**AERO 430 – Exam 2**

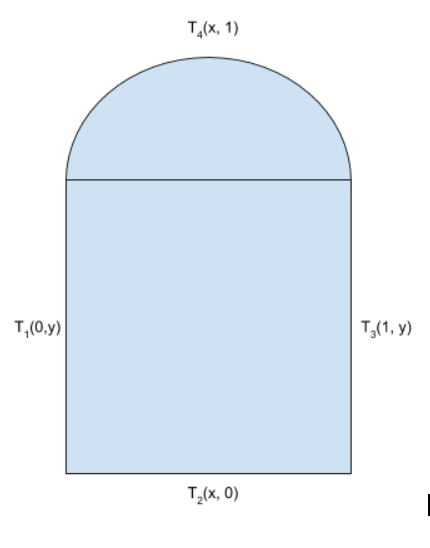


Antonio Diaz ‘22

UIN 327003625

**Due 04/14/2020**

**1)** **Analytical Solution to 2D Heat Equation for infinitely long plate.**

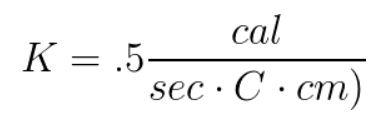


Case 1 Boundary conditions:

Case 2 Boundary conditions:

, where K is a conductivity constant.

Constants are defined as follows:



Length = 1 cm

Radius = .1 cm

The governing equation describing heat flow across any surface is .

Following the conservation of heat flow, Q is defined as:

This can be reduced by simplifying the K constants,

**Separation of variables to solve Dirichlet** **2nd order homogenous differential equation:**

The heat function can be separated into . Plugging back into PDE:

or

Boundary Conditions for Case 1:

Boundary Conditions for Case 2:

General solution to :

from boundary conditions

gives us A = 0

gives us , n is any integer

This gives us

General solution to :

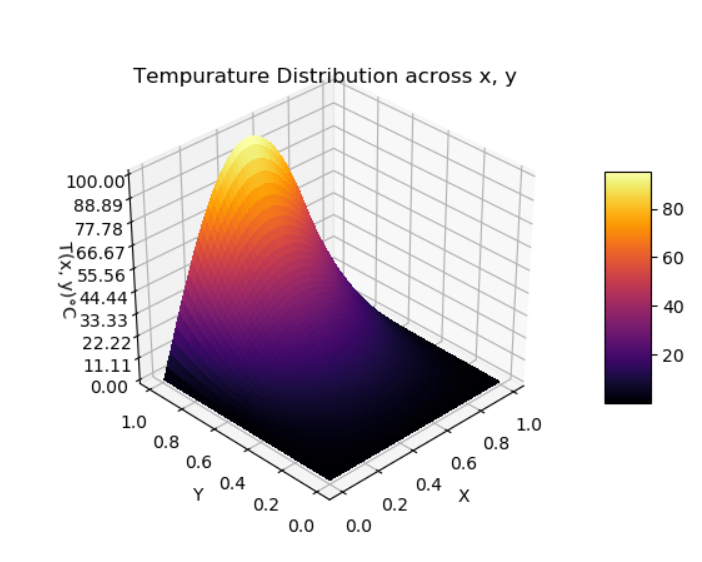
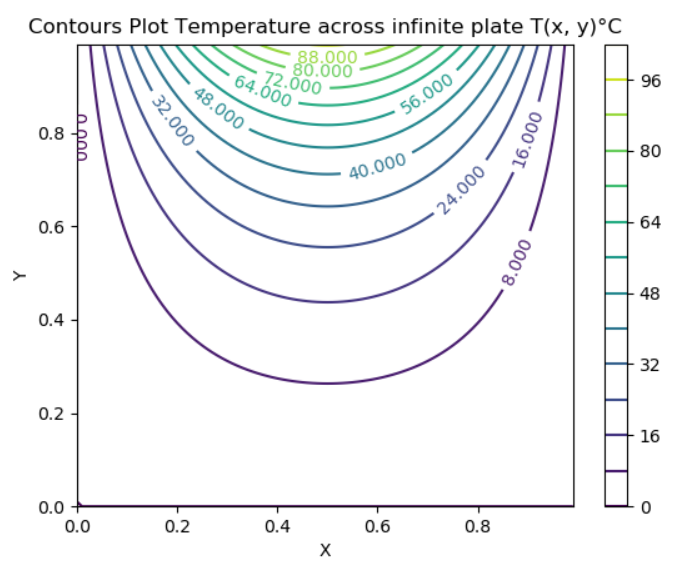
from boundary conditions

= 0

This system of equations combined with boundary conditions reduces to:

Final General Solution to Heat Equation:

This heat temperature function is graphed below:

**2) FDM for point :**

Ordinary Taylor Series Expansion:

+ Error

+ Error

Taylor series expansion for point:

+

Subtracting Taylor series to find the second order approximation of U’:

=

Dividing both sides by gives us

Similarly, for

Adding Taylor series to find the second order approximation of U’’:

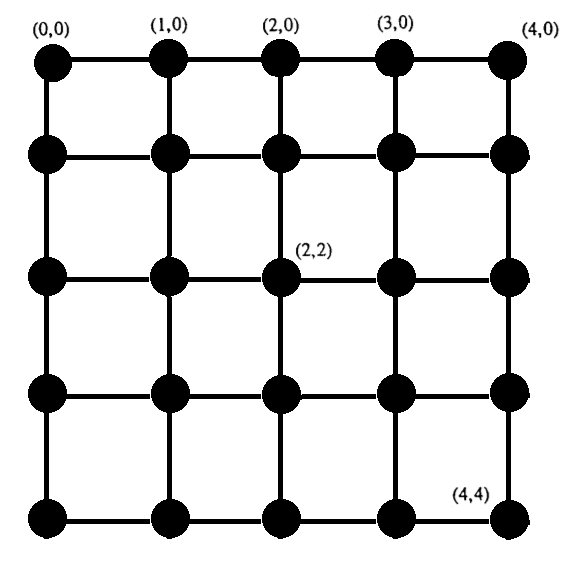
This is then reduced after removing truncation error, becomes:

Using the Taylor series similar for the fourth order approximation of U’’:

Simplifying these expressions using the heat equations:

**3. Application of FDM across 2D mesh:**

A uniform 2D mesh of is used to create the following:



Applying the conditions for approximating the heat equation:

Using previously derived second order approximation

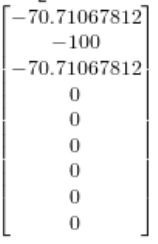
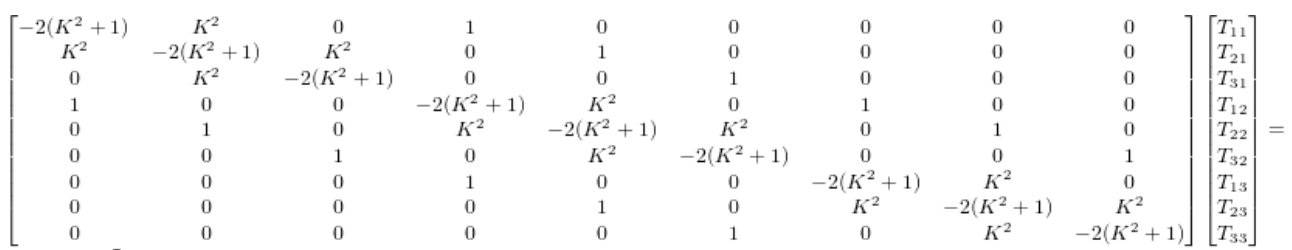
= 0

Using

Applying system of equations to

Applying system of equations to for shown mesh:

This is then reduced to a global matrix that solved for each temperature node:



\*note that order of elements is from top to right and down in mesh. Some other published results may start from bottom right of nodes in mesh.

4th order FDM is similarly developed as follows:

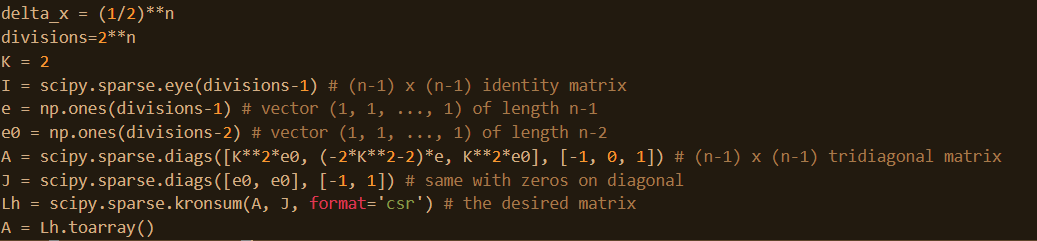
Applying the conditions for approximating the heat equation:

Using previously derived 4th order approximation:

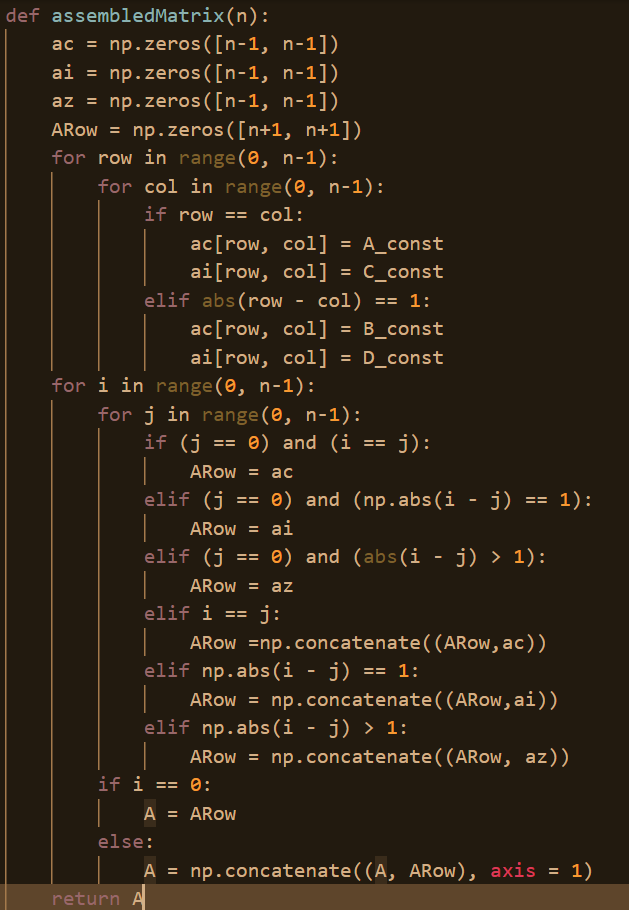
= 0

Summary of python code for 2nd order approximation:

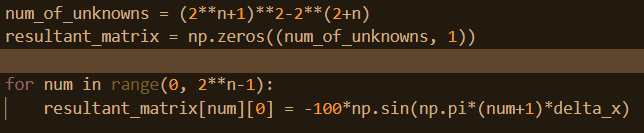
Code to create global matrix A as function of n, where n is degree from :



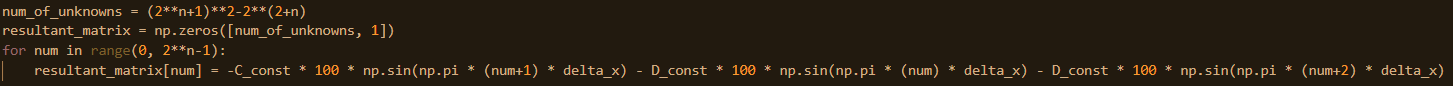
Assembled global matrix for 4th order approximation:



Code to create right hand side with 2nd order boundary conditions:



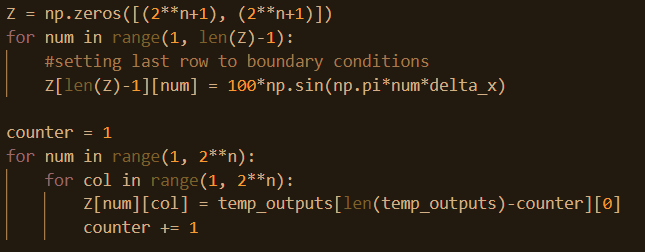
Code to create right had side with 4th order boundary conditions:



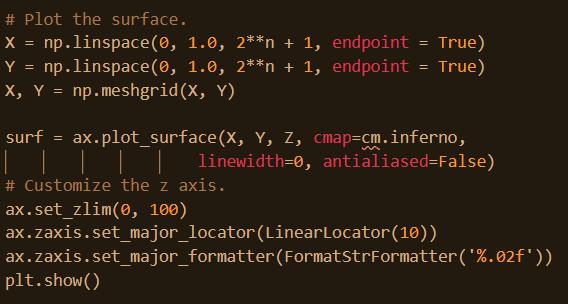
Solving for temperature matrix:



Code for mapping temperature matrix to mesh for matplotlib to plot:



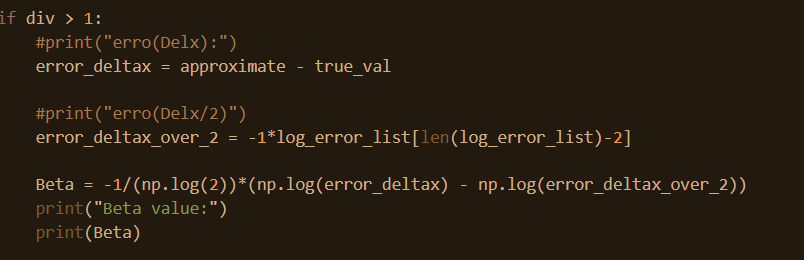
Plotting x, y, z, mesh from finite difference:

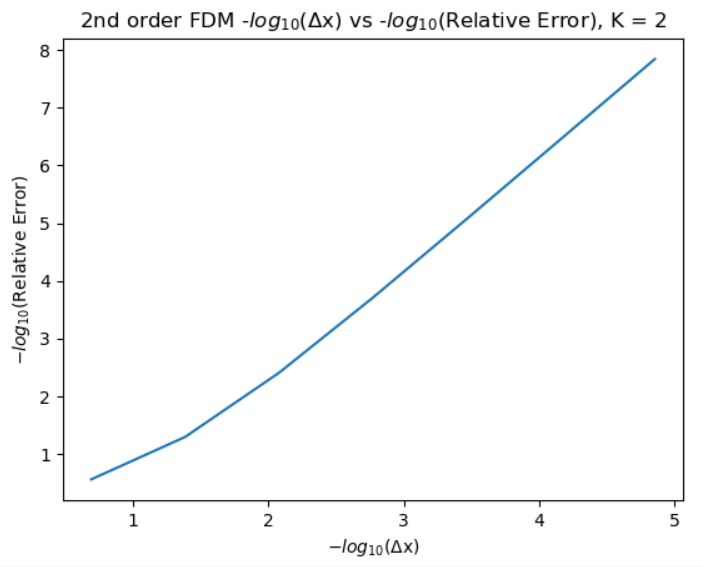


**4. Convergence Rates and plots for 2nd order FDM:**

Beta is measured using , where .

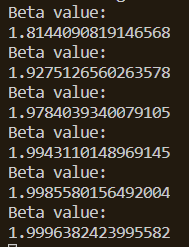
This code uses the index of the relative error array created to calculate Beta as follows:





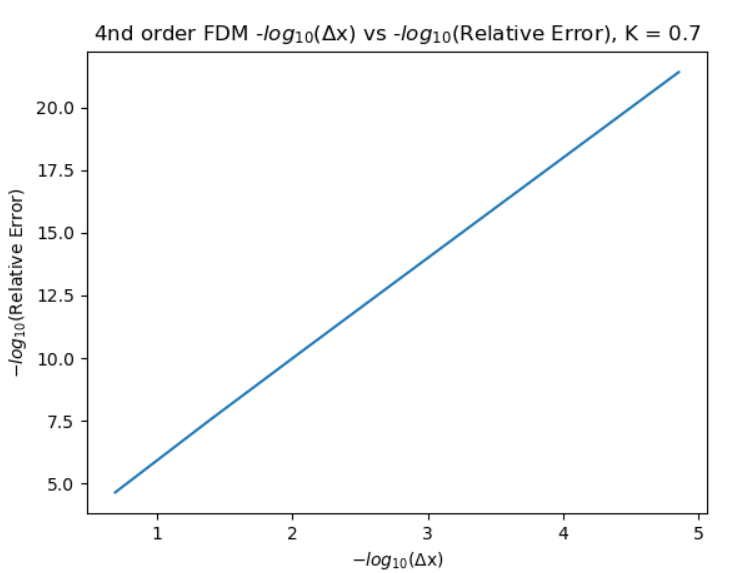
Slope is 1.8 with = .993

This output is the following in Python:

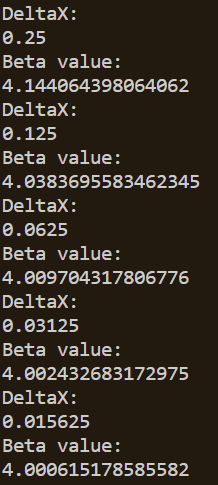


**5. Convergence Rates and plots for 4th order FDM:**

Slope approaches 4.44, with



Beta values for 4th order FDM:



**Conclusion**:

Using finite element models, the Temperature as a function of position on the heat rod can be estimated. The current models used in this assignment make use of hierarchical functions, condensation, assembly, and penalty functions to find the temperature along the rod. The results were found using p = 1, 2, 3, 4, and 5 for two different boundary conditions. The values of convergence reach their expected values for all orders for case 1 of about 2\*p. As the deltaX increased, for larger p values, the convergence began to be dominated by truncation error as expected. This can likely be reduced by using other numpy extended precision functions in python. This code could also be improved by making a function that will automatically calculate Schur’s complement based on the p value instead of hardcoding each condensation for each value of p.

Compared to the error graphs of FDM, for the same deltaX, FEM reaches better precision, noted from the log(deltaX) and log(error graphs).

This assignment was useful in seeing how quickly the accuracy can improve with higher orders of FEM convergence. ­­