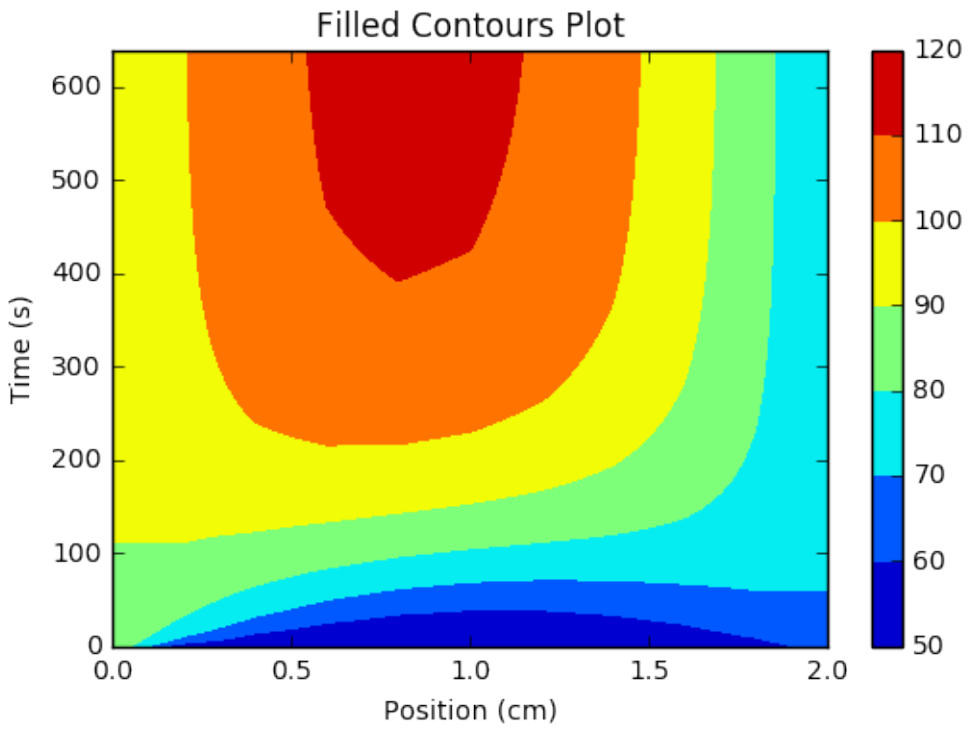
In [1]:
```
import csv;

# FIRST, WE READ-IN THE OUTPUT FROM THE SIMULATION, DELIMITER IS NEXT LINE

with open('position.txt') as inf:
    reader = csv.reader(inf, delimiter="\n")
    pos = list(zip(*reader))[0]

with open('time.txt') as inf:
    reader = csv.reader(inf, delimiter="\n")
    time = list(zip(*reader))[0]


# CHANGE THE STRING TO THE FLOAT VALUE FOR EACH READ-IN VALUES

po={};
ti={};
gridpt={};

for i in range(len(pos)):
    po[i]=float(pos[i])

for i in range(len(time)):
    ti[i]=float(time[i])

po_len=len(po)
ti_len=len(ti)

# READ-IN THE TEMPERATURE VALUE WHERE WE READ LINE BY LINE
# ONCE READ IN THEN JOIN AND FLATTENED TO THE DESIRED SHAPE(TI_LEN, PO_LEN)

import numpy as np;

file_handle2 = open('temperature.txt', 'r')
# Read in all the lines of your file into a list of lines
lines_list2 = file_handle2.readlines()
my_data2 = [[val for val in line.split()] for line in lines_list2[0:]]
```

```
flattened= [val for sublist in my_data2 for val in sublist];
flattened1=np.reshape(flattened,(ti_len,po_len))

# CHANGE THE STRING TO FLOAT

for i in range(ti_len):
    for j in range(po_len):
        gridpt[i,j]=float(flattened1[i,j])

# CREATE A CONTOUR PLOT
import numpy as np
import matplotlib.pyplot as plt

# CHANGE X, Y, Z FROM DICTIONARY TO LIST
X, Y = np.meshgrid(list(po.values()), list(ti.values()))
Z=[];
for i in range(ti_len):
    for j in range(po_len):
        Z.append(gridpt[i,j])

# RESHAPE THE Z LIST
Z=np.reshape(Z,(ti_len,po_len))

# CREATE CONTOUR AND COLOR BAR
cp = plt.contourf(X, Y, Z)
plt.colorbar(cp)
plt.title('Filled Contours Plot')
plt.xlabel('Position (cm)')
plt.ylabel('Time (s)')
plt.show()
```



In [8]:
```
import csv;

# FIRST, WE READ-IN THE OUTPUT FROM THE SIMULATION, DELIMITER IS NEXT LINE

with open('position1.txt') as inf:
    reader = csv.reader(inf, delimiter="\n")
    pos = list(zip(*reader))[0]

with open('time1.txt') as inf:
    reader = csv.reader(inf, delimiter="\n")
    time = list(zip(*reader))[0]


# CHANGE THE STRING TO THE FLOAT VALUE FOR EACH READ-IN VALUES

po={};
ti={};
gridpt={};

for i in range(len(pos)):
    po[i]=float(pos[i])

for i in range(len(time)):
```

```
    ti[i]=float(time[i])

po_len=len(po)
ti_len=len(ti)

# READ-IN THE TEMPERATURE VALUE WHERE WE READ LINE BY LINE
# ONCE READ IN THEN JOIN AND FLATTENED TO THE DESIRED SHAPE(TI_LEN, PO_LEN)

import numpy as np;

file_handle2 = open('temperature1.txt', 'r')
# Read in all the lines of your file into a list of lines
lines_list2 = file_handle2.readlines()
my_data2 = [[val for val in line.split()] for line in lines_list2[0:]]

flattened= [val for sublist in my_data2 for val in sublist];
flattened1=np.reshape(flattened,(ti_len,po_len))

# CHANGE THE STRING TO FLOAT

for i in range(ti_len):
    for j in range(po_len):
        gridpt[i,j]=float(flattened1[i,j])

# CREATE A CONTOUR PLOT
import numpy as np
import matplotlib.pyplot as plt

# CHANGE X, Y, Z FROM DICTIONARY TO LIST
X, Y = np.meshgrid(list(po.values()), list(ti.values()))
Z=[];
for i in range(ti_len):
    for j in range(po_len):
        Z.append(gridpt[i,j])

# RESHAPE THE Z LIST
Z=np.reshape(Z,(ti_len,po_len))

# CREATE CONTOUR AND COLOR BAR
cp = plt.contourf(X, Y, Z)
plt.colorbar(cp)
plt.title('Filled Contours Plot')
plt.xlabel('Position (cm)')
plt.ylabel('Time (s)')
plt.show()
```


Filled Contours Plot
Time (s)
Position (cm)
0
100
200
300
400
500
600
0.0
0.5
1.0
1.5
2.0
120
110
100
90
80
70
60
50

In [ ]: