**AERO 430 – Exam 2**

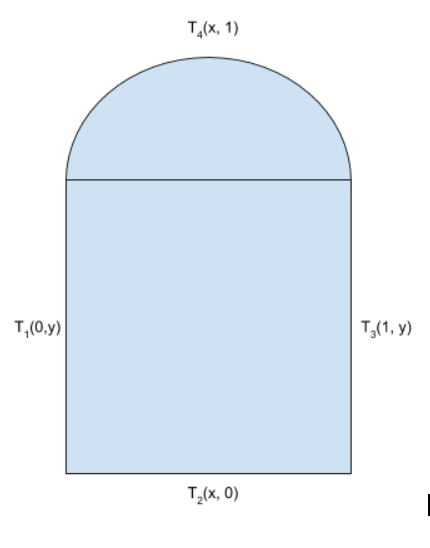


Antonio Diaz ‘22

UIN 327003625

**Due 04/20/2020**

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

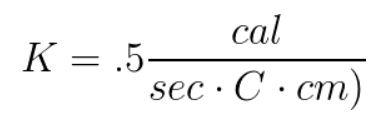


Case 1 Boundary conditions:

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 :

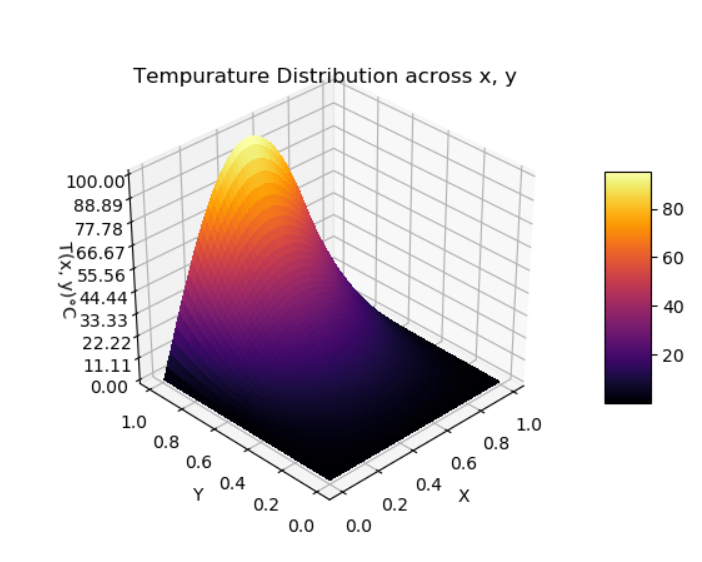
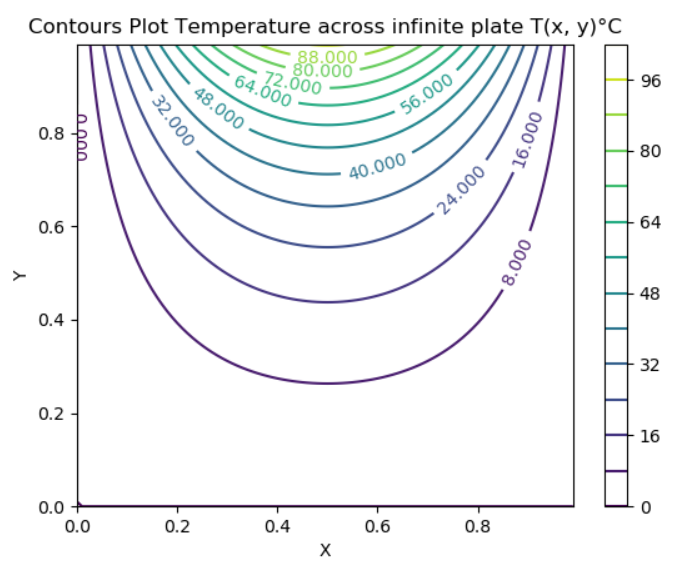
from boundary conditions

= 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’’:

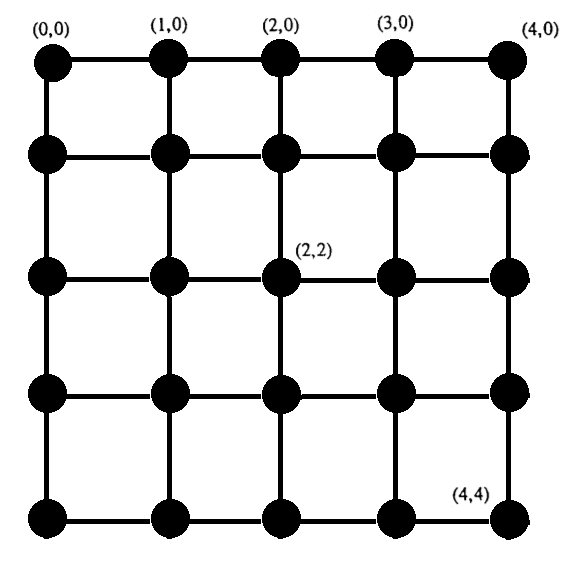
This is then reduced after removing truncation error, becomes:

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

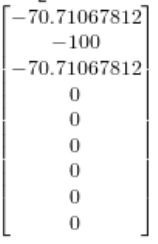
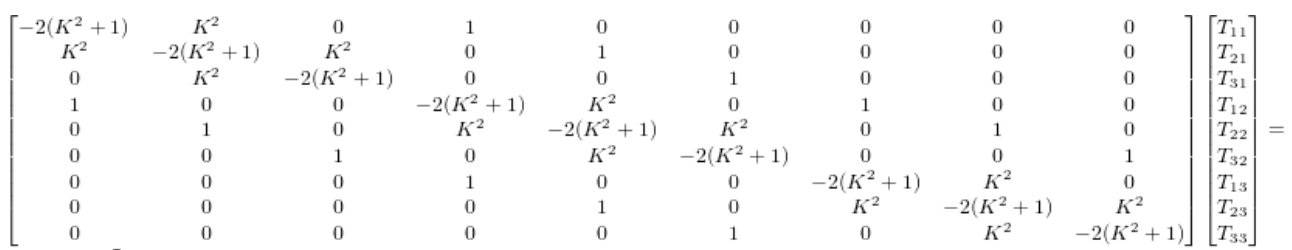
= 0

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:

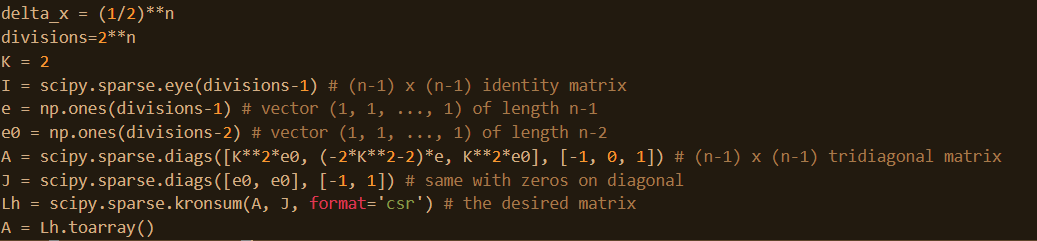
Applying the conditions for approximating the heat equation:

Using previously derived 4th order approximation:

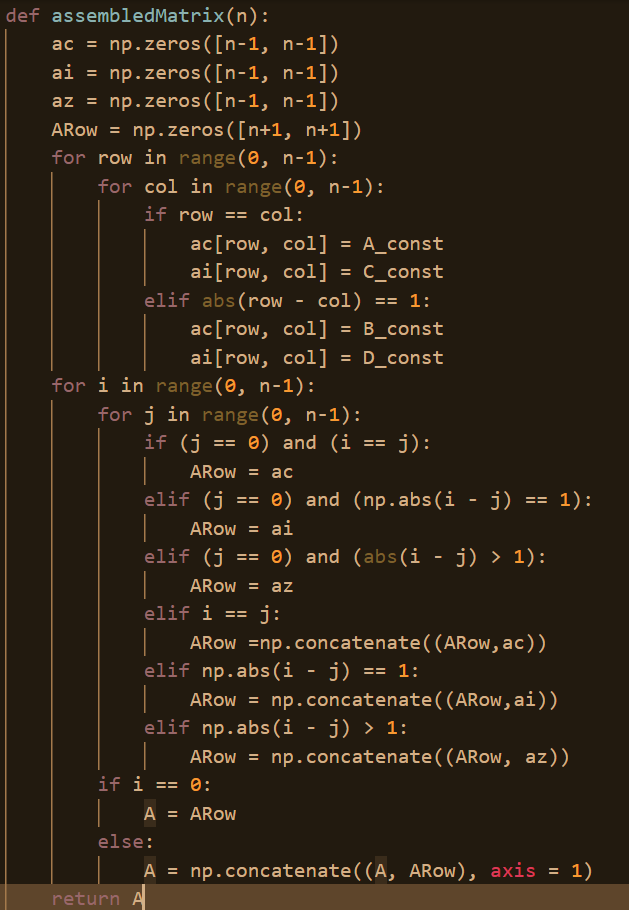
= 0

Summary of python code for 2nd order approximation:

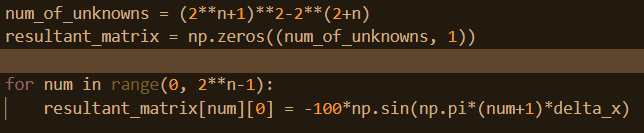
Code to create global matrix A as function of n, where n is degree from :



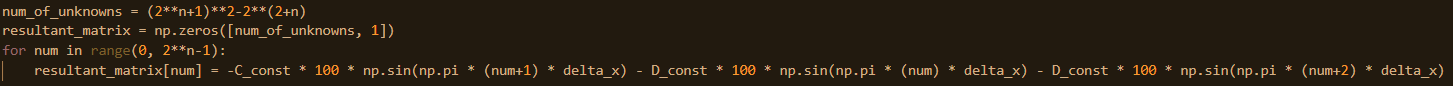
Assembled global matrix for 4th order approximation:



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



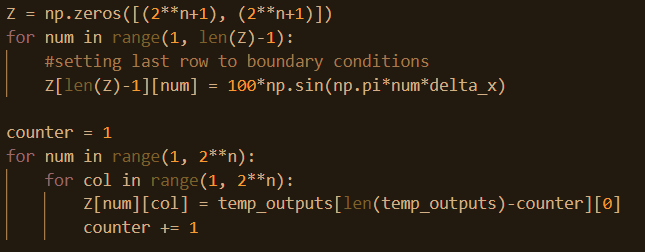
Code to create right had side with 4th order boundary conditions:



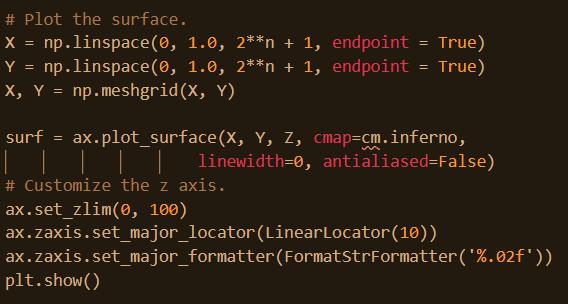
Solving for temperature matrix:



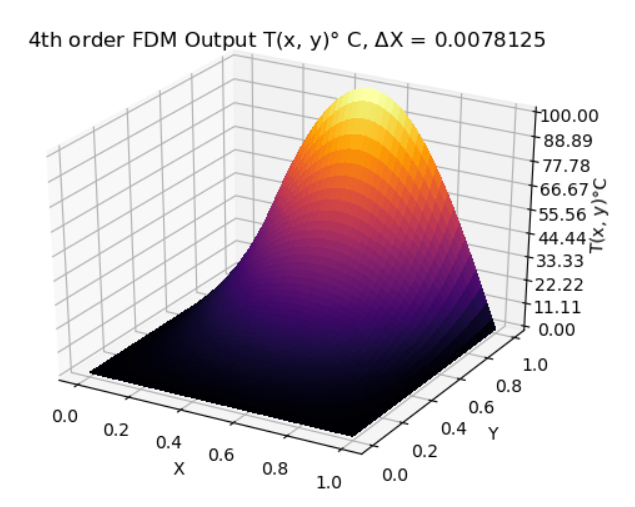
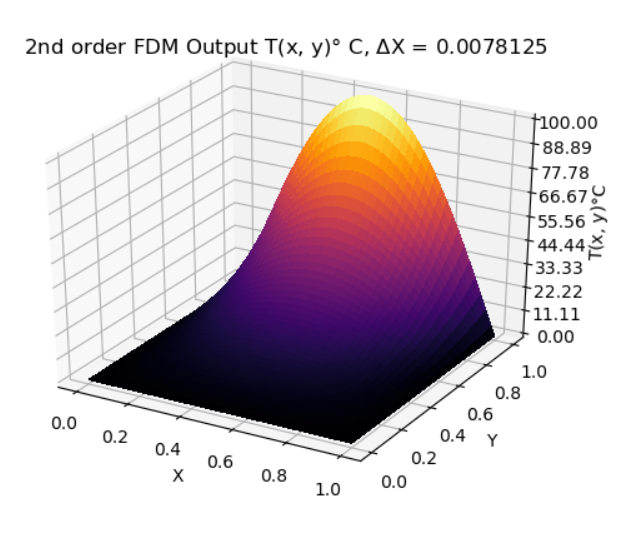
Code for mapping temperature matrix to mesh for matplotlib to plot:



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



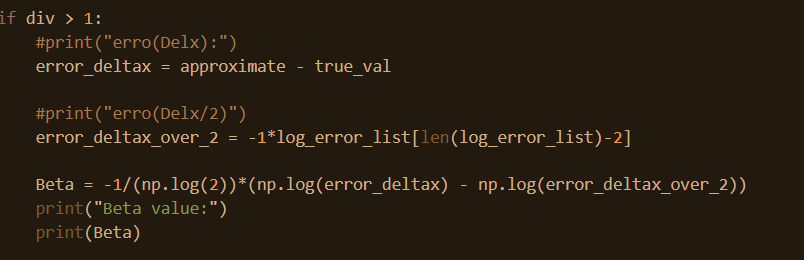
2nd and 4th order FDM output for 2\*\*7 divisions, K = .7:

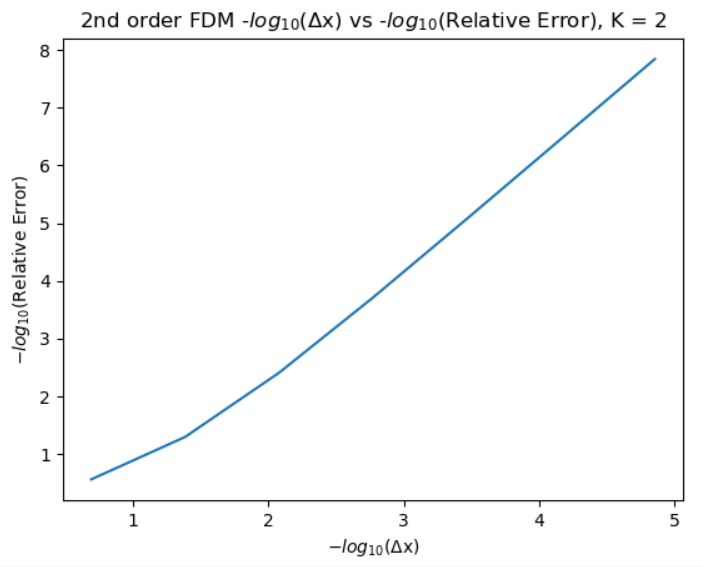


**4. Convergence Rates and plots for 2nd order FDM:**

Beta is measured using , where .

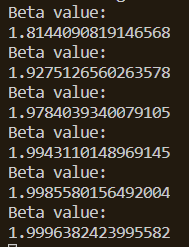
This code uses the index of the relative error array created to calculate Beta as follows:





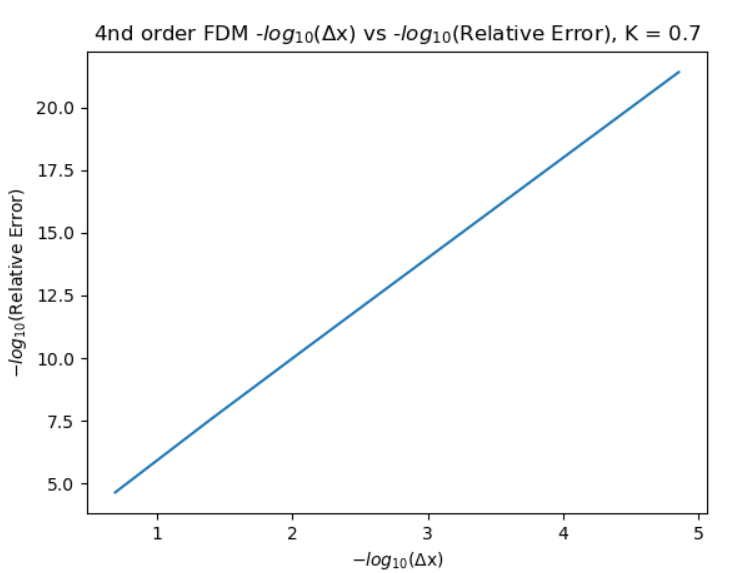
Slope is 1.8 with = .993

This output is the following in Python:

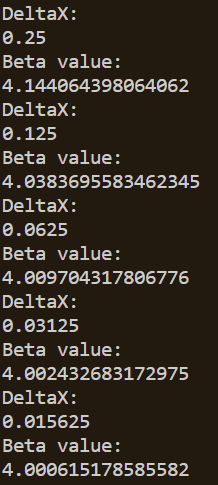


**5. Convergence Rates and plots for 4th order FDM:**

Slope approaches 4.44, with



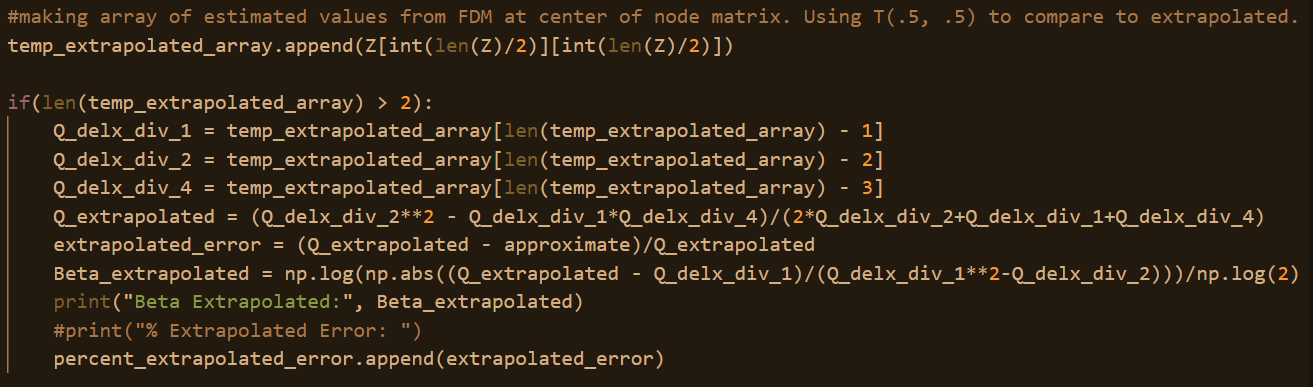
Beta values for 4th order FDM:



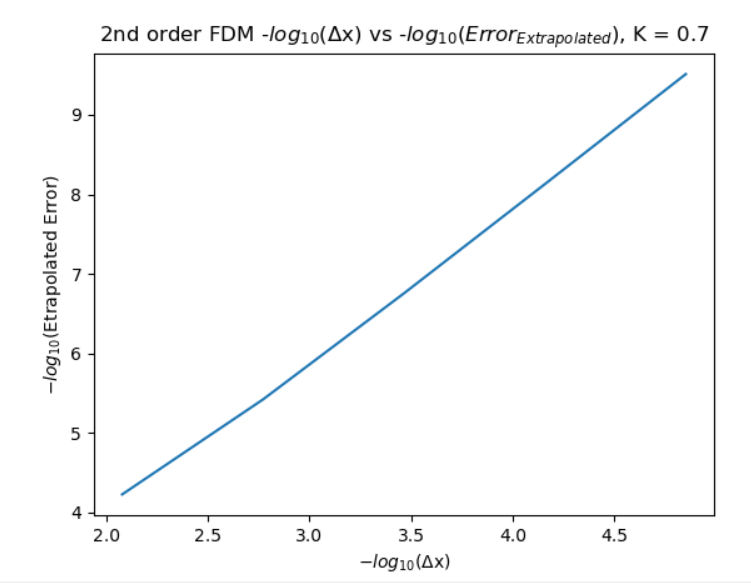
**6. Applying Richardson Extrapolation to FDM Output**

**/()**

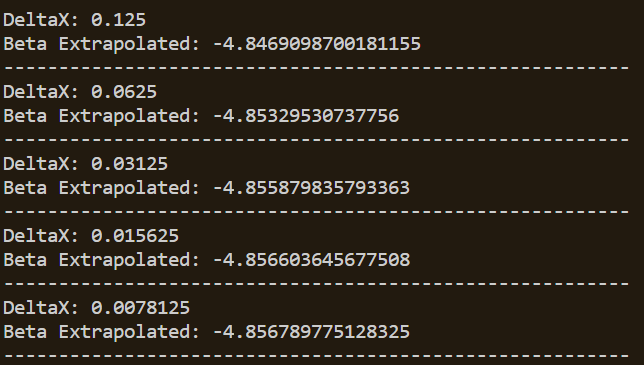
Each iteration of increasing step size, adds the FDM output at T(.5, .5) to the an array. Once this array has a length greater than 2, the Richardson extrapolation can be applied. This is useful in comparing the convergence of the FDM output to an extrapolated value. The convergence () is calculated by:



**2nd order output log() vs -log(Error from Extrapolation), K = .7:**

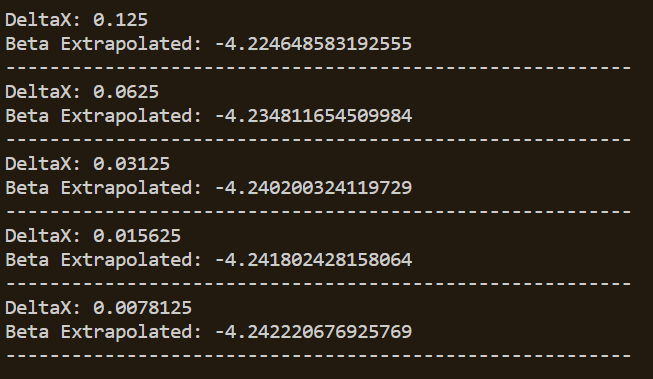


**2nd order Convergence values using Extrapolation, K = .7:**

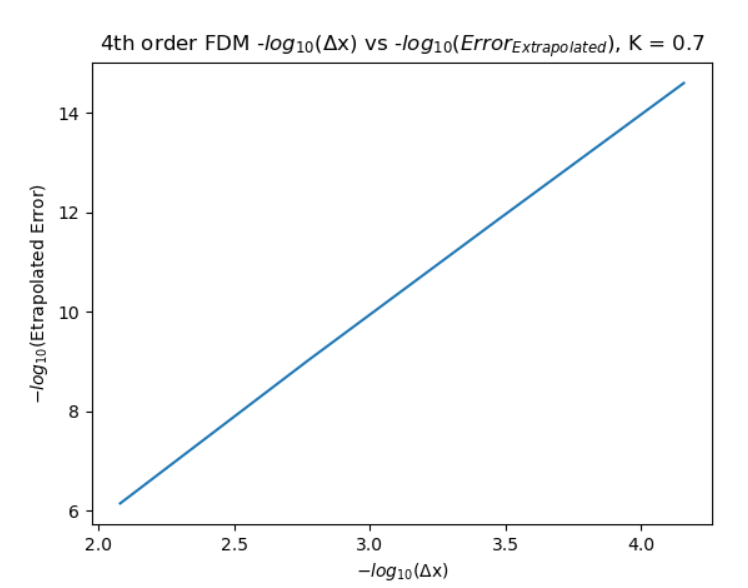


**2nd order Convergence values using Extrapolation, K = 1:**

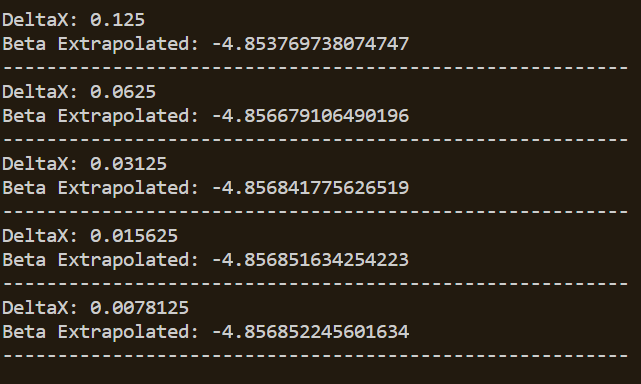
**\*smaller convergence value can be observed when comparing to K = .7**



**4th order output log() vs -log(Error from Extrapolation), K = .7:**

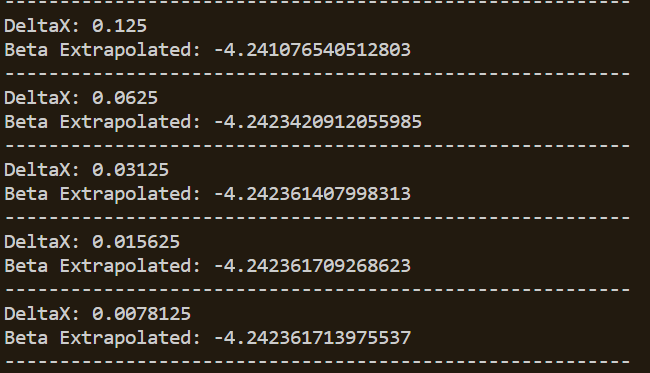


**4th order Convergence values using Extrapolation, K = .7:**



**4th order Convergence values using Extrapolation, K = 1:**

**\*smaller convergence value can be observed when comparing to K = .7**

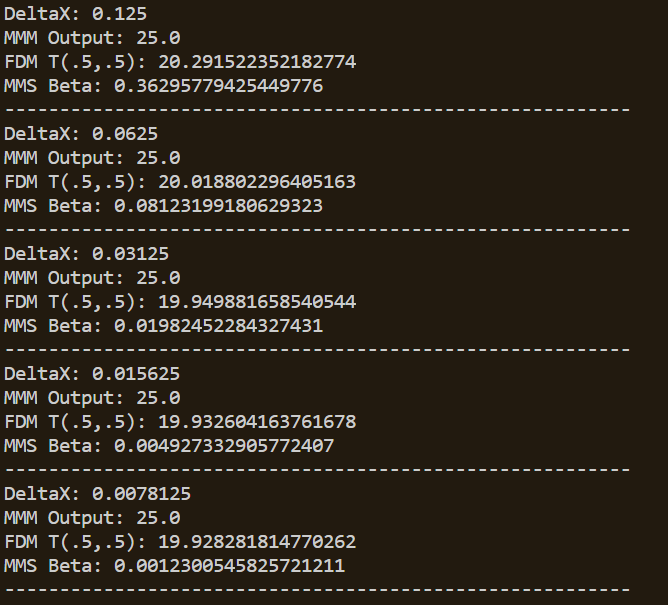
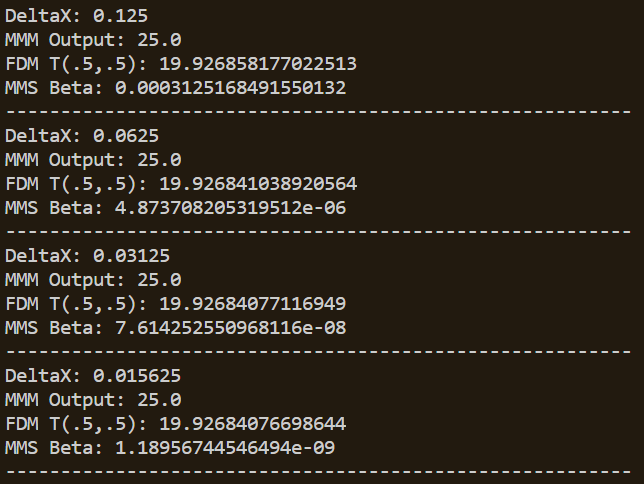


**7. Convergence of Estimated Heat Distribution Using Method of Manufactured Solutions:**

The set of boundary conditions could be approximated by the solution: T = 100\* at x = .5 cm.

This manufactured solution can be compared to the FDM output at T(.5, .5).

The following convergence values were obtained used this Method of Manufactured Solutions for 2nd and 4th order FDM comparison, K = .7:



**8. Approximating Heat Flux Across Top Surface Using 1/3 Simpson Integration:**

Analytical Solution to Total Heat Flux:

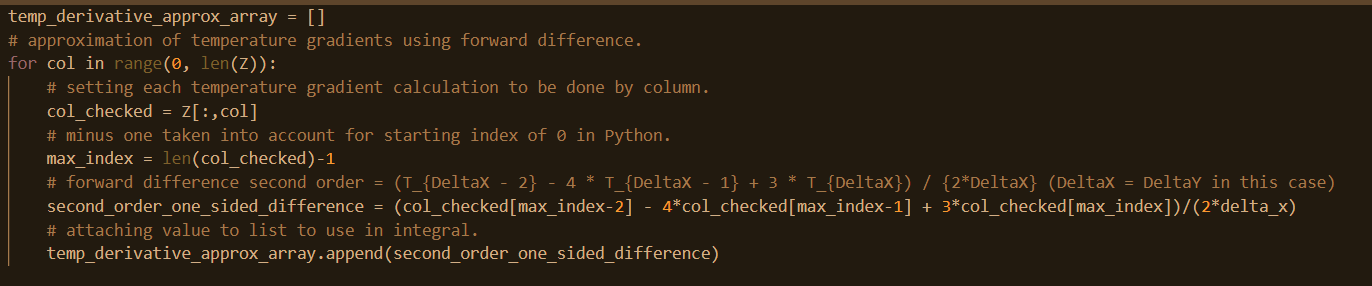
, where

This leads to . For K = 1, -200.748374639

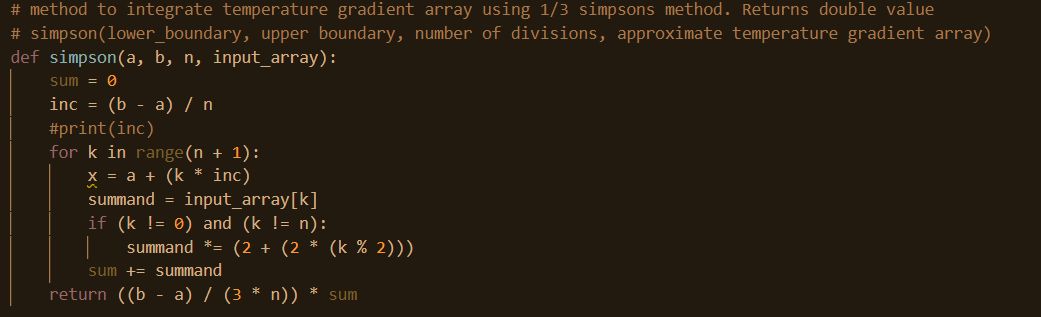
Using finite difference, this can be approximated using the forward finite difference.

For second order FDM,

For fourth order FDM,



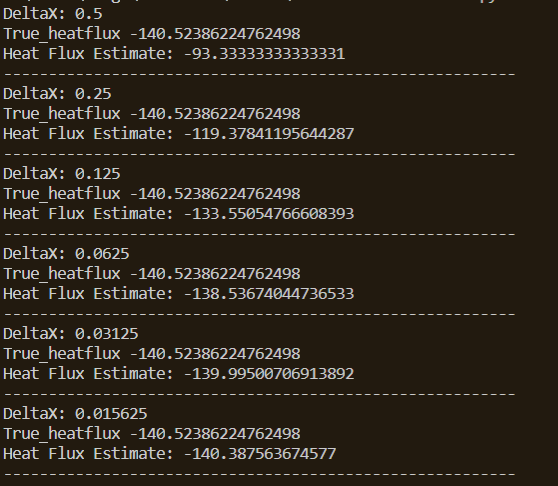
This is then integrated using the following 1/3 Simpsons method:



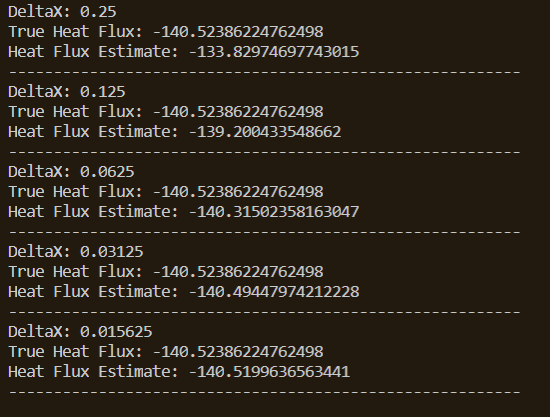
The estimated total heat flux is then printed using the code below:



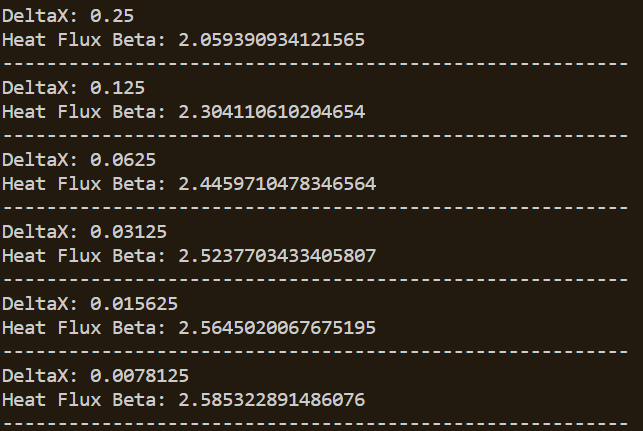
2nd order estimated heat flux from K = .7



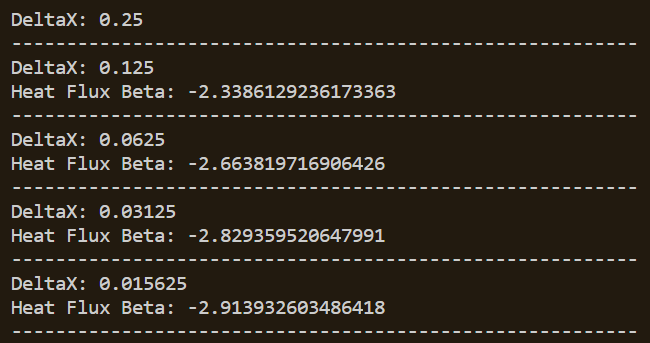
4th order estimated heat flux, K = .7:



2nd Order Heat Flux Convergence, K = .7:



4th Order Heat Flux Convergence, K = .7:



**Conclusion**:

This project made use of FDM to break down a boundary condition problem into a set of finite difference equations. This set of equations was then broken down into an assembled matrix and resultant matrix from the boundary conditions. These were then solved with linear algebra to get a matrix of temperatures representing the distribution on the surface defined.

Python was used out of convenience, however there are issues with accuracy of Python, as the computer would shut down when attempting to solve for 4th order meshes with degrees greater than 8. Extended precision packages would come in use to increase accuracy. Increasing efficiency may also be changed by avoiding the use of available Python packages for linear algebra. A programming language such as C++ might increase available size of matrices to solve for temperature distributions.

The estimated heat temperatures converged when compared to the True analytical result and converged when compared to the Method of Manufactured Solutions.

When comparing 2nd order to 4th order convergence, 4th order converged much faster than 2nd order for the same K and DeltaX values.

This assignment was useful in seeing how quickly the accuracy can improve with higher orders of FDM convergences for solutions to this set of partial differential equations along with seeing the effect of Richardson Extrapolation and Method of Manufactured Solutions to check convergence rates. ­­