Methods of Temperature Distribution Analysis

Benjamin Brown
Quentavious Fragher
Jefferson Kim
Jacob Mercado
Justin Pierce
Collin Williams

Goals

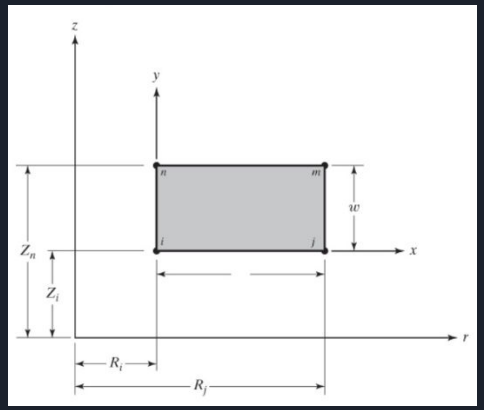
1. Create a method that gives results more accurate than FEA
2. Use results and code as proof of concept for further expansion for more complex problems and into other fields
3. Preferred a design to be quicker and more efficient that ANSYS

Project Introduction

Experimental Procedures:

Finite Element Technique Vs. Fourier's Law

- FEA calculates the temperature at each corner
  - Averages corner temperatures for temperature in direct center
  - Any other point is proportional to the average in respect to location
  - This method results in an estimation
- The analytical method used Fourier's Law as an equation to determine the exact solution



$$T=Si(Ti)+Sj(Tj)+Sk(Tk)...$$

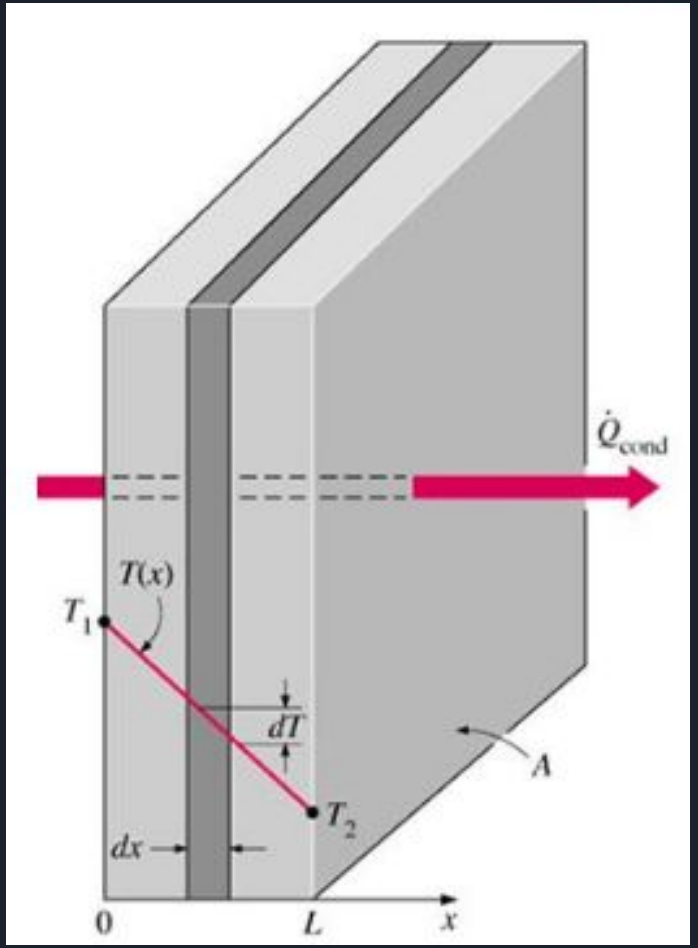
FOURIER'S LAW

$$q_x = -k A \frac{dT}{dx}$$

Scenario: 1

Conduction Through a Plane Wall

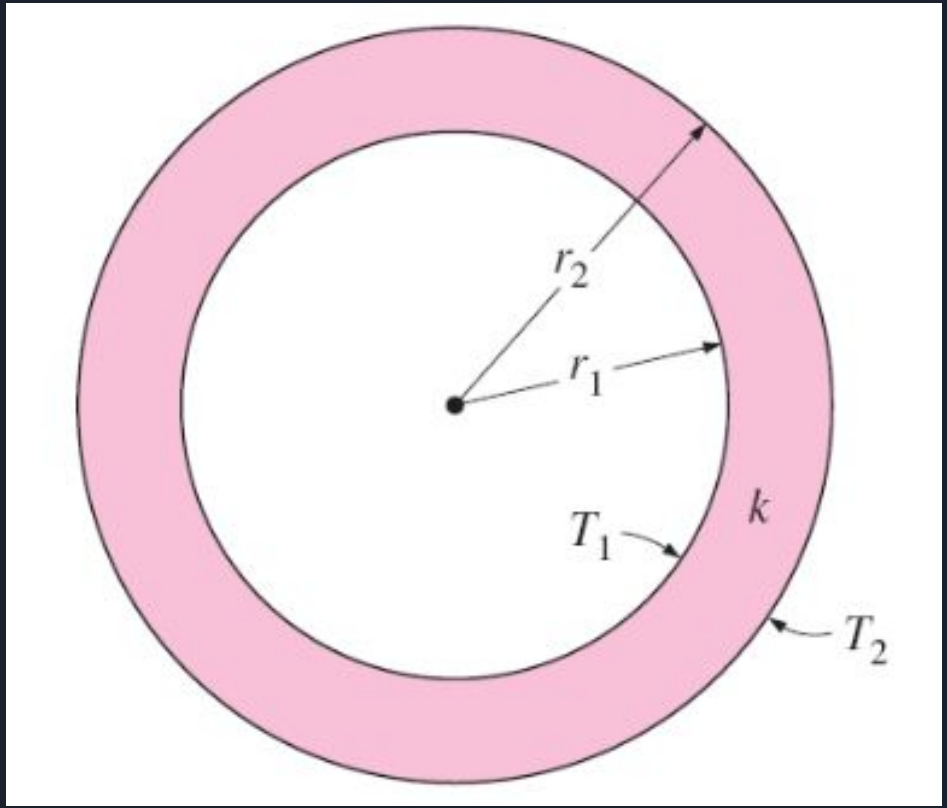
$$\dot{Q}_{\text{cond, wall}} = kA\,\frac{T_1 - T_2}{L}$$



Scenario: 2

Conduction through cylinder

$$\dot{Q}_{\text{cond, cyl}} = 2\pi Lk \frac{T_1 - T_2}{\ln(r_2/r_1)}$$



Language Selection

- Python was chosen for its reputation for scientific and numerical computing as well as its data visualization capabilities.
- The many and well documented libraries helped in this choice
- Matlab was considered but decided against due to its lack of availability compared to Python (Licensed vs. Free)

Writing the program:

Our program contains 400 words and uses commands from 4 different libraries in addition to the many natural commands. Without these libraries our code would contain an approximated 168,000 words

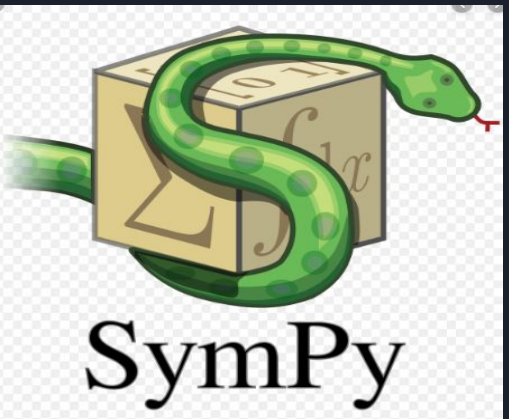
Libraries used:

- Sympy
- Numpy
- Matplotlib
- Math (built in)

Sympy

Python library with mathematical capabilities outside the realm of the standard python functionality.

Within the library exists specific packages with specific functionalities for certain purposes.



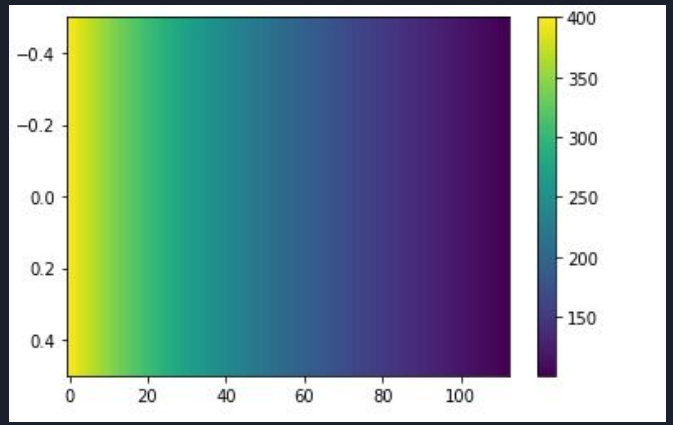

```
from sympy.matrices import Matrix
```

Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays and visualize the data.

```
import numpy as np
```

```
Tarray =  np.array(Tlist)
```

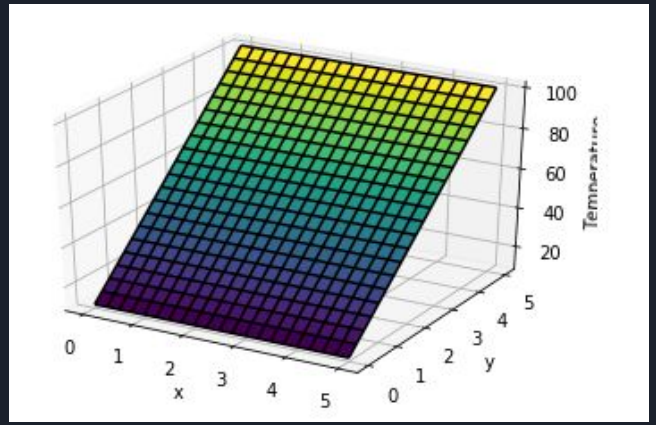


NumPy

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython.




```
import matplotlib.pyplot as plt
```

```
plt.imshow(arr, cmap='viridis', aspect='auto')
plt.colorbar()
plt.show()
```

Math

In python a number of mathematical operations can be performed with ease by importing a module named “math” which defines various functions which makes our tasks easier. This library was developed by Python as a standard library.


```
import math
```

```
pi = math.pi
```

Analytical Methodology

- User inputs
  - Verify if the geometry is rectangular or cylindrical
  - The dimensions of the object as well as the material property ‘k’
  - Desired number of intervals per unit length
  - Boundary Temperatures
  - Coordinates within the object for returning temperature
- Program creates a matrix for every single point from 0 to inputted thickness
- Returns the temperature value at the point requested by the user
- Allows user to request more temperature points

User chooses a rectangular prism or cylinder for analysis

User defines parameters for rectangular prism

User inputs coordinates to return the temperature at that point

Returns temperature, user can input another point

```
G=int(input('Rectangular(1) or Cylinder(2) '))
if G==1:
    #completed rectangular code
    # install sympy

    from sympy.matrices import Matrix

    #insert user defined parameters
    k = float(input('k value '))
    x = float(input('width of body '))
    y = float(input('length of body '))
    z = float(input('depth of body '))
    i = float(input('number of intervals per unit length '))
    Ti = float(input('Thot '))
    Tf = float(input('Tcold '))

    # A is the surface area exposed to the heat
    A = z*x
    #interval equations
    l=int((i*x)+1)
    w=int((i*y)+1)

    #Calculate the heat transfer q
    q = k*A*(Ti -Tf)/y
    #print('q = ' ,q)
    #print('Ti = ' ,((q*y)/(k*A))+Tf)

    #input requested coordinates
    u=1
    while u==1:
        xi=float(input('x coordinate: '))
        yi=float(input('y coordinate: '))
        #equation for requested coordinates
        Tr=(Ti-((q*yi/(k*A))))
        print('This is the temperature for the location you requested:',Tr)
        o=input('Request another temperature?(Y/N)')
        if o=="Y":
            u=1
        else:
            u=0
```

Creation of temperature graph

User inputs parameters for cylinder

```
    #importing matplotlib for graph
    from mpl_toolkits import mplot3d
    import numpy as np
    import matplotlib.pyplot as plt

    #defining the fuction and equation to be used
    def f(x, y):
        return (q*y)/(k*A)+Tf

    #defining the x and y area of the graph, the 3rd variable is no.elements=variable-1
    x = np.linspace(0, x, l)
    y = np.linspace(0, y, w)

    #creating the x and y data ranges
    X, Y = np.meshgrid(x, y)

    #defining the z axis
    Z = f(X, Y)
    #creating the graphical space
    fig = plt.figure()
    ax = plt.axes(projection='3d')

    #creating the graph type and appearance
    ax.plot_surface(X, Y, Z, rstride=1, cstride=1,cmap='viridis',edgecolor='none')
    ax.plot_wireframe(X, Y, Z, color='black')

    #axix settings
    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_zlabel('z');
else:
    #finished cylinder code
    #insert user defined parameters
    ri = float(input('Inner Radius '))
    ro = float(input('Outer Radius '))
    L = float(input('Length of Cylinder '))
    Ti = float(input('Inner Temperature '))
    To = float(input('Outer Temperature '))
    k = float(input('k value '))
    i = float(input('number of intervals per unit length '))
```

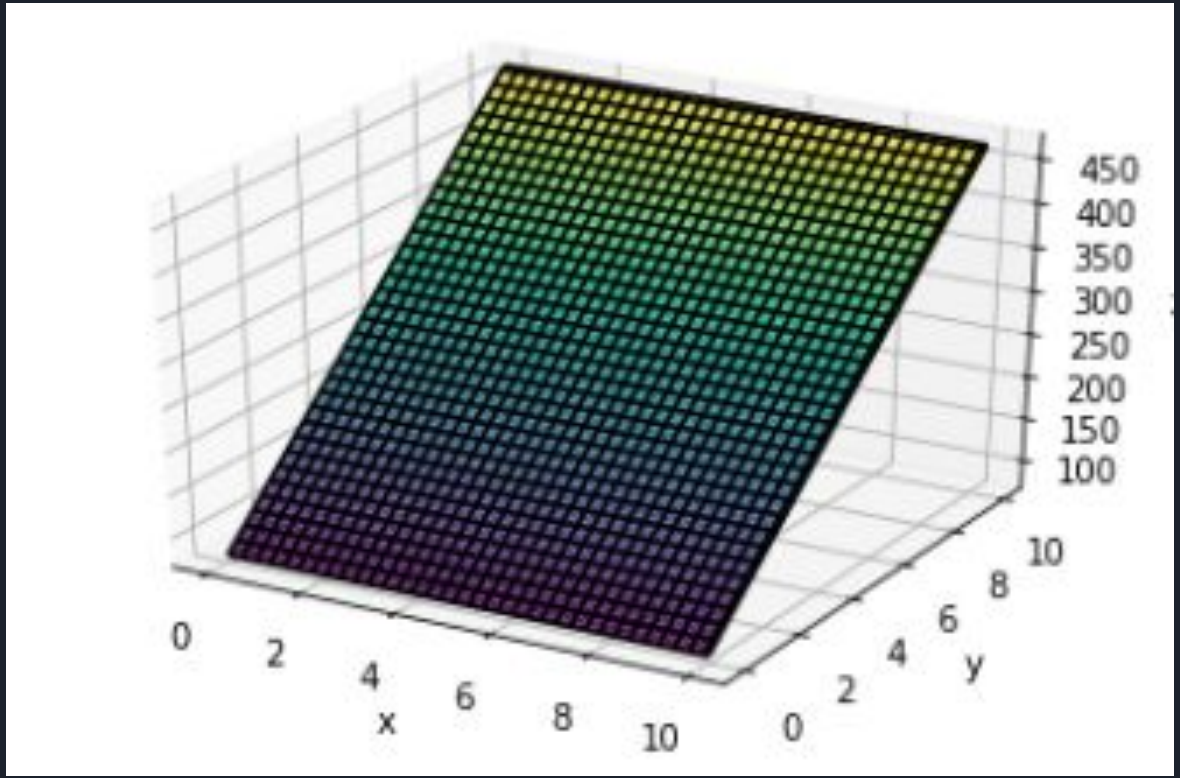
User inputs distance from inner radius for temperature

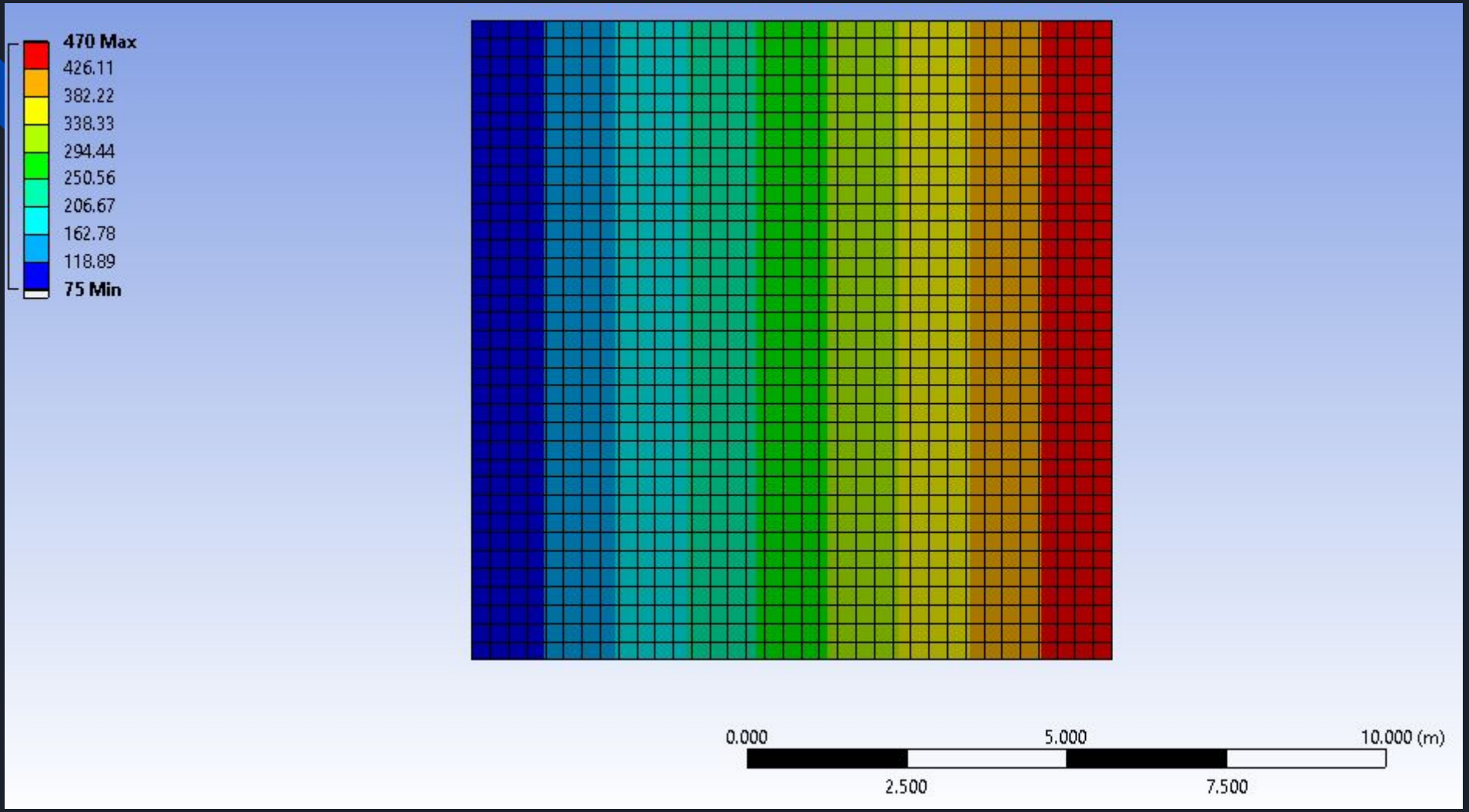
Temperature graph

```
#Calculate the heat transfer q
import numpy as np
import math

pi = math.pi
q = 2*pi*L*k*((Ti-To)/np.log(ro/ri))
print('q = ' ,q)
#print('To = ' ,Ti-(q*np.log(ro/ri))/(2*pi*L*k))

#input requested coordinates
u=1
while u==1:
    rreq=float(input('Distance from ri: '))
    #equation for requested coordinates
    Tr=Ti-(q*np.log(rreq/ri))/(2*pi*L*k)
    print('This is the temperature for the location you requested:',Tr)
    o=input('Request another temperature?(Y/N)')
    if o=="Y":
        u=1
    else:
        u=0

import matplotlib.pyplot as plt
import numpy as np

def heatmap2d(arr: np.ndarray):
    plt.imshow(arr, cmap='viridis', aspect='auto')
    plt.colorbar()
    plt.show()
#creates an array of data using np.arange of size () then reshapes into a matrix using .reshape of size (x (,:by) y)
r=ri
size=int(i*(ro-ri)+1)

Tlist = []

while r <= ro:
    T = Ti-(q*np.log(r/ri))/(2*pi*L*k)
    Tlist.append(T)
    r=r+(1/i)

Tarray =  np.array(Tlist)
Tmatrix = Tarray.reshape(1,size)
#print(Tmatrix)
heatmap2d(Tmatrix)
```

Structural Steel Rectangular Prism

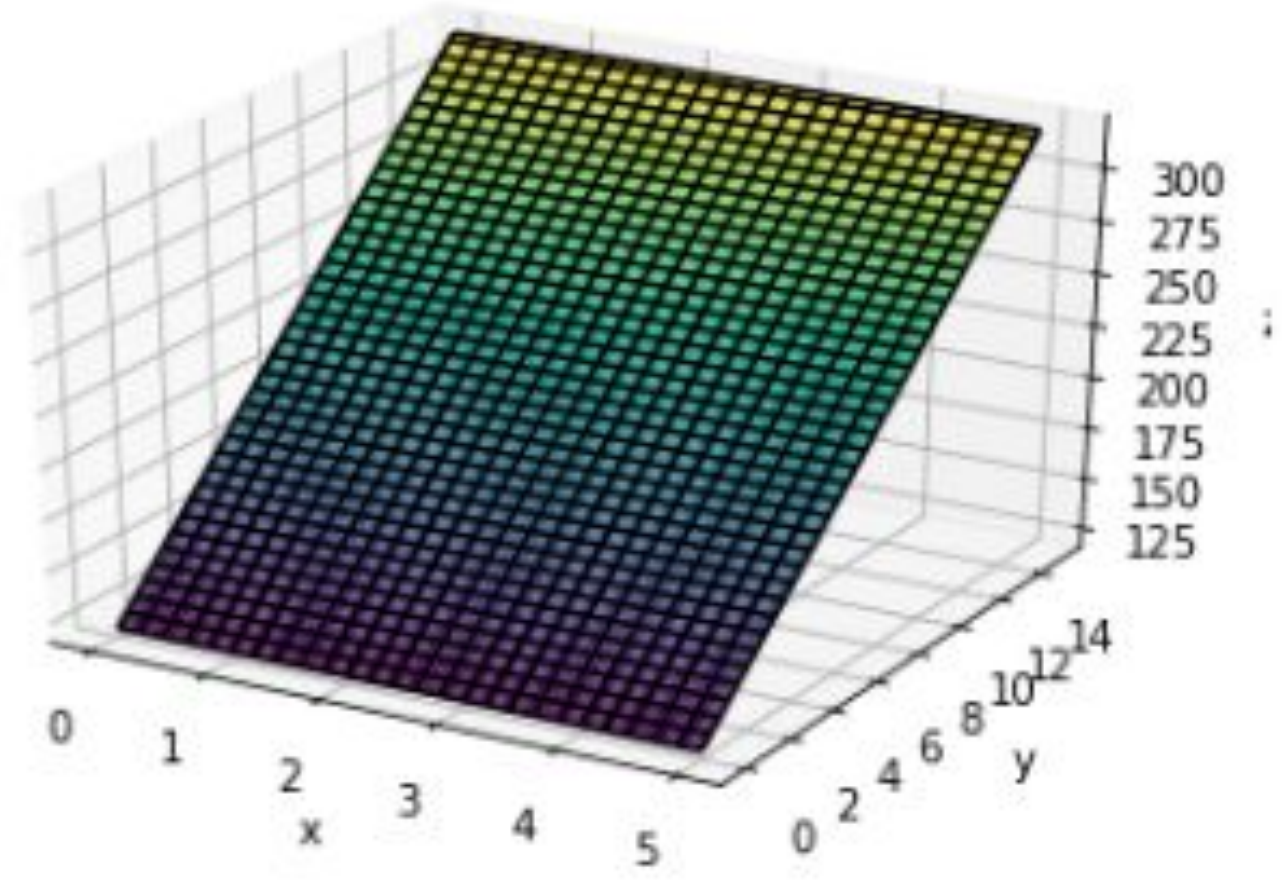
```
Rectangular(1) or Cylinder(2) 1
k value 60.5
width of body 10
length of body 10
depth of body 5
number of intervals per unit length 10
Thot 470
Tcold 75
x coordinate: 1.45
y coordinate: 2.86
This is the temperature for the location you requested: 357.03
Request another temperature?(Y/N)Y
x coordinate: 9.75
y coordinate: 6.874
This is the temperature for the location you requested: 198.47700000000003
Request another temperature?(Y/N)Y
x coordinate: 3.33
y coordinate: 5.42
This is the temperature for the location you requested: 255.91
Request another temperature?(Y/N)N
```

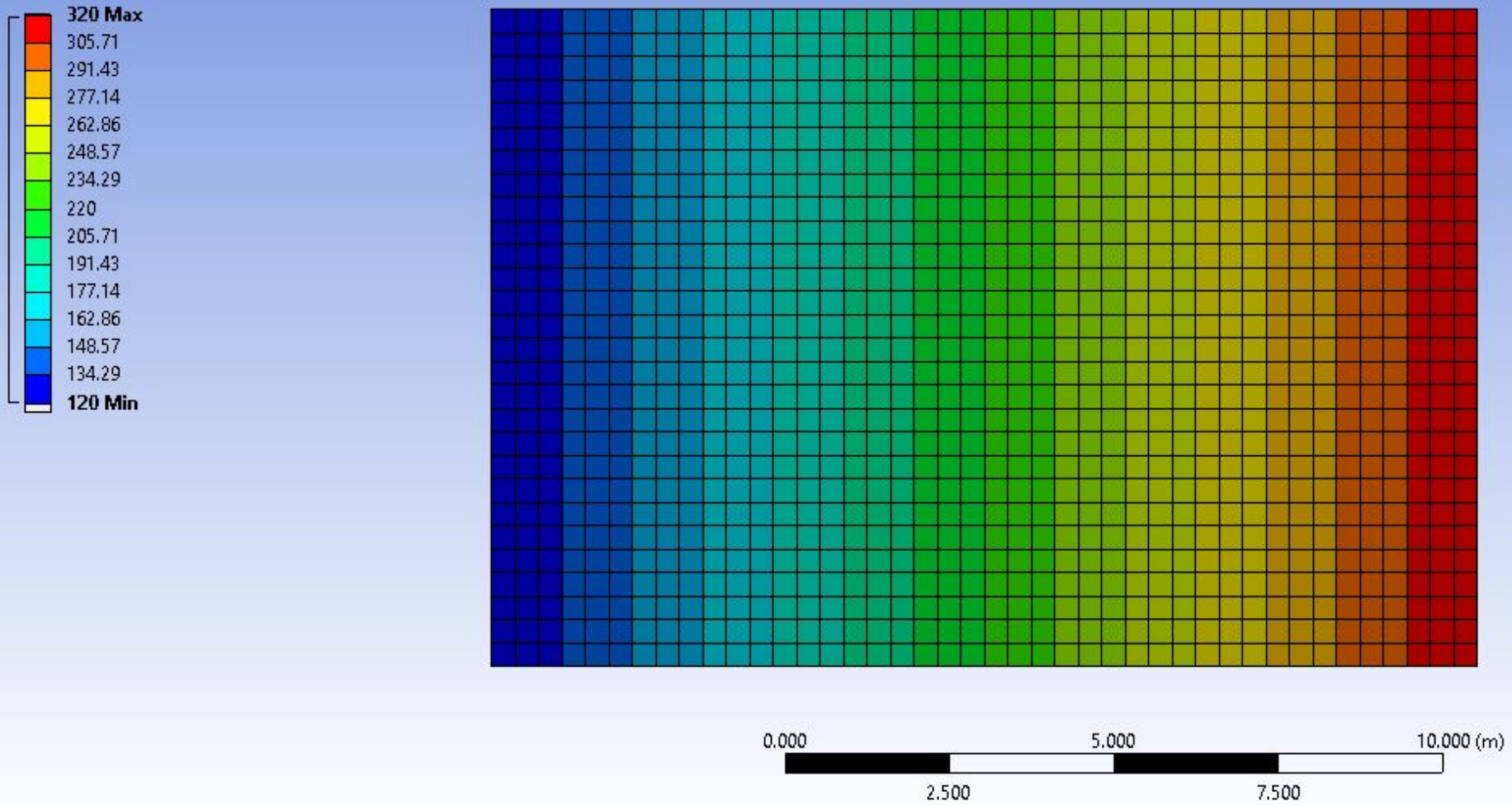

450
400
350
300
250
200
150
100
0
2
4
6
8
10
x
0
2
4
6
8
10
y


470 Max
426.11
382.22
338.33
294.44
250.56
206.67
162.78
118.89
75 Min
0.000
5.000
10.000 (m)
2.500
7.500

Copper-Bronze Alloy Rectangular Prism

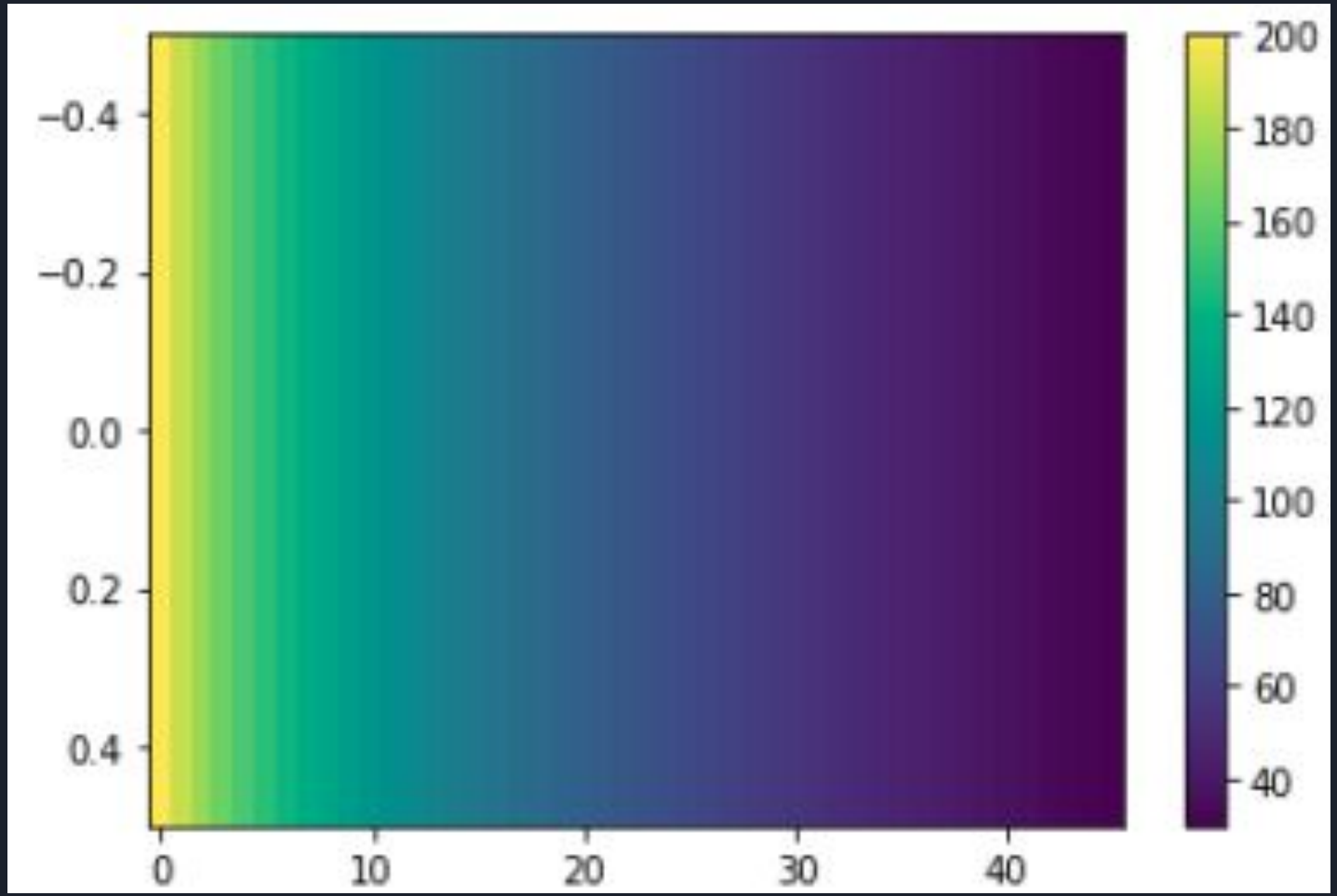
```
Rectangular(1) or Cylinder(2) 1
k value 26
width of body 5
length of body 15
depth of body 10
number of intervals per unit length 10
Thot 320
Tcold 120
x coordinate: 4
y coordinate: 11
This is the temperature for the location you requested: 173.33333333333334
Request another temperature?(Y/N)Y
x coordinate: 3
y coordinate: 14.79
This is the temperature for the location you requested: 122.80000000000001
Request another temperature?(Y/N)Y
x coordinate: 1
y coordinate: 3.789
This is the temperature for the location you requested: 269.48
Request another temperature?(Y/N)N
```

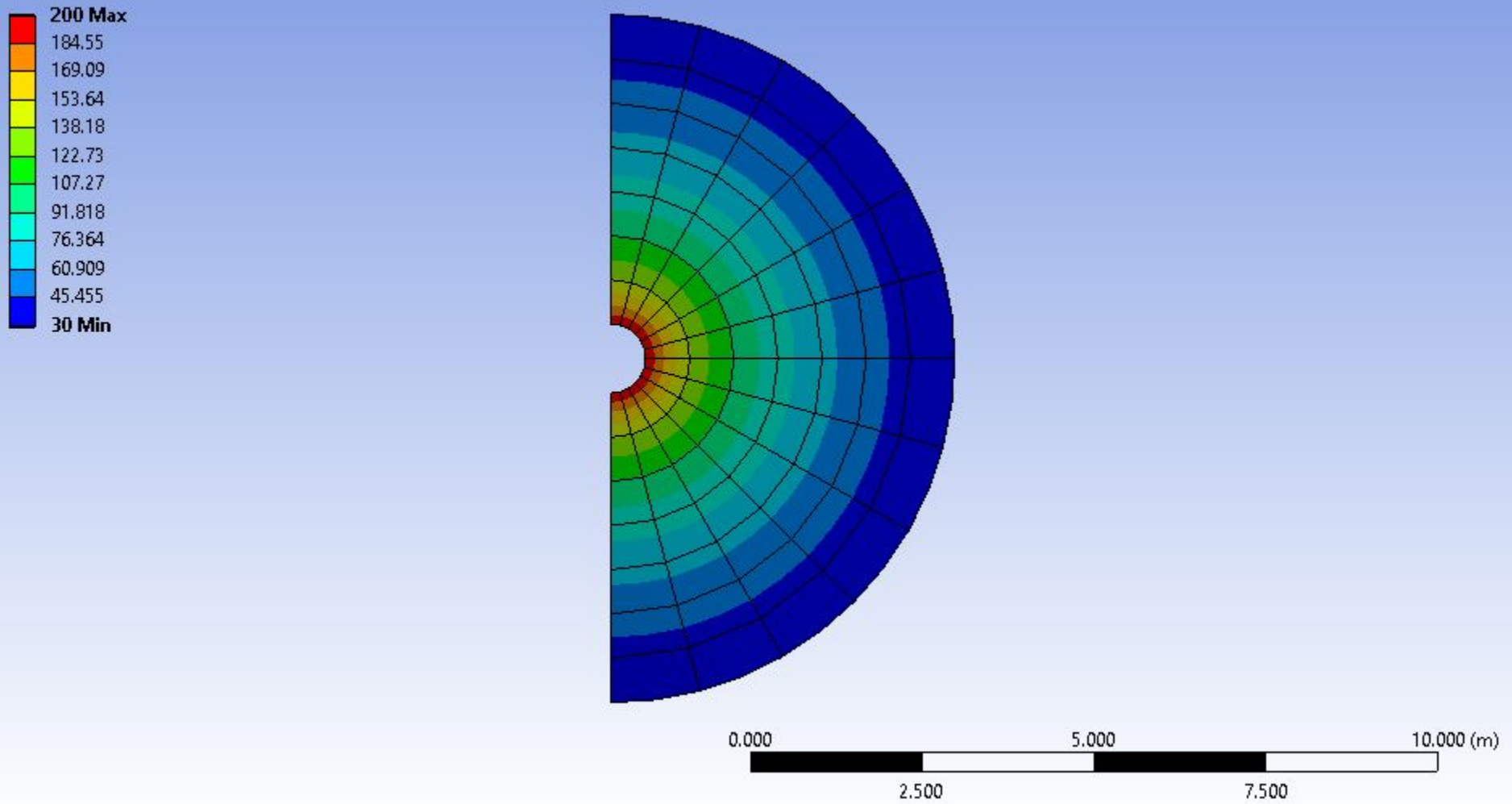

300
275
250
225
200
175
150
125
0
1
2
3
4
5
x
0
2
4
6
8
10
12
14
y


320 Max
305.71
291.43
277.14
262.86
248.57
234.29
220
205.71
191.43
177.14
162.86
148.57
134.29
120 Min
0.000
5.000
10.000 (m)
2.500
7.500

Structural Steel Cylinder

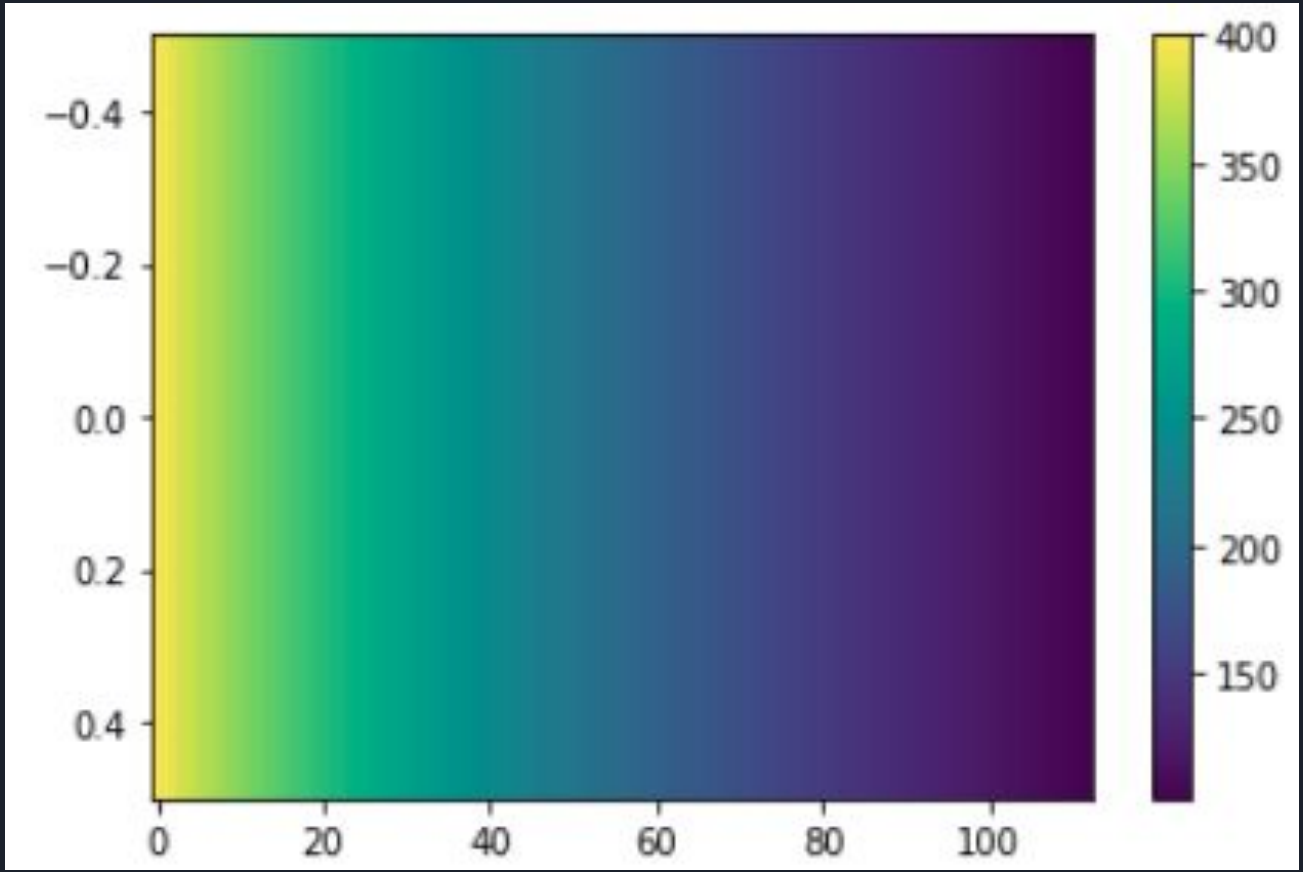
```
Rectangular(1) or Cylinder(2) 2
Inner Radius 0.5
Outer Radius 5
Length of Cylinder 10
Inner Temperature 200
Outer Temperature 30
k value 60.5
number of intervals per unit length 10
q =  280652.2159852667
Distance from ri: 1.25
This is the temperature for the location you requested: 132.35019852575363
Request another temperature?(Y/N)Y
Distance from ri: 4
This is the temperature for the location you requested: 46.47470221136962
Request another temperature?(Y/N)Y
Distance from ri: 3.05
This is the temperature for the location you requested: 66.49392804816964
Request another temperature?(Y/N)N
```

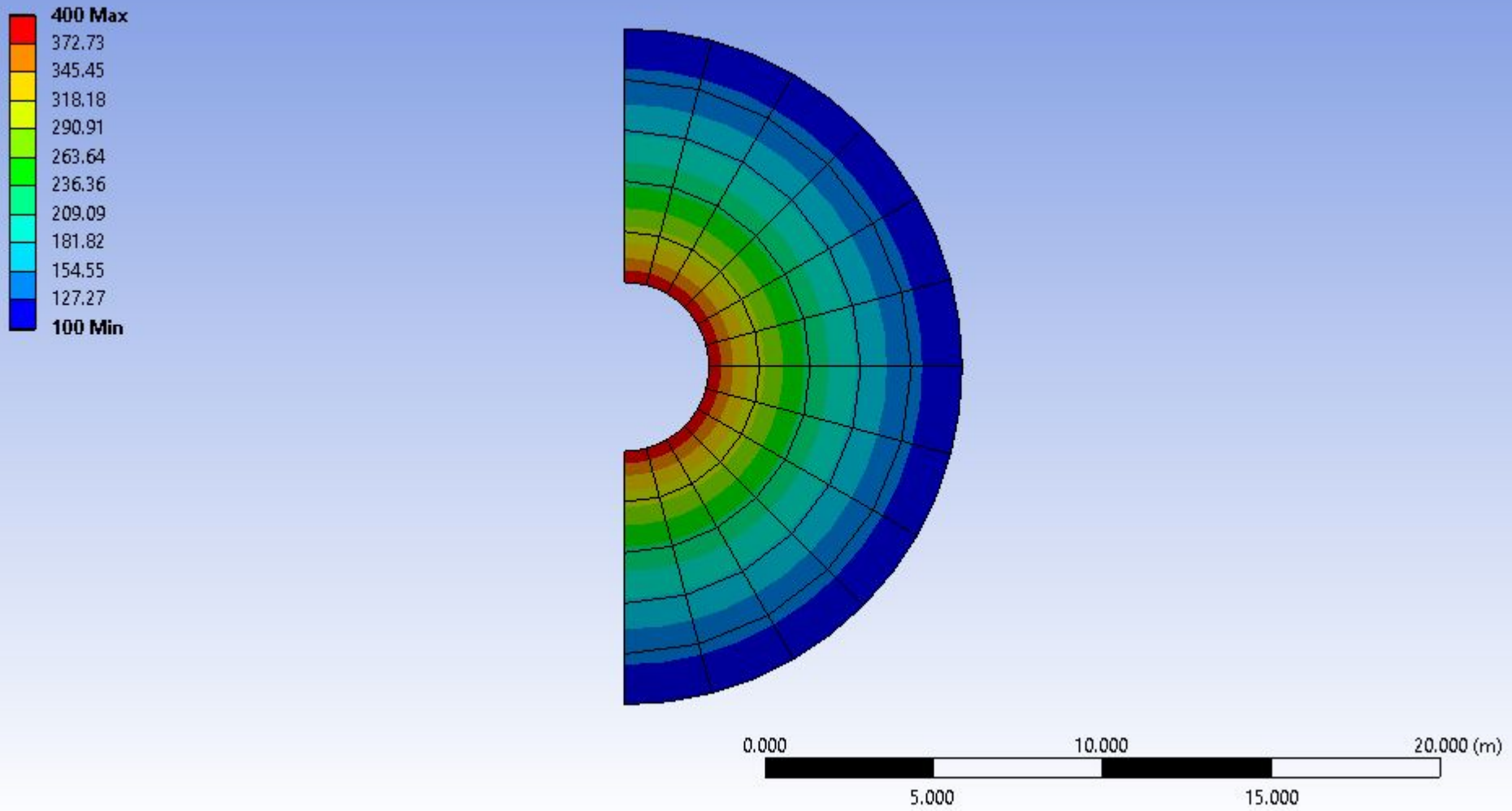

−0.4
−0.2
0.0
0.2
0.4
0
10
20
30
40
200
180
160
140
120
100
80
60
40


200 Max
184.55
169.09
153.64
138.18
122.73
107.27
91.818
76.364
60.909
45.455
30 Min
0.000
5.000
10.000 (m)
2.500
7.500

Copper-Bronze Alloy Cylinder

```
Rectangular(1) or Cylinder(2) 2
Inner Radius 2.5
Outer Radius 10
Length of Cylinder 25
Inner Temperature 400
Outer Temperature 100
k value 26
number of intervals per unit length 15
q =  883810.2276563029
Distance from ri: 7
This is the temperature for the location you requested: 177.18597592446375
Request another temperature?(Y/N)Y
Distance from ri: 1.83
This is the temperature for the location you requested: 467.51266695670665
Request another temperature?(Y/N)Y
Distance from ri: 4.32
This is the temperature for the location you requested: 281.6345173747927
Request another temperature?(Y/N)N
```


−0.4
−0.2
0.0
0.2
0.4
0
20
40
60
80
100
400
350
300
250
200
150


400 Max
372.73
345.45
318.18
290.91
263.64
236.36
209.09
181.82
154.55
127.27
100 Min
0.000
10.000
20.000 (m)
5.000
15.000

Conclusion

- Data from the coding and FEA methods return the same values
- Each method has its advantages
- Once code has been established, it is very fast and easy to change parameters and return specific points within the body
- It is also much easier for non tech savvy people to collect data
- FEA method is more visually appealing
- It is also easier to create and analyze more complex shapes and bodies
- If only one figure/model needs to be created, FEA would take much less time to create it than coding would

References

[1] Park, Ki-Bong; Jee, Nam-Yong; Yoon, In-Seok; Lee, Han-Seung. 2008. “Prediction of Temperature Distribution in High-Strength Concrete Using Hydration Model.” *ACI Materials Journal.* Vol. 105, Iss. 2. (Accessed 11/13/2019) https://search.proquest.com/docview/197937710?pq-origsite=gscholar

[2] Toparli, M.; Sahin, S.; Ozkaya, E.; Sasaki, S. 2002. “Residual thermal stress analysis in cylindrical steel bars using finite element method and artificial neural networks.” *Computers & Structures*. Vol. 80, Iss. 23. (Accessed 11/13/2019) https://www.sciencedirect.com/science/article/abs/pii/S0045794902002158

[3] Serajzadeh, Siamak. 2007. “Prediction of temperature distribution and required energy in hot forging process by coupling neural networks and finite element analysis.” *Materials Letters.* Vol. 61, Iss. 14-15. (Accessed 11/13/2019) https://www.sciencedirect.com/science/article/abs/pii/S0167577X06013711

[4] Jambunathan, K.; Hartle, S.L.; Ashforth-Frost, S.; Fontama, V.N. 1996. “Evaluating convective heat transfer coefficients using neural networks.” *International Journal of Heat and Mass Transfer.* Vol. 39, Iss. 11. (Accessed 11/13/2019) https://www.sciencedirect.com/science/article/pii/0017931095003320

[5] Lienhard, John H. IV; Lienhard, John H. V. 2019. “A Heat Transfer Textbook.” 5th Edition. *Phlogiston Press*. (Accessed 12/3/19) https://ahtt.mit.edu/wp-content/uploads/2019/08/AHTTv500.pdf