Assignment 2

Texas A&M University
AERO-430-500 Numerical Simulation
Andrew Hollister

October 5th, 2021

Contents

1 Abstract

The purpose of this assignment is to formulate the analytical solution to the problem of heat conduction in a bi-material rod, a behavior governed by a second order ODE, as well as finding the total heat loss to the environment. Both analytical and finite difference method (FDM) techniques were used in the analysis of two separate cases. For the first case, the first material will range from $0 \leq x \leq 1/2$ and the second material will range from $1/2 \leq x \leq 1$. For the second case, the material will range from $0 \leq x \leq 3/4$ and the second material will range from $3/4 \leq x \leq 1$. In both cases, the boundary conditions of $T(0) = 0℃$ and $T(L) = 100℃$ were used.

2 Analytical Solution

2.1 Initialization

The analytical solution can be derived by breaking up the function over the entire bar into a piecewise function, each defining the section of the bar either side of the interface.

$$Case\ 1\text{:}\quad T(x) = \begin{cases} T_1(x) & 0 < x < 1/2 \\ T_2(x) & 1/2 < x < 1 \end{cases}$$

$$Case\ 2\text{:}\quad T(x) = \begin{cases} T_1(x) & 0 < x < 3/4 \\ T_2(x) & 3/4 < x < 1 \end{cases}$$

2.2 Piecewise Functions

In order to obtain the respective equations in the previous section, it is necessary to derive separate expressions to define each subdomain. This can be accomplished by utilizing harmonic interpolation functions.

Case 1:

$$T^+(x) = -\frac{T_0}{\sinh\left(\frac{\alpha_0}{2}\right)}\sinh\left(\alpha_0\left(x-\frac{1}{2}\right)\right) + \frac{T_{\frac{1}{2}}}{\sinh\left(\frac{\alpha_0}{2}\right)}\sinh(\alpha_0 x) \qquad 0 < x < 1/2$$

$$T^-(x) = -\frac{T_{\frac{1}{2}}}{\sinh\left(\frac{\alpha_1}{2}\right)}\sinh\left(\alpha_1\left(x-\frac{1}{2}\right)\right) + \frac{T_1}{\sinh\left(\frac{\alpha_1}{2}\right)}\sinh(\alpha_1 x) \qquad 1/2 < x < 1$$

Case 2:

$$T^+(x) = -\frac{T_0}{\sinh\left(\frac{3\alpha_0}{4}\right)}\sinh\left(\alpha_0\left(x-\frac{3}{4}\right)\right) + \frac{T_{\frac{3}{4}}}{\sinh\left(\frac{3\alpha_0}{4}\right)}\sinh(\alpha_0 x) \qquad 0 < x < 3/4$$

$$T^-(x) = -\frac{T_{\frac{3}{4}}}{\sinh\left(\frac{\alpha_1}{4}\right)}\sinh\left(\alpha_1\left(x-\frac{1}{4}\right)\right) + \frac{T_1}{\sinh\left(\frac{\alpha_1}{4}\right)}\sinh(\alpha_1 x) \qquad 3/4 < x < 1$$

Here, it is clear that there is unknown in each case. $T_{\frac{1}{2}}$ for case one, and $T_{\frac{3}{4}}$ for case two. The following sections will delve into how these values can be determined.

2.3 Boundary Conditions

The equation governing the temperature along the bar includes two boundary conditions that we can use to obtain the above unknowns.

Case 1:

1. $Q\left(\frac{1}{2}-0\right) = Q(\frac{1}{2}+0)$
   $\boldsymbol{-k_0AT_1'\left(\frac{1}{2}-0\right) = -k_1AT_2'(\frac{1}{2}+0)}$

Case 2:

1. $Q\left(\frac{1}{2}-0\right) = Q(\frac{1}{2}+0)$
   $\boldsymbol{-k_0AT_1'\left(\frac{1}{2}-0\right) = -k_1AT_2'(\frac{1}{2}+0)}$

Using these boundary conditions for each case, we can obtain the following values for the unknown in each case.

Case 1:

$$T_{\frac{1}{2}} = -\frac{T_1\alpha_1 k_1}{\sinh\left(\frac{\alpha_1}{2}\right)}\left[\frac{k_0\alpha_0^2}{\left(\frac{h}{k}\sinh^2\frac{\alpha_0}{2}-\alpha_0\sinh\left(\frac{\alpha_0}{2}\right)\cosh\left(-\frac{\alpha_0}{2}\right)\right)}-\frac{k_0\alpha_0}{\sinh\left(\frac{\alpha_0}{2}\right)}\cosh\left(\frac{\alpha_0}{2}\right)-\frac{k_1\alpha_1}{\sinh\left(\frac{\alpha_1}{2}\right)}\cosh\left(-\frac{\alpha_1}{2}\right)\right]^{-1}$$

Case 2:

$$T_{\frac{3}{4}} = -\frac{T_1\alpha_1 k_1}{\sinh\left(\frac{\alpha_1}{4}\right)}\left[\frac{k_0\alpha_0^2}{\left(\frac{h}{k}\sinh^2\frac{3a_0}{4}-\alpha_0\sinh\left(\frac{3\alpha_0}{4}\right)\cosh\left(-\frac{3\alpha_0}{4}\right)\right)}-\frac{k_0\alpha_0}{\sinh\left(\frac{3\alpha_0}{4}\right)}\cosh\left(\frac{3\alpha_0}{4}\right)-\frac{k_1\alpha_1}{\sinh\left(\frac{\alpha_1}{4}\right)}\cosh\left(-\frac{\alpha_1}{4}\right)\right]^{-1}$$

2.4 Heat Transfer

In order to derive the equation for the heat transfer at the right end of the bar, the derivative of the temperature equation of the right bar is taken at L.

Case 1:

$$\frac{dT}{dx}\,|_L = -k_1A\left[-\frac{T_{\frac{1}{2}}\alpha_1}{\sinh\left(\frac{\alpha_1}{2}\right)}+\frac{T_L\alpha_1\cosh\left(\frac{\alpha_1}{2}\right)}{\sinh\left(\frac{\alpha_1}{2}\right)}\right]$$

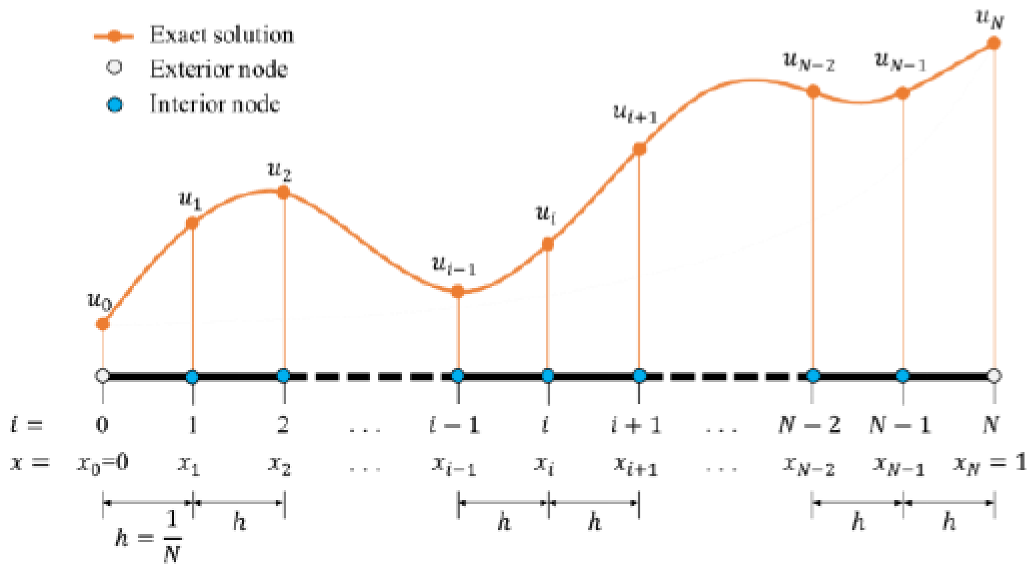
Case 2:

$$\frac{dT}{dx}\,|_L = -k_1A\left[-\frac{T_{\frac{3}{4}}\alpha_1}{\sinh\left(\frac{\alpha_1}{4}\right)}+\frac{T_L\alpha_1\cosh\left(\frac{\alpha_1}{4}\right)}{\sinh\left(\frac{\alpha_1}{4}\right)}\right]$$

3 Finite Difference Method

3.1 Discretization of the Domain

To utilize the FDM, the domain must be discretized into N+1 evenly spaced points. As shown in the figure below, the discretized domain is x_i for $i = 0, 1, \dots, N$ with the spacing between the nodes $\Delta x = h$ and $u_i = u(x_i)$.



3.2 Taylor Series Derivation

From Taylor Series, the following is derived:

$$u(x_{i+1}) = u(x_i + dx) = u(x_i) + dx u'(x_i) + \frac{dx^2}{2}u''(x_i) + \frac{dx^3}{6}u'''(x_i) + \frac{dx^4}{24}u^{(4)}(x_i) + \dots$$

$$u(x_{i-1}) = u(x_i - dx) = u(x_i) - dx u'(x_i) + \frac{dx^2}{2}u''(x_i) - \frac{dx^3}{6}u'''(x_i) + \frac{dx^4}{24}u^{(4)}(x_i) - \dots$$

Adding these equations yields:

$$u(x_{i+1}) + u(x_{i-1}) = 2u(x_i) + dx^2 u''(x_i) + \frac{dx^4}{12}u^{(4)}(x_i) + \dots$$

Solving for $u''(x_i)$ yields:

$$u''(x_i) = \frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1})}{dx^2} + \mathcal{O}(dx^2)$$

By neglecting the truncation error of the second order Taylor Series expansion, we can obtain an approximate value for $u''(x_i)$ which we denote by $\bar{u}_i''$ as shown below:

$$u''(x_i) \approx \bar{u}_i'' = \frac{\bar{u}_{i-1} - 2\bar{u}_i + \bar{u}_{i+1}}{dx^2}$$

Substituting this approximation into Equation 1 yields the following set of equations:

$$\mp\frac{\bar{u}_0 - 2\bar{u}_1 + \bar{u}_2}{dx^2} + w^2\bar{u}_1 = 0$$

$$\mp\frac{\bar{u}_1 - 2\bar{u}_2 + \bar{u}_3}{dx^2} + w^2\bar{u}_2 = 0$$

$$\mp\frac{\bar{u}_2 - 2\bar{u}_3 + \bar{u}_4}{dx^2} + w^2\bar{u}_3 = 0$$

$$\vdots$$

$$\mp\frac{\bar{u}_{N-2} - 2\bar{u}_{N-1} + \bar{u}_N}{dx^2} + w^2\bar{u}_{N-1} = 0$$

If these are now multiplied by our h^2, our equations take on the form:

$$\mp(\bar{u}_{i-1} - 2\bar{u}_i + \bar{u}_{i+1}) + w^2dx^2\bar{u}_i = 0$$

$$\mp\bar{u}_{i-1} \pm 2\bar{u}_i \mp \bar{u}_{i+1} + w^2dx^2\bar{u}_i = 0$$

$$\mp\bar{u}_{i-1} + (\pm 2 + w^2dx^2)\bar{u}_i \mp \bar{u}_{i+1} = 0$$

If these are now multiplied by $\mp$, this yields:

$$-\bar{u}_{i-1} + (2 \pm w^2dx^2)\bar{u}_i - \bar{u}_{i+1} = 0$$

The specified boundary conditions are $u(0) = 0$ and $u(1) = 100$. If we let $\kappa = 2 \mp \omega^2dx^2$, then the above expressions can be represented by a matrix below. This matrix can then calculate the unknowns $\overline{u_1}, \overline{u_2}, \ \overline{u_3}, \ldots \ \overline{u_{N-1}}$.

$$\begin{bmatrix} \kappa & -1 & 0 & \ldots & 0 & 0 \\ -1 & \kappa & -1 & \ldots & 0 & 0 \\ 0 & -1 & \kappa & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & -1 & 0 \\ 0 & 0 & 0 & -1 & \kappa & -1 \\ 0 & 0 & 0 & 0 & -1 & \kappa \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \bar{u}_3 \\ \vdots \\ \bar{u}_{N-2} \\ \bar{u}_{N-1} \end{Bmatrix} = \begin{Bmatrix} \bar{u}_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \bar{u}_L \end{Bmatrix}$$

Thus, values for all the nodes, $i = 0, 1, \ldots, N$, and the intermediate values can be obtained by performing interpolation.

3.3 Bi-material Considerations

However, since this is a bi-material bar, some additional modifications must be made to the original matrix equation above. The first segment of the bar will have parameters denoted with a subscript of 0 and the second segment of the bar will have parameters denoted with a subscript of 1. The third and most important medication that must be made is that continuity must be kept at the interface between the two halves. The first segment of the matrix will use the same matrix formulation and will use $\kappa_0 = 2 \mp \omega_0^2dx^2$. Here $\omega_0^2 = \frac{hP}{k_0A}$. The second segment of the bar will also use the same matrix formulation and will have $\kappa_0 = 2 \mp \omega_0^2dx^2$. Here $\omega_0^2 = \frac{hP}{k_0A}$.

At the interface, it is known that $Q\left(\frac{1}{2}-0\right) = Q(\frac{1}{2}+0)$, where $Q\left(\frac{1}{2}-0\right) = -k_0 A u_0'\left(\frac{1}{2}-0\right)$ and $Q\left(\frac{1}{2}+0\right) = -k_1 A u_1'\left(\frac{1}{2}+0\right)$. From here, $u_{0,\left(\frac{N}{2}-1\right)}$ can be approximated.

$$u_0'\left(\frac{1}{2}-0\right) = \frac{Q_{\frac{1}{2}}}{-k_0 A}$$

$$\frac{u_{0,(\frac{N}{2}+1)} - u_{0,(\frac{N}{2}-1)}}{\Delta x} = -\frac{Q_{\frac{1}{2}}}{k_0 A}$$

$$u_{0,(\frac{N}{2}+1)} = u_{0,(\frac{N}{2}-1)} - \frac{Q_{\frac{1}{2}}\Delta x}{k_0 A}$$

The original differential equation can also be rewritten in the form $-k_i A u_i'' + hPu_i = 0$. Combining this equation with the second order approximation yields:

$$-k_0 A u_0''(1/2) + hPu(1/2) = 0$$

$$-k_0 A \frac{u_{0(\frac{N}{2}-1)} - 2u_{0(\frac{N}{2})} + u_{0(\frac{N}{2}+1)}}{\Delta x^2} + hPu_{0(\frac{N}{2})} = 0$$

Plugging in the previous equation yields:

$$-k_0 A \frac{u_{0(\frac{N}{2}-1)} - 2u_{0(\frac{N}{2})} + u_{0,(\frac{N}{2}-1)} - \frac{Q_{\frac{1}{2}}\Delta x}{k_0 A}}{\Delta x^2} + hPu_{0(\frac{N}{2})} = 0$$

Rearranging and simplifying yields:

$$-k_0 A \frac{2(u_{0(\frac{N}{2}-1)} - u_{0(\frac{N}{2})})}{\Delta x^2} + hPu_{0(\frac{N}{2})} = \frac{Q_{\frac{1}{2}}}{\Delta x}$$

For the other segment of the bar:

$$-k_1 A \frac{2(u_{1(\frac{N}{2}+1)} - u_{1(\frac{N}{2})})}{\Delta x^2} + hPu_{1(\frac{N}{2})} = -\frac{Q_{\frac{1}{2}}}{\Delta x}$$

Adding the two equations and enforcing continuity yields:

$$-\frac{k_0 A}{\Delta x} u_{\frac{N}{2}-1} + \left(\frac{k_0 A}{\Delta x} + \frac{k_1 A}{\Delta x} + hP\Delta x\right) u_{\frac{N}{2}} - \frac{k_1 A}{\Delta x} u_{\frac{N}{2}+1} = 0$$

If $k_0 = k_1$ then the same results will be obtained as before. Now that the interface equation has been established, a proper matrix equation for bi-material heat conduction can be created.

$$\begin{bmatrix} \kappa_0 & -1 & 0 & \dots & \dots & 0 & 0 \\ -1 & \kappa_0 & -1 & 0 & \dots & 0 & 0 \\ 0 & -1 & \ddots & -1 & 0 & 0 & 0 \\ 0 & 0 & \frac{-k_0 A}{\Delta x} & \left(\frac{k_0 A}{\Delta x} + \frac{k_1 A}{\Delta x} + hP\Delta x\right) & \frac{-k_1 A}{\Delta x} & 0 & 0 \\ 0 & 0 & 0 & -1 & \ddots & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & \kappa_1 & -1 \\ 0 & 0 & \dots & \dots & 0 & -1 & \kappa_1 \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \vdots \\ \bar{u}_{\frac{N}{2}} \\ \vdots \\ \bar{u}_{N-2} \\ \bar{u}_{N-1} \end{Bmatrix} = \begin{Bmatrix} \bar{u}_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \bar{u}_L \end{Bmatrix}$$

In order to derive the matrix for the interface at x=3L/4, some modifications to the above matrix can be made:

$$\begin{bmatrix} \kappa_0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & \kappa_0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & \kappa_0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & \ddots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \kappa_0 & -1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & -1 & \kappa_0 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -\frac{k_0 A}{\Delta x} & (\frac{k_0 A}{\Delta x} + \frac{k_1 A}{\Delta x} + hP\Delta X) & -\frac{k_1 A}{\Delta x} \\ 0 & 0 & 0 & \cdots & 0 & 0 & -1 & \kappa_1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_{\frac{3N}{4}} \\ u_{N-1} \end{Bmatrix} = \begin{Bmatrix} u_0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ u_L \end{Bmatrix}$$

3.4 Heat Transfer

The heat transfer in the bar can be obtained by calculating the derivate of the right end of the bar at x = L. This operation can be performed with the following expression:

$$Q_n = \frac{-k_1 A U_{N-1} + k_1 A + \frac{\Delta x^2}{2} hPU_N}{\Delta x}$$

4 Performance Analysis

4.1 Error

The errors of the numerical solutions are calculated using the equations below:

$$e_h = u(x) - u_h(x)$$

4.2 Percent Error

The percent error of an estimated quantity $Q_{Estimated}$ (calculated using FDM) against its exact values Q_{Exact} is calculated using the equation below:

$$\%Error = \left| \frac{Q_{Exact} - Q_{Estimated}}{Q_{Exact}} \right| \times 100\%$$

4.3 Extrapolation and Convergence

Richardson's Extrapolation was used to extrapolate an approximate of the exact value from a series of approximated values. In general, error is modeled as:

$$Q_{ex} - Q_h = Ch^\beta$$

Where Q is the quantity of interest, Q_h is the approximate value at some mesh size h, C is some constant, and β is the convergence rate. In general, it is rare for the exact value to be known, and it is often difficult or impossible to obtain analytical solutions. In this case it is possible to use Richardson's Extrapolation to obtain reasonably accurate approximate value of the exact solution. If we write this equation at another mesh size, say h/2, the two can be divided and the unknown β can be found.

$$Q_{ex} - Q_h = C(h)^\beta$$

$$Q_{ex} - Q_{\frac{h}{2}} = C\left(\frac{h}{2}\right)^\beta$$

$$\frac{Q_{ex} - Q_h}{Q_{ex} - Q_{\frac{h}{2}}} = \frac{C(h)^\beta}{C(\frac{h}{2})^\beta}$$

$$log\left(\frac{Q_{ex} - Q_h}{Q_{ex} - Q_{\frac{h}{2}}}\right) = log\left(\frac{C(h)^\beta}{C(\frac{h}{2})^\beta}\right)$$

$$log(Q_{ex} - Q_h) - log(Q_{ex} - Q_{\frac{h}{2}}) = \beta log(h) - \beta log\left(\frac{h}{2}\right)$$

$$\boxed{\beta = \frac{log(Q_{ex} - Q_h) - log(Q_{ex} - Q_{\frac{h}{2}})}{log(h) - log(\frac{h}{2})}}$$

Again, Richardson's Extrapolation will be used to derive an expression for an extrapolated value. Here we will have to utilize three mesh sizes rather than the previous two.

$$Q_{ex} - Q_h = C(h)^\beta \approx 2^\beta$$

$$Q_{ex} - Q_{\frac{h}{2}} = C\left(\frac{h}{2}\right)^\beta \approx 2^\beta$$

$$Q_{ex} - Q_{\frac{h}{4}} = C\left(\frac{h}{4}\right)^\beta \approx 2^\beta$$

$$\frac{Q_{ex} - Q_h}{Q_{ex} - Q_{\frac{h}{2}}} \approx 2^\beta \approx \frac{Q_{ex} - Q_{\frac{h}{2}}}{Q_{ex} - Q_{\frac{h}{4}}}$$

$$\frac{Q_{extr} - Q_h}{Q_{extr} - Q_{\frac{h}{2}}} = 2^\beta = \frac{Q_{extr} - Q_{\frac{h}{2}}}{Q_{extr} - Q_{\frac{h}{4}}}$$

$$(Q_{extr} - Q_h)(Q_{extr} - Q_{\frac{h}{4}}) = (Q_{extr} - Q_{\frac{h}{2}})^2$$

$$\boxed{Q_{extr} = \frac{Q_{\frac{h}{2}}^2 - Q_h * Q_{\frac{h}{4}}}{2Q_{\frac{h}{2}} - Q_h - Q_{\frac{h}{4}}}}$$

From here Q_{extra} can be substituted to solve for β, which yields:

$$\boxed{\frac{log\left(\frac{Q_{extr} - Q_h}{Q_{extr} - Q_{\frac{h}{2}}}\right)}{log(2)} = \beta = \frac{log\left(\frac{Q_{extr} - Q_{\frac{h}{2}}}{Q_{extr} - Q_{\frac{h}{4}}}\right)}{log(2)}}$$

5 Results

In this section, the results of the analytical solution and the FDM solution for the interface located at x = L/2 and x = 3L/4 are provided. For each case, the analysis was conducted for mesh sizes varying as following: [8, 16, 32, 64, 128, 256, 512, 1024] and k_1 values varying as follows: $k_1 = k_0[0.01, 0.1, 1, 10, 100]$. Once again, the temperatures at the left end of the two segmented bar was kept at a constant value of 0℃, while the right end of the bar was kept at 100℃. The radius of the bar was set to 0.1 cm, h = 0.4, $k_0 = 0.5$, and α_1=4.

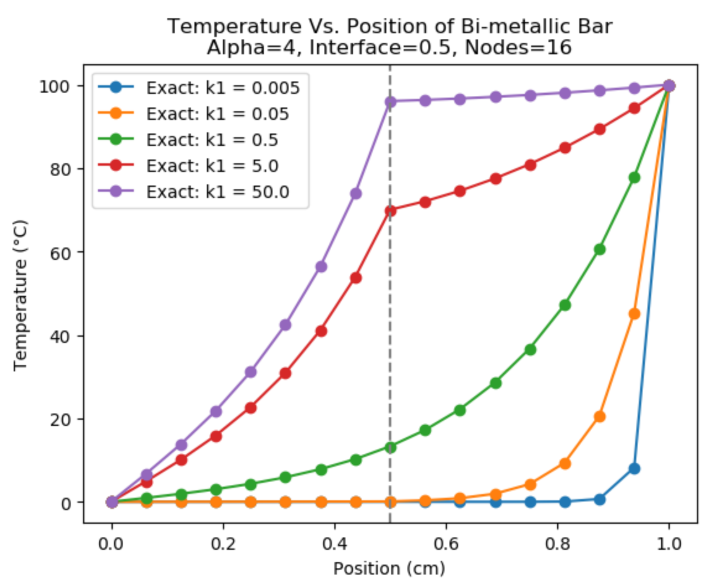
5.1 Analytical Results

The exact solutions for varying c values are presented below along a range from $x = 0\ to\ x = 1$. The table below provides the exact numerical values shown in the figure below. The results displayed correspond for a 16-node analysis. Additional data performed for each mesh size can be found in the appropriate section in the appendix section of this report.

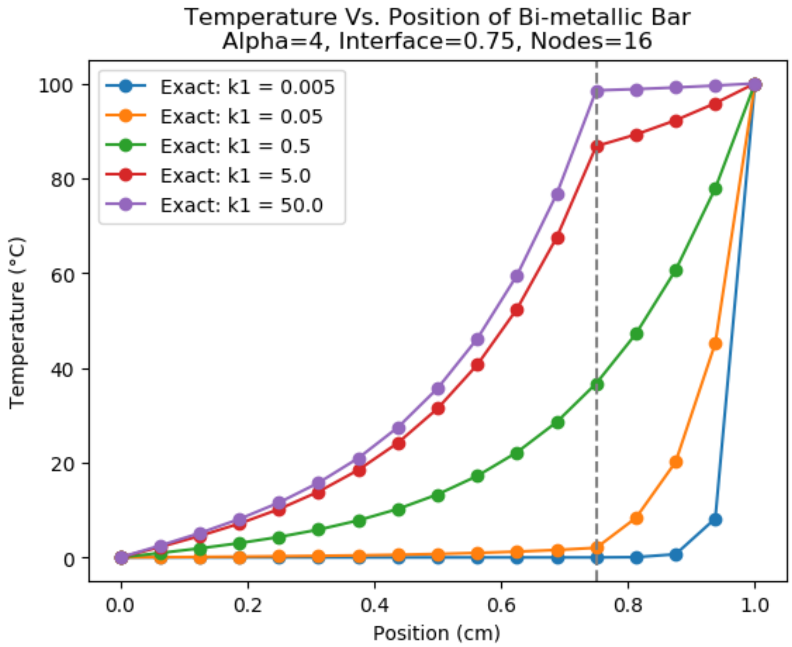
5.1.1 Case 1: Interface Located at L/2

Temperature Vs. Position Tables for Exact Analysis
Nodes = 16

Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50	T, k1 = 500
0	0	0	0	0	0
0.0625	2.52455e-09	0.00583127	0.925662	4.87636	6.69103
0.125	5.2077e-09	0.0120289	1.90948	10.0591	13.8024
0.1875	8.21803e-09	0.0189822	3.01326	15.8738	21.781
0.25	1.17447e-08	0.0271282	4.30636	22.6857	31.128
0.3125	1.60092e-08	0.0369785	5.87	30.923	42.4306
0.375	2.12795e-08	0.049152	7.80244	41.103	56.399
0.4375	2.78867e-08	0.0644135	10.2251	53.8654	73.9107
0.5	3.6246e-08	0.083722	13.2901	70.0119	96.0659
0.5625	2.49706e-06	0.351725	17.1901	72.0483	96.3451
0.625	3.05891e-05	0.851247	22.1701	74.5352	96.6845
0.6875	0.000372665	1.91109	28.543	77.4882	97.0843
0.75	0.00453999	4.22889	36.7091	80.9258	97.5448
0.8125	0.0553084	9.3303	47.1815	84.8694	98.0663
0.875	0.673795	20.5733	60.6181	89.3438	98.6491
0.9375	8.2085	45.3583	77.8631	94.3768	99.2935
1	100	100	100	100	100



5.1.2 Case 2: Interface Located at 3L/4



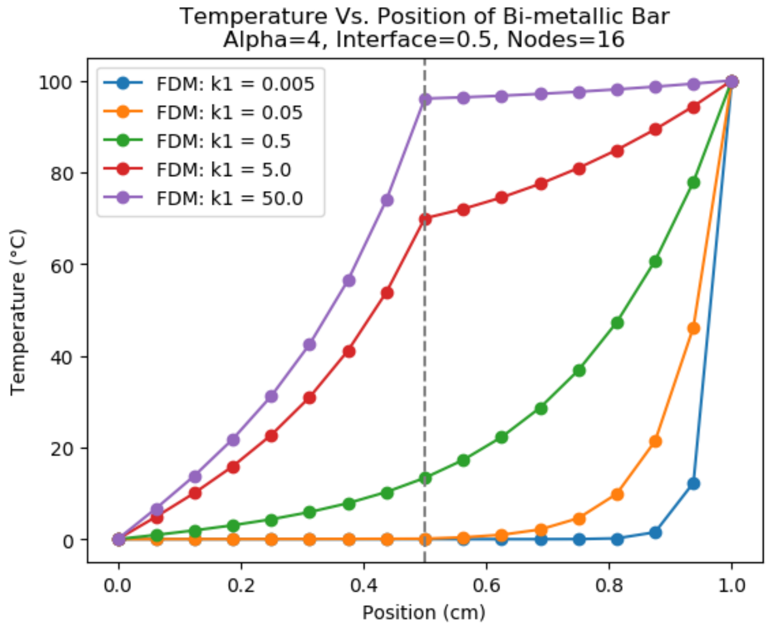
Temperature Vs. Position Tables for Exact Analysis
Nodes = 16

Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50	T, k1 = 500
0	0	0	0	0	0
0.0625	2.07211e-05	0.0511431	0.925662	2.18777	2.48418
0.125	4.27441e-05	0.105499	1.90948	4.51298	5.12443
0.1875	6.74525e-05	0.166484	3.01326	7.12173	8.08662
0.25	9.63987e-05	0.237928	4.30636	10.1779	11.5569
0.3125	0.000131401	0.32432	5.87	13.8735	15.7532
0.375	0.000174659	0.431087	7.80244	18.4408	20.9392
0.4375	0.000228891	0.564939	10.2251	24.1666	27.4408
0.5	0.000297502	0.734283	13.2901	31.4107	35.6664
0.5625	0.000384805	0.94976	17.1901	40.6282	46.1327
0.625	0.000496283	1.22491	22.1701	52.3982	59.4974
0.6875	0.000638941	1.57701	28.543	67.4602	76.6001
0.75	0.000821741	2.02819	36.7091	86.7605	98.5153
0.8125	0.0550032	8.33901	47.1815	89.2138	98.7936
0.875	0.67377	20.1389	60.6181	92.225	99.1337
0.9375	8.2085	45.1949	77.8631	95.8129	99.5358
1	100	100	100	100	100

5.2 Finite Difference Method Results

The finite difference method solutions for varying c values are presented below along a range from $x = 0 \ to \ x = 1$. The table below provides the exact numerical values shown in the figure below. The results displayed correspond for a 16-node analysis. Additional data performed for each mesh size can be found in the appropriate section in the appendix of this report.

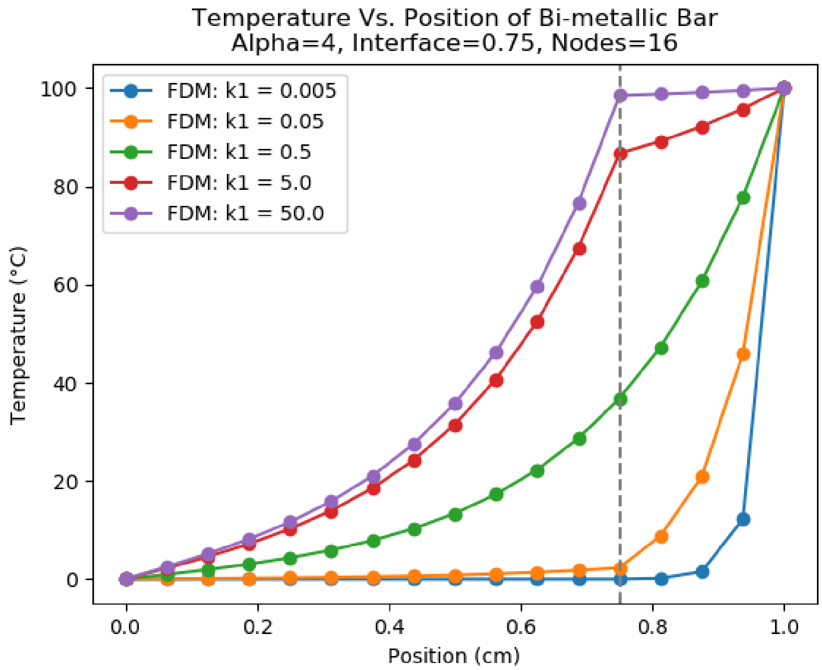
5.2.1 Case 1: Interface Located at L/2



Temperature Vs. Position Tables for FDM Analysis
Nodes = 16

Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50	T, k1 = 500
0	0	0	0	0	0
0.0625	9.74424e-08	0.00716406	0.932824	4.88467	6.7082
0.125	2.00975e-07	0.0147759	1.92395	10.0746	13.8357
0.1875	3.17069e-07	0.0233112	3.03532	15.8942	21.8279
0.25	4.52979e-07	0.0333034	4.3364	22.7073	31.1843
0.3125	6.172e-07	0.0453771	5.90851	30.9395	42.4897
0.375	8.19997e-07	0.0602869	7.8499	41.1054	56.4508
0.4375	1.07404e-06	0.0789646	10.2819	53.8404	73.94
0.5	1.39522e-06	0.102578	13.3565	69.9405	96.0505
0.5625	4.22327e-05	0.402819	17.2659	71.9876	96.3316
0.625	0.000347025	0.954821	22.2545	74.4846	96.673
0.6875	0.00282072	2.10359	28.6339	77.4472	97.0747
0.75	0.0229239	4.5671	36.803	80.8939	97.5372
0.8125	0.186302	9.88504	47.2722	84.8461	98.0606
0.875	1.51407	21.3811	60.6959	89.3286	98.6453
0.9375	12.3047	46.2404	77.9132	94.3694	99.2916
1	100	100	100	100	100

5.2.2 Case 2: Interface Located at 3L/4



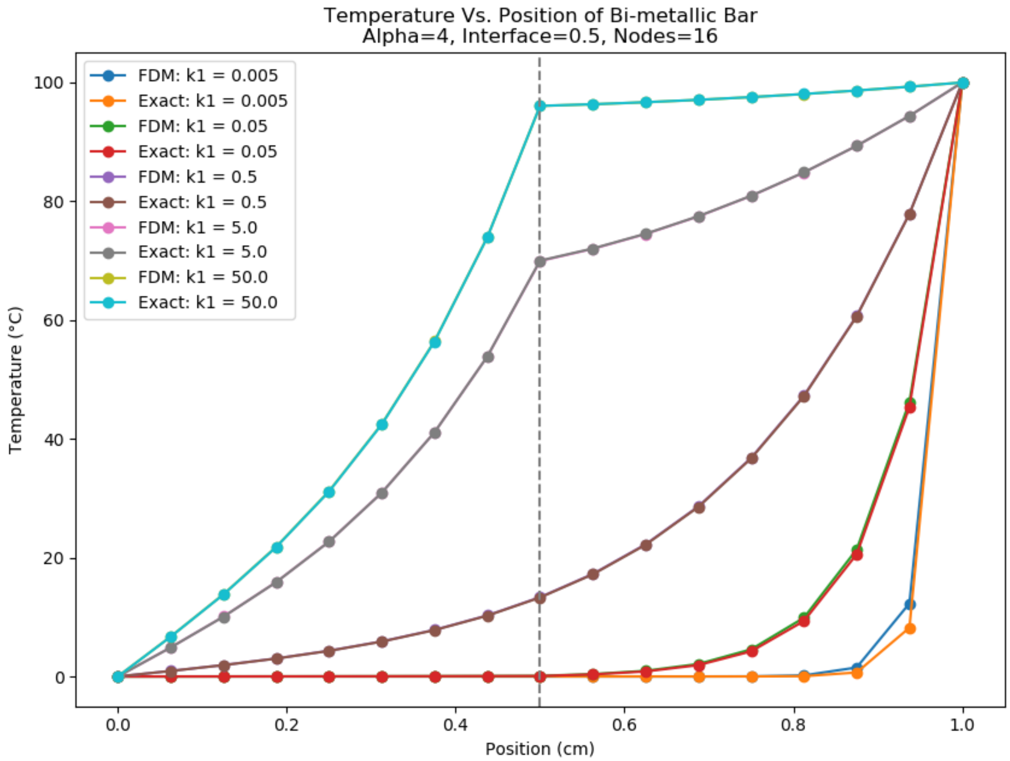
Temperature Vs. Position Tables for FDM Analysis
Nodes = 16

Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50	T, k1 = 500
0	0	0	0	0	0
0.0625	0.000158595	0.0583121	0.932824	2.1978	2.49682
0.125	0.000327102	0.120269	1.92395	4.53295	5.1497
0.1875	0.000516054	0.189742	3.03532	7.15142	8.12443
0.25	0.000737258	0.271075	4.3364	10.2169	11.6069
0.3125	0.00100454	0.369349	5.90851	13.9208	15.8149
0.375	0.00133461	0.490708	7.8499	18.4949	21.0112
0.4375	0.00174809	0.642736	10.2819	24.2248	27.5208
0.5	0.00227082	0.834935	13.3565	31.4689	35.7504
0.5625	0.00293549	1.07932	17.2659	40.6797	46.2145
0.625	0.00378362	1.39116	22.2545	52.433	59.5669
0.6875	0.00486822	1.78994	28.6339	67.4634	76.6423
0.75	0.00625709	2.3006	36.803	86.7102	98.5078
0.8125	0.184251	8.84507	47.2722	89.1768	98.788
0.875	1.51381	20.9177	60.6959	92.2008	99.1299
0.9375	12.3047	46.0639	77.9132	95.801	99.5339
1	100	100	100	100	100

5.3 FDM Compared to Analytical Results

The finite difference method results compared to the exact solution are presented below along a range from $x = 0\ to\ x = 1$. The table below provides the exact numerical values shown in the figure below. The results displayed correspond for a 16-node analysis. Additional data performed for each mesh size can be found in the appropriate section in the appendix section of this report.

5.3.1 Case 1: Interface Located at L/2



As can be noted from the graph above, the exact and FDM solutions align very closely. A more detailed comparison of these values are listed on the next page in the form of tables for each value of k_1.

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.005 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	2.52455e-09	0.0625	9.74424e-08
0.125	5.2077e-09	0.125	2.00975e-07
0.1875	8.21803e-09	0.1875	3.17069e-07
0.25	1.17447e-08	0.25	4.52979e-07
0.3125	1.60092e-08	0.3125	6.172e-07
0.375	2.12795e-08	0.375	8.19997e-07
0.4375	2.78867e-08	0.4375	1.07404e-06
0.5	3.6246e-08	0.5	1.39522e-06
0.5625	2.49706e-06	0.5625	4.22327e-05
0.625	3.05891e-05	0.625	0.000347025
0.6875	0.000372665	0.6875	0.00282072
0.75	0.00453999	0.75	0.0229239
0.8125	0.0553084	0.8125	0.186302
0.875	0.673795	0.875	1.51407
0.9375	8.2085	0.9375	12.3047
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.05 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	0.00583127	0.0625	0.00716406
0.125	0.0120289	0.125	0.0147759
0.1875	0.0189822	0.1875	0.0233112
0.25	0.0271282	0.25	0.0333034
0.3125	0.0369785	0.3125	0.0453771
0.375	0.049152	0.375	0.0602869
0.4375	0.0644135	0.4375	0.0789646
0.5	0.083722	0.5	0.102578
0.5625	0.351725	0.5625	0.402819
0.625	0.851247	0.625	0.954821
0.6875	1.91109	0.6875	2.10359
0.75	4.22889	0.75	4.5671
0.8125	9.3303	0.8125	9.88504
0.875	20.5733	0.875	21.3811
0.9375	45.3583	0.9375	46.2404
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.5 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	0.925662	0.0625	0.932824
0.125	1.90948	0.125	1.92395
0.1875	3.01326	0.1875	3.03532
0.25	4.30636	0.25	4.3364
0.3125	5.87	0.3125	5.90851
0.375	7.80244	0.375	7.8499
0.4375	10.2251	0.4375	10.2819
0.5	13.2901	0.5	13.3565
0.5625	17.1901	0.5625	17.2659
0.625	22.1701	0.625	22.2545
0.6875	28.543	0.6875	28.6339
0.75	36.7091	0.75	36.803
0.8125	47.1815	0.8125	47.2722
0.875	60.6181	0.875	60.6959
0.9375	77.8631	0.9375	77.9132
1	100	1	100

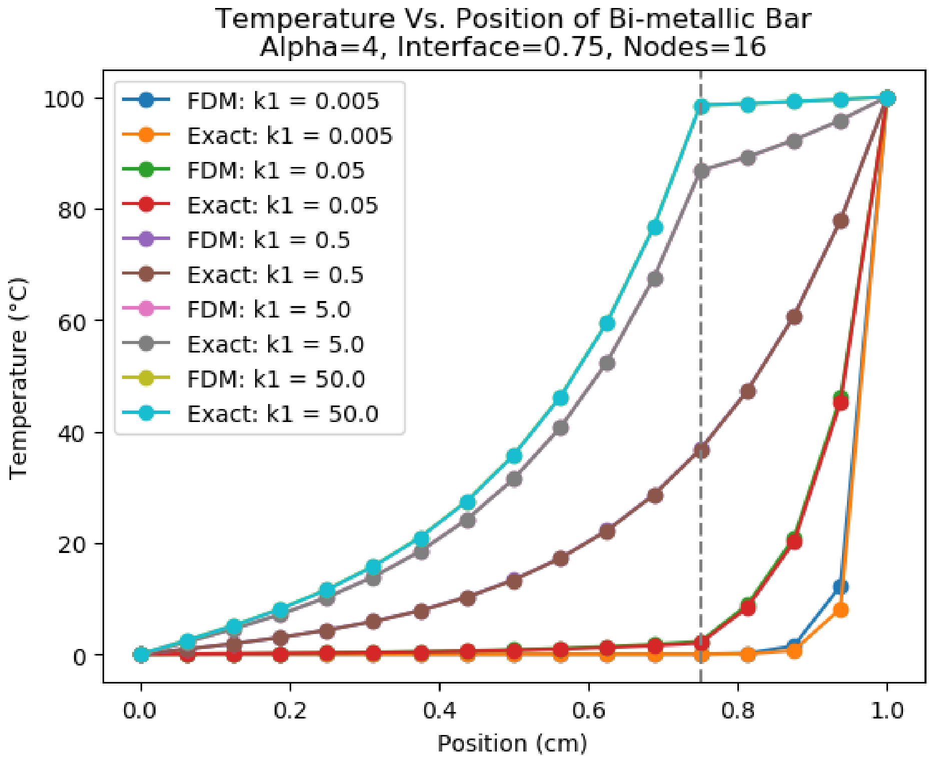
Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 5.0 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	4.87636	0.0625	4.88467
0.125	10.0591	0.125	10.0746
0.1875	15.8738	0.1875	15.8942
0.25	22.6857	0.25	22.7073
0.3125	30.923	0.3125	30.9395
0.375	41.103	0.375	41.1054
0.4375	53.8654	0.4375	53.8404
0.5	70.0119	0.5	69.9405
0.5625	72.0483	0.5625	71.9876
0.625	74.5352	0.625	74.4846
0.6875	77.4882	0.6875	77.4472
0.75	80.9258	0.75	80.8939
0.8125	84.8694	0.8125	84.8461
0.875	89.3438	0.875	89.3286
0.9375	94.3768	0.9375	94.3694
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 50.0 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	6.69103	0.0625	6.7082
0.125	13.8024	0.125	13.8357
0.1875	21.781	0.1875	21.8279
0.25	31.128	0.25	31.1843
0.3125	42.4306	0.3125	42.4897
0.375	56.399	0.375	56.4508
0.4375	73.9107	0.4375	73.94
0.5	96.0659	0.5	96.0505
0.5625	96.3451	0.5625	96.3316
0.625	96.6845	0.625	96.673
0.6875	97.0843	0.6875	97.0747
0.75	97.5448	0.75	97.5372
0.8125	98.0663	0.8125	98.0606
0.875	98.6491	0.875	98.6453
0.9375	99.2935	0.9375	99.2916
1	100	1	100

5.3.2 Case 2: Interface Located at 3L/4



As can be noted from the graph above, the exact and FDM solutions align very closely. A more detailed comparison of these values are listed on the next page in the form of tables for each value of k_1. The translation of the interface location has had a major impact in the heat transfer within the bar, as expected.

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.005 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	2.07211e-05	0.0625	0.000158595
0.125	4.27441e-05	0.125	0.000327102
0.1875	6.74525e-05	0.1875	0.000516054
0.25	9.63987e-05	0.25	0.000737258
0.3125	0.000131401	0.3125	0.00100454
0.375	0.000174659	0.375	0.00133461
0.4375	0.000228891	0.4375	0.00174809
0.5	0.000297502	0.5	0.00227082
0.5625	0.000384805	0.5625	0.00293549
0.625	0.000496283	0.625	0.00378362
0.6875	0.000638941	0.6875	0.00486822
0.75	0.000821741	0.75	0.00625709
0.8125	0.0550032	0.8125	0.184251
0.875	0.67377	0.875	1.51381
0.9375	8.2085	0.9375	12.3047
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.05 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	0.0511431	0.0625	0.0583121
0.125	0.105499	0.125	0.120269
0.1875	0.166484	0.1875	0.189742
0.25	0.237928	0.25	0.271075
0.3125	0.32432	0.3125	0.369349
0.375	0.431087	0.375	0.490708
0.4375	0.564939	0.4375	0.642736
0.5	0.734283	0.5	0.834935
0.5625	0.94976	0.5625	1.07932
0.625	1.22491	0.625	1.39116
0.6875	1.57701	0.6875	1.78994
0.75	2.02819	0.75	2.3006
0.8125	8.33901	0.8125	8.84507
0.875	20.1389	0.875	20.9177
0.9375	45.1949	0.9375	46.0639
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.5 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	0.925662	0.0625	0.932824
0.125	1.90948	0.125	1.92395
0.1875	3.01326	0.1875	3.03532
0.25	4.30636	0.25	4.3364
0.3125	5.87	0.3125	5.90851
0.375	7.80244	0.375	7.8499
0.4375	10.2251	0.4375	10.2819
0.5	13.2901	0.5	13.3565
0.5625	17.1901	0.5625	17.2659
0.625	22.1701	0.625	22.2545
0.6875	28.543	0.6875	28.6339
0.75	36.7091	0.75	36.803
0.8125	47.1815	0.8125	47.2722
0.875	60.6181	0.875	60.6959
0.9375	77.8631	0.9375	77.9132
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 5.0 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	2.18777	0.0625	2.1978
0.125	4.51298	0.125	4.53295
0.1875	7.12173	0.1875	7.15142
0.25	10.1779	0.25	10.2169
0.3125	13.8735	0.3125	13.9208
0.375	18.4408	0.375	18.4949
0.4375	24.1666	0.4375	24.2248
0.5	31.4107	0.5	31.4689
0.5625	40.6282	0.5625	40.6797
0.625	52.3982	0.625	52.433
0.6875	67.4602	0.6875	67.4634
0.75	86.7605	0.75	86.7102
0.8125	89.2138	0.8125	89.1768
0.875	92.225	0.875	92.2008
0.9375	95.8129	0.9375	95.801
1	100	1	100

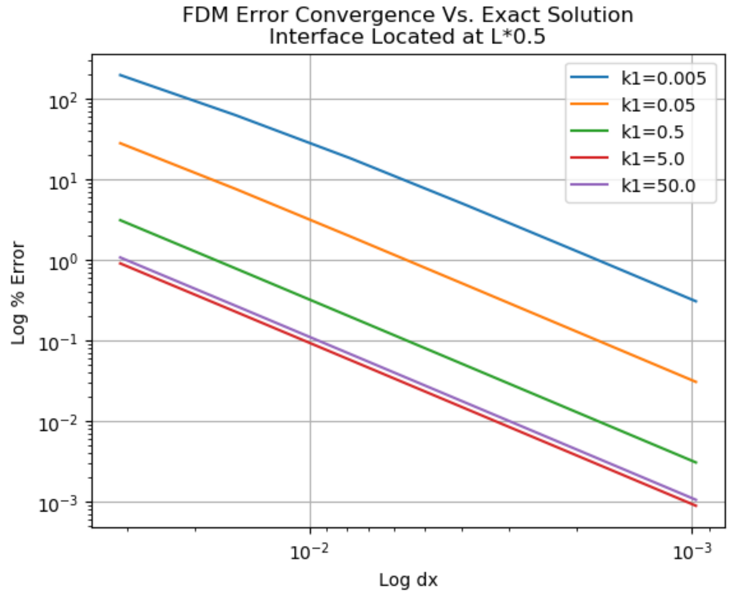
Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 50.0 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	2.48418	0.0625	2.49682
0.125	5.12443	0.125	5.1497
0.1875	8.08662	0.1875	8.12443
0.25	11.5569	0.25	11.6069
0.3125	15.7532	0.3125	15.8149
0.375	20.9392	0.375	21.0112
0.4375	27.4408	0.4375	27.5208
0.5	35.6664	0.5	35.7504
0.5625	46.1327	0.5625	46.2145
0.625	59.4974	0.625	59.5669
0.6875	76.6001	0.6875	76.6423
0.75	98.5153	0.75	98.5078
0.8125	98.7936	0.8125	98.788
0.875	99.1337	0.875	99.1299
0.9375	99.5358	0.9375	99.5339
1	100	1	100

5.4 Convergence of FDM Against the Exact Solution

The following graphs and tables show the convergence of the FDM solution with the exact solution.

5.4.1 Case 1: Interface Located at L/2



Convergence using exact solution with k1 = 0.005

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-1.6918	-0.628319	169.258	nan
0.125	-1.6918	-0.628319	169.258	1.49431
0.0625	-1.0058	-0.628319	60.0781	1.74489
0.03125	-0.740943	-0.628319	17.9248	1.91017
0.015625	-0.658284	-0.628319	4.76909	1.97473
0.0078125	-0.635942	-0.628319	1.21334	1.99347
0.00390625	-0.630233	-0.628319	0.304712	1.99835
0.00195312	-0.628798	-0.628319	0.0762649	1.99959

Convergence using exact solution with k1 = 0.05

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-2.53285	-1.98692	27.4761	nan
0.125	-2.53285	-1.98692	27.4761	1.86761
0.0625	-2.13652	-1.98692	7.52918	1.96058
0.03125	-2.02536	-1.98692	1.93444	1.98962
0.015625	-1.9966	-1.98692	0.487101	1.99737
0.0078125	-1.98935	-1.98692	0.121997	1.99934
0.00390625	-1.98753	-1.98692	0.0305133	1.99983
0.00195312	-1.98708	-1.98692	0.0076292	1.99996

Convergence using exact solution with kl = 0.5

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-6.48127	-6.2874	3.08348	nan
0.125	-6.48127	-6.2874	3.08348	1.98369
0.0625	-6.33642	-6.2874	0.779635	1.99583
0.03125	-6.29969	-6.2874	0.195473	1.99895
0.015625	-6.29048	-6.2874	0.0489036	1.99974
0.0078125	-6.28817	-6.2874	0.0122281	1.99993
0.00390625	-6.28759	-6.2874	0.00305717	1.99998
0.00195312	-6.28745	-6.2874	0.000764302	2

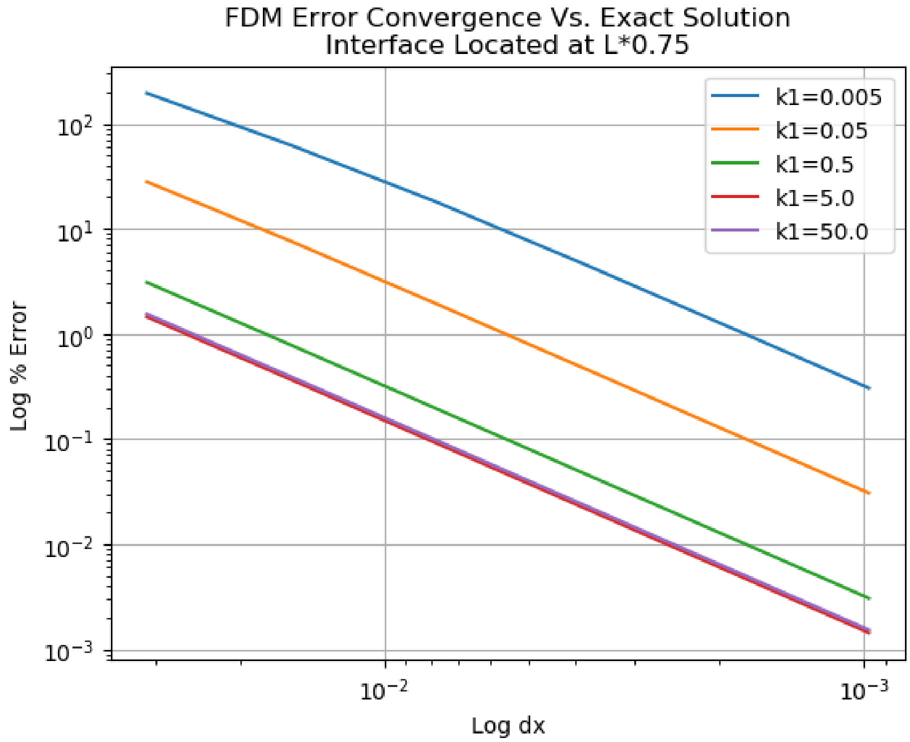
Convergence using exact solution with kl = 5.0

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-15.0367	-14.9029	0.897749	nan
0.125	-15.0367	-14.9029	0.897749	1.98576
0.0625	-14.9367	-14.9029	0.226664	1.99635
0.03125	-14.9114	-14.9029	0.0568095	1.99908
0.015625	-14.905	-14.9029	0.0142114	1.99977
0.0078125	-14.9034	-14.9029	0.00355342	1.99994
0.00390625	-14.903	-14.9029	0.00088839	1.99999
0.00195312	-14.9029	-14.9029	0.000222099	1.99991

Convergence using exact solution with kl = 50.0

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-18.7366	-18.5395	1.06324	nan
0.125	-18.7366	-18.5395	1.06324	1.98511
0.0625	-18.5893	-18.5395	0.268569	1.99619
0.03125	-18.552	-18.5395	0.0673196	1.99904
0.015625	-18.5426	-18.5395	0.0168411	1.99976
0.0078125	-18.5403	-18.5395	0.00421097	1.99994
0.00390625	-18.5397	-18.5395	0.00105279	1.99992
0.00195312	-18.5396	-18.5395	0.000263212	1.99878

5.4.2 Case 2: Interface Located at 3L/4



Convergence using exact solution with k1 = 0.005

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-1.6918	-0.628319	169.259	nan
0.125	-1.6918	-0.628319	169.259	1.49432
0.0625	-1.0058	-0.628319	60.0781	1.74489
0.03125	-0.740943	-0.628319	17.9248	1.91017
0.015625	-0.658284	-0.628319	4.76909	1.97473
0.0078125	-0.635942	-0.628319	1.21334	1.99347
0.00390625	-0.630233	-0.628319	0.304712	1.99835
0.00195312	-0.628798	-0.628319	0.0762649	1.99959

Convergence using exact solution with k1 = 0.05

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-2.53957	-1.99063	27.5761	nan
0.125	-2.53957	-1.99063	27.5761	1.86854
0.0625	-2.14096	-1.99063	7.55176	1.96084
0.03125	-2.02925	-1.99063	1.93989	1.98969
0.015625	-2.00036	-1.99063	0.48845	1.99739
0.0078125	-1.99307	-1.99063	0.122334	1.99934
0.00390625	-1.99124	-1.99063	0.0305973	1.99984
0.00195312	-1.99079	-1.99063	0.0076502	1.99996

Convergence using exact solution with kl = 0.5

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-6.48127	-6.2874	3.08348	nan
0.125	-6.48127	-6.2874	3.08348	1.98369
0.0625	-6.33642	-6.2874	0.779635	1.99583
0.03125	-6.29969	-6.2874	0.195473	1.99895
0.015625	-6.29048	-6.2874	0.0489036	1.99974
0.0078125	-6.28817	-6.2874	0.0122281	1.99993
0.00390625	-6.28759	-6.2874	0.00305717	1.99998
0.00195312	-6.28745	-6.2874	0.000764302	2

Convergence using exact solution with kl = 5.0

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-11.4608	-11.2973	1.44739	nan
0.125	-11.4608	-11.2973	1.44739	1.9836
0.0625	-11.3386	-11.2973	0.365985	1.99581
0.03125	-11.3076	-11.2973	0.0917627	1.99895
0.015625	-11.2999	-11.2973	0.0229574	1.99974
0.0078125	-11.2979	-11.2973	0.00574041	1.99993
0.00390625	-11.2974	-11.2973	0.00143517	1.99999
0.00195312	-11.2973	-11.2973	0.000358796	1.99995

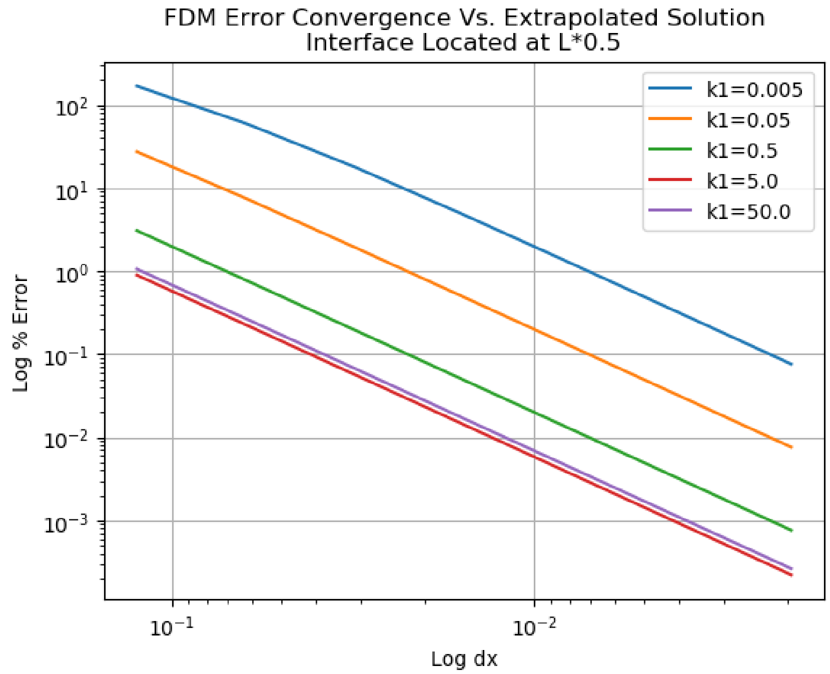
Convergence using exact solution with kl = 50.0

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-12.6439	-12.452	1.5408	nan
0.125	-12.6439	-12.452	1.5408	1.98389
0.0625	-12.5005	-12.452	0.389526	1.99588
0.03125	-12.4642	-12.452	0.0976597	1.99897
0.015625	-12.4551	-12.452	0.0244324	1.99974
0.0078125	-12.4528	-12.452	0.00610921	1.99993
0.00390625	-12.4522	-12.452	0.00152737	1.99995
0.00195312	-12.4521	-12.452	0.000381857	1.99934

5.5 Convergence of FDM Against the Extrapolated Solution

The following graphs and tables represent the convergence of the FDM solution with the extrapolated values of the FDM solution.

5.5.1 Case 1: Interface Located at L/2



Convergence using Richardson extrapolation with k1 = 0.005

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-1.6918	-0.574374	194.547	1.37299
0.015625	-1.0058	-0.620783	62.0214	1.67996
0.0078125	-0.740943	-0.627667	18.0472	1.88746
0.00390625	-0.658284	-0.628273	4.77661	1.96839
0.00195312	-0.635942	-0.628316	1.21381	1.99184
0.000976562	-0.630233	-0.628318	0.304741	1.99794

Convergence using Richardson extrapolation with k1 = 0.05

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-2.53285	-1.98203	27.7911	1.83402
0.015625	-2.13652	-1.98657	7.54852	1.95067
0.0078125	-2.02536	-1.9869	1.93564	1.98703
0.00390625	-1.9966	-1.98692	0.487176	1.99671
0.00195312	-1.98935	-1.98692	0.122002	1.99918
0.000976562	-1.98753	-1.98692	0.0305136	1.99979

Convergence using Richardson extrapolation with kl = 0.5

dx	q_dot(l)	q_dot(l)_extra	% Error	Beta
0.03125	-6.48127	-6.28722	3.08654	1.9796
0.015625	-6.33642	-6.28739	0.779825	1.99479
0.0078125	-6.29969	-6.2874	0.195484	1.99869
0.00390625	-6.29048	-6.2874	0.0489044	1.99967
0.00195312	-6.28817	-6.2874	0.0122282	1.99992
0.000976562	-6.28759	-6.2874	0.00305717	1.99998

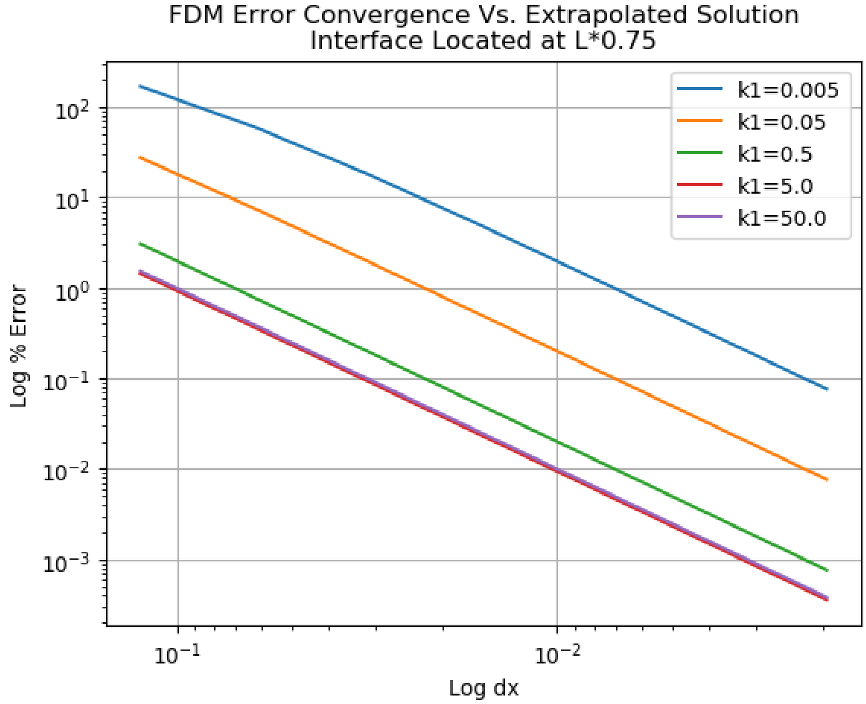
Convergence using Richardson extrapolation with kl = 5.0

dx	q_dot(l)	q_dot(l)_extra	% Error	Beta
0.03125	-15.0367	-14.9028	0.898506	1.98219
0.015625	-14.9367	-14.9029	0.226712	1.99544
0.0078125	-14.9114	-14.9029	0.0568125	1.99885
0.00390625	-14.905	-14.9029	0.0142116	1.99971
0.00195312	-14.9034	-14.9029	0.00355343	1.99992
0.000976562	-14.903	-14.9029	0.000888387	1.99998

Convergence using Richardson extrapolation with kl = 50.0

dx	q_dot(l)	q_dot(l)_extra	% Error	Beta
0.03125	-18.7366	-18.5393	1.06418	1.98138
0.015625	-18.5893	-18.5395	0.268628	1.99524
0.0078125	-18.552	-18.5395	0.0673233	1.9988
0.00390625	-18.5426	-18.5395	0.0168413	1.9997
0.00195312	-18.5403	-18.5395	0.00421096	1.99994
0.000976562	-18.5397	-18.5395	0.00105269	2.00032

5.5.2 Case 2: Interface Located at 3L/4



Convergence using Richardson extrapolation with k1 = 0.005

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-1.6918	-0.574375	194.547	1.373
0.015625	-1.0058	-0.620783	62.0214	1.67996
0.0078125	-0.740943	-0.627667	18.0472	1.88746
0.00390625	-0.658284	-0.628273	4.77661	1.96839
0.00195312	-0.635942	-0.628316	1.21381	1.99184
0.000976562	-0.630233	-0.628318	0.304741	1.99794

Convergence using Richardson extrapolation with k1 = 0.05

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-2.53957	-1.98575	27.8898	1.8352
0.015625	-2.14096	-1.99028	7.57102	1.951
0.0078125	-2.02925	-1.99061	1.94108	1.98711
0.00390625	-2.00036	-1.99063	0.488524	1.99673
0.00195312	-1.99307	-1.99063	0.122338	1.99918
0.000976562	-1.99124	-1.99063	0.0305976	1.9998

Convergence using Richardson extrapolation with kl = 0.5

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-6.48127	-6.28722	3.08654	1.9796
0.015625	-6.33642	-6.28739	0.779825	1.99479
0.0078125	-6.29969	-6.2874	0.195484	1.99869
0.00390625	-6.29048	-6.2874	0.0489044	1.99967
0.00195312	-6.28817	-6.2874	0.0122282	1.99992
0.000976562	-6.28759	-6.2874	0.00305717	1.99998

Convergence using Richardson extrapolation with kl = 5.0

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-11.4608	-11.2971	1.44881	1.97949
0.015625	-11.3386	-11.2972	0.366075	1.99476
0.0078125	-11.3076	-11.2973	0.0917683	1.99868
0.00390625	-11.2999	-11.2973	0.0229578	1.99967
0.00195312	-11.2979	-11.2973	0.00574044	1.99992
0.000976562	-11.2974	-11.2973	0.00143516	2.00001

Convergence using Richardson extrapolation with kl = 50.0

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-12.6439	-12.4518	1.54229	1.97986
0.015625	-12.5005	-12.452	0.389619	1.99485
0.0078125	-12.4642	-12.452	0.0976655	1.99871
0.00390625	-12.4551	-12.452	0.0244328	1.99968
0.00195312	-12.4528	-12.452	0.00610921	1.99993
0.000976562	-12.4522	-12.452	0.0015273	2.00016

6 Discussion

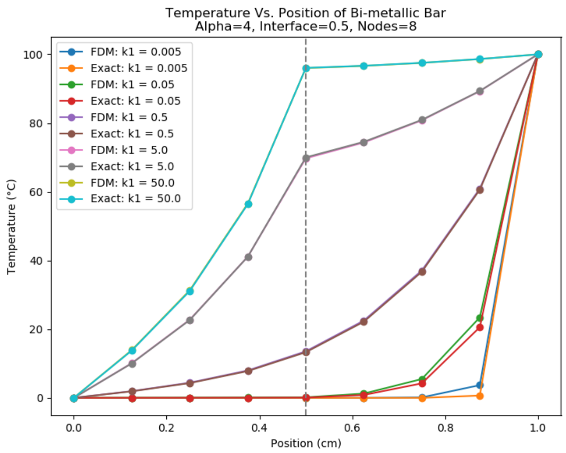
In this assignment, the heat transfer within a bi-material bar with boundary conditions of $T(0) = 0$℃ and $T(L) = 100$℃ was analyzed using two methods. The first method consisted of deriving the analytical solution with the utilization of harmonic functions. The benefit of this method is that this exact solution will derive the temperature at any place within the bar. However, obtaining an exact solution is not always a possibility, thus, a solution derived utilizing the finite difference method can be used instead. This second method can derive a solution for the temperature of the bar at several nodes within the bar. However, its accuracy is dependent on the number of nodes used. These two methods were utilized in two different cases. In the first case, the interface between the two materials was located at the midpoint of the bar. In the second case, the interface between the two materials was located at the three-fourths point of the bar.

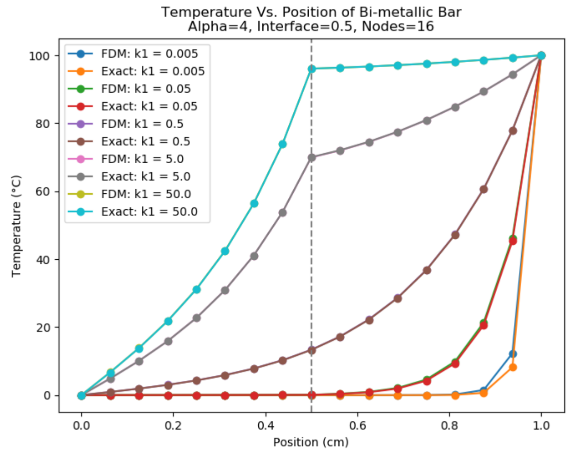
As evident from the various graphs provided in the results section of this report, the heat conduction as a relatively smooth transition for values of $k_1 < k_0$. When the value of $k_1 = k_0$, the heat transfer within the bar resembles the solution of a bar made of a singular material. This result is expected. For values of $k_1 > k_0$, a sharper transition in the heat transfer solution can be seen. This result makes intuitive sense as materials with a lower heat conductivity value will resist changes in temperature and will produce a greater heat gradient.

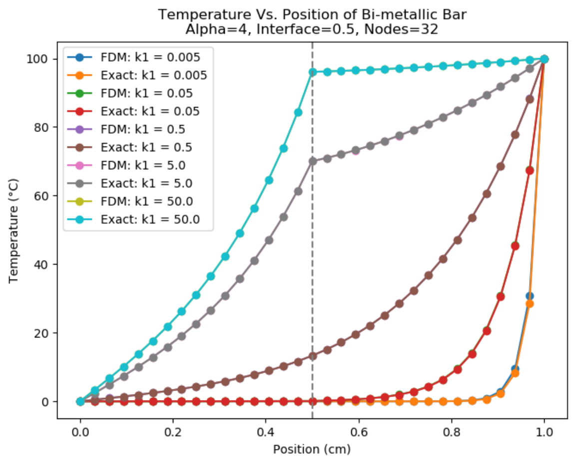
An investigation into the convergence plots of the FDM solution against the exact solution shows that the slope of the plots is 2, which approximately corresponds to the β values found in the convergence tables. This also matched the second order finite difference method solution. An similar investigation into the convergence plots of the FDM solution against the Richardson's extrapolated solution shows the same results. This shows that the Richardson's Extrapolation is an extremely useful tool in quickly deriving values close to the exact solution when an exact solution is incapable of being derived.

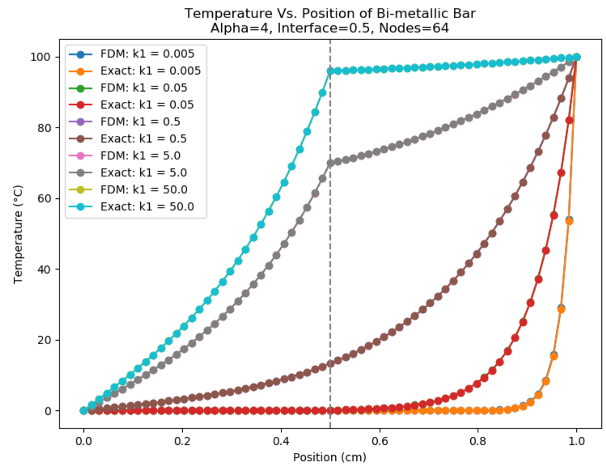
7 Appendix A – FDM/Exact Solution Comparison Graphs

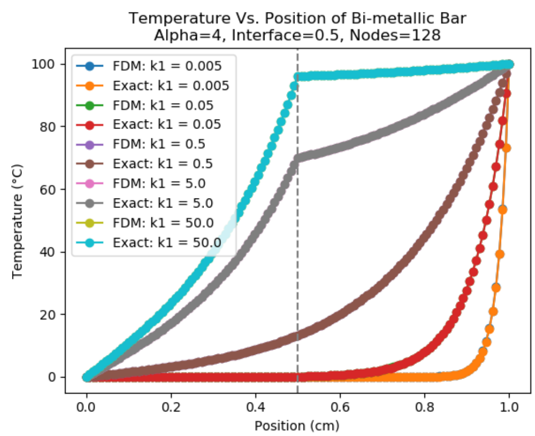
The following graphs show the comparison of the FDM and exact solutions at various thermal conductivity values. Each graph was obtained using a different number of nodes. It is notable that the exact and FDM solutions tend to align over top of each other and thus may falsely give the appearance that there is missing data.

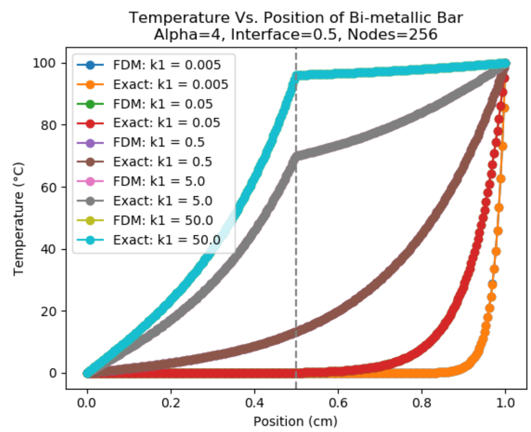


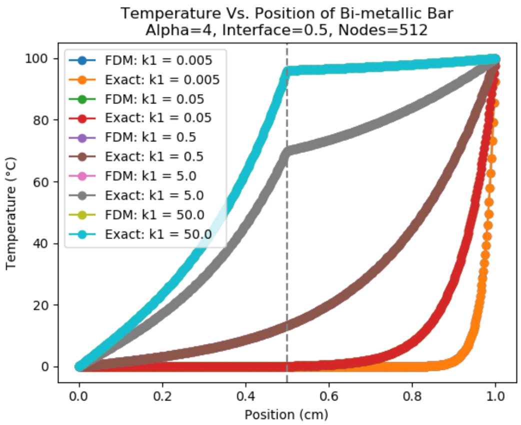


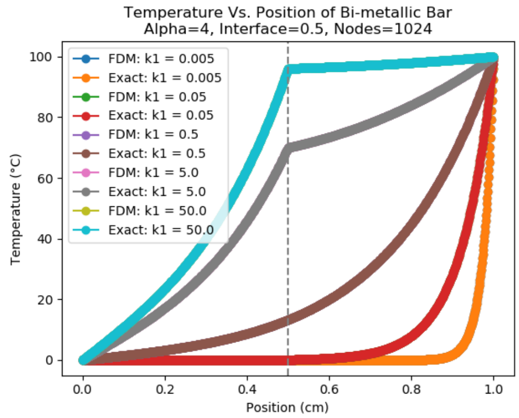


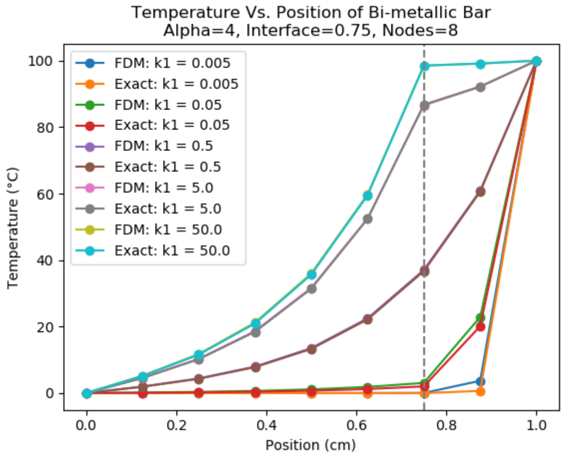


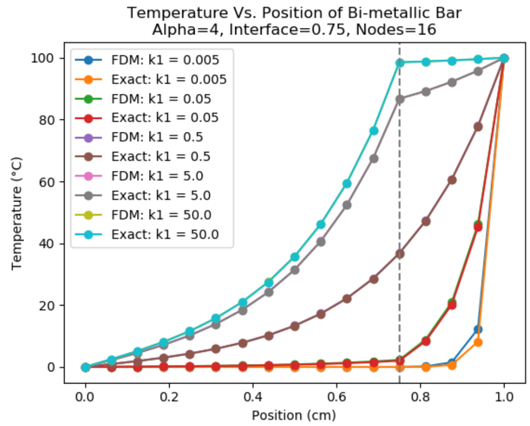


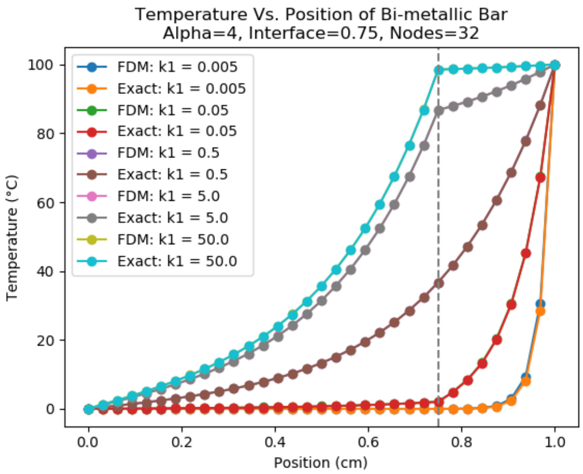


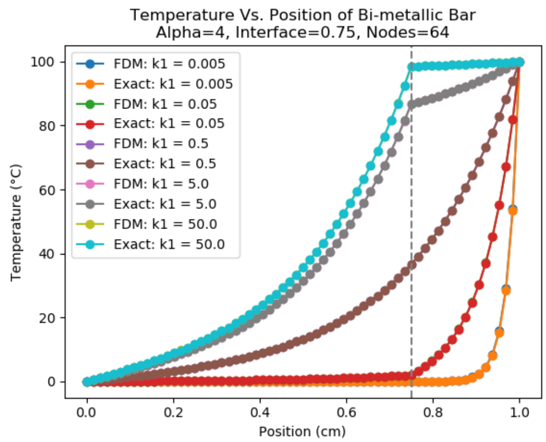

Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.5, Nodes=512
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)

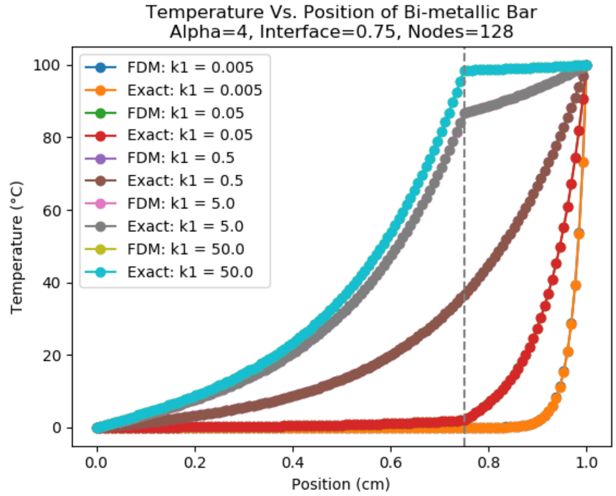

Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.5, Nodes=1024
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)

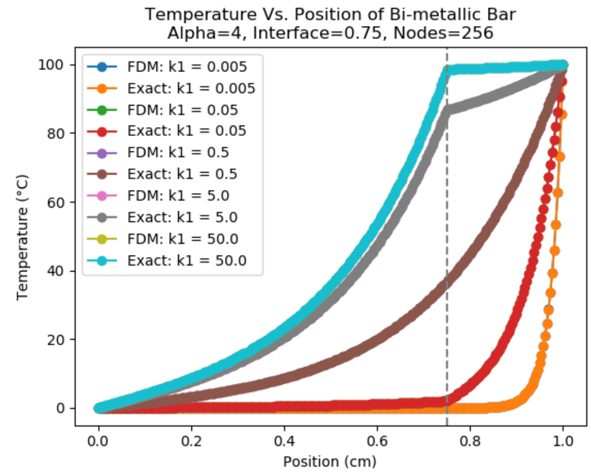

Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.75, Nodes=8
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)

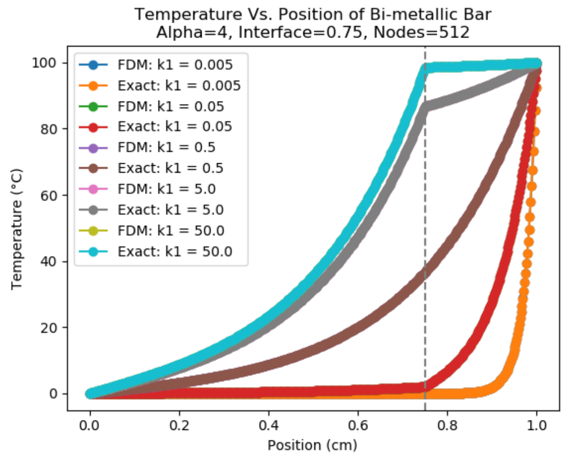

Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.75, Nodes=16
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)

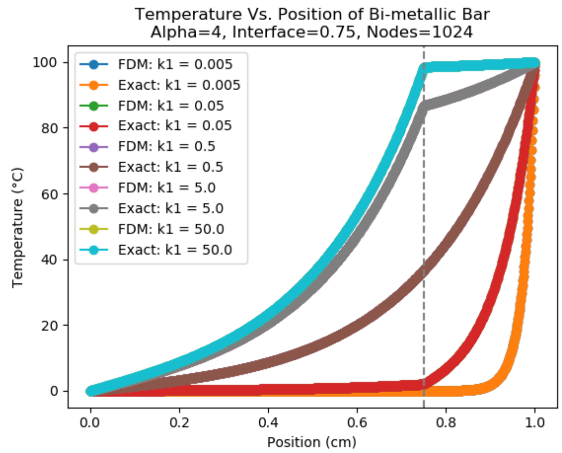

Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.75, Nodes=32
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)


Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.75, Nodes=64
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)


Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.75, Nodes=128
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)


Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.75, Nodes=256
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)


Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.75, Nodes=512
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)


Temperature Vs. Position of Bi-metallic Bar
Alpha=4, Interface=0.75, Nodes=1024
FDM: k1 = 0.005
Exact: k1 = 0.005
FDM: k1 = 0.05
Exact: k1 = 0.05
FDM: k1 = 0.5
Exact: k1 = 0.5
FDM: k1 = 5.0
Exact: k1 = 5.0
FDM: k1 = 50.0
Exact: k1 = 50.0
Temperature (°C)
Position (cm)

8 Appendix B – Temperature Tables for Various Number of Nodes

Note that the tables for nodes greater than 32 are not provided due to the abundance in data provided by these tables. These unincluded tables can be generated using the attached python script.

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.005 Nodes = 8

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.125	5.2077e-09	0.125	1.10339e-05
0.25	1.17447e-08	0.25	2.48263e-05
0.375	2.12795e-08	0.375	4.48253e-05
0.5	3.6246e-08	0.5	7.60306e-05
0.625	3.05891e-05	0.625	0.00509732
0.75	0.00453999	0.75	0.137552
0.875	0.673795	0.875	3.7088
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.5 Nodes = 8

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.125	1.90948	0.125	1.96675
0.25	4.30636	0.25	4.42519
0.375	7.80244	0.375	7.98993
0.5	13.2901	0.5	13.5521
0.625	22.1701	0.625	22.5024
0.75	36.7091	0.75	37.0783
0.875	60.6181	0.875	60.9237
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.05 Nodes = 8

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.125	0.0120289	0.125	0.0239775
0.25	0.0271282	0.25	0.0539494
0.375	0.049152	0.375	0.0974086
0.5	0.083722	0.5	0.16522
0.625	0.851247	0.625	1.25638
0.75	4.22889	0.75	5.48851
0.875	20.5733	0.875	23.4419
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 5.0 Nodes = 8

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.125	10.0591	0.125	10.1198
0.25	22.6857	0.25	22.7695
0.375	41.103	0.375	41.1115
0.5	70.0119	0.5	69.7315
0.625	74.5352	0.625	74.3368
0.75	80.9258	0.75	80.8005
0.875	89.3438	0.875	89.2842
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 50.0 Nodes = 8

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.125	13.8024	0.125	13.9327
0.25	31.128	0.25	31.3485
0.375	56.399	0.375	56.6015
0.5	96.0659	0.5	96.0049
0.625	96.6845	0.625	96.6389
0.75	97.5448	0.75	97.5146
0.875	98.6491	0.875	98.634
1	100	1	100

Temperature Vs. Position Tables for Exact Analysis
Nodes = 8

Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50	T, k1 = 500
0	0	0	0	0	0
0.125	5.2077e-09	0.0120289	1.90948	10.0591	13.8024
0.25	1.17447e-08	0.0271282	4.30636	22.6857	31.128
0.375	2.12795e-08	0.049152	7.80244	41.103	56.399
0.5	3.6246e-08	0.083722	13.2901	70.0119	96.0659
0.625	3.05891e-05	0.851247	22.1701	74.5352	96.6845
0.75	0.00453999	4.22889	36.7091	80.9258	97.5448
0.875	0.673795	20.5733	60.6181	89.3438	98.6491
1	100	100	100	100	100

Temperature Vs. Position Tables for FDM Analysis
Nodes = 8

Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50	T, k1 = 500
0	0	0	0	0	0
0.125	1.10339e-05	0.0239775	1.96675	10.1198	13.9327
0.25	2.48263e-05	0.0539494	4.42519	22.7695	31.3485
0.375	4.48253e-05	0.0974086	7.98993	41.1115	56.6015
0.5	7.60306e-05	0.16522	13.5521	69.7315	96.0049
0.625	0.00509732	1.25638	22.5024	74.3368	96.6389
0.75	0.137552	5.48851	37.0783	80.8005	97.5146
0.875	3.7088	23.4419	60.9237	89.2842	98.634
1	100	100	100	100	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.005 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	2.52455e-09	0.0625	9.74424e-08
0.125	5.2077e-09	0.125	2.00975e-07
0.1875	8.21803e-09	0.1875	3.17069e-07
0.25	1.17447e-08	0.25	4.52979e-07
0.3125	1.60092e-08	0.3125	6.172e-07
0.375	2.12795e-08	0.375	8.19997e-07
0.4375	2.78867e-08	0.4375	1.07404e-06
0.5	3.6246e-08	0.5	1.39522e-06
0.5625	2.49706e-06	0.5625	4.22327e-05
0.625	3.05891e-05	0.625	0.000347025
0.6875	0.000372665	0.6875	0.00282072
0.75	0.00453999	0.75	0.0229239
0.8125	0.0553084	0.8125	0.186302
0.875	0.673795	0.875	1.51407
0.9375	8.2085	0.9375	12.3047
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.05 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	0.00583127	0.0625	0.00716406
0.125	0.0120289	0.125	0.0147759
0.1875	0.0189822	0.1875	0.0233112
0.25	0.0271282	0.25	0.0333034
0.3125	0.0369785	0.3125	0.0453771
0.375	0.049152	0.375	0.0602869
0.4375	0.0644135	0.4375	0.0789646
0.5	0.083722	0.5	0.102578
0.5625	0.351725	0.5625	0.402819
0.625	0.851247	0.625	0.954821
0.6875	1.91109	0.6875	2.10359
0.75	4.22889	0.75	4.5671
0.8125	9.3303	0.8125	9.88504
0.875	20.5733	0.875	21.3811
0.9375	45.3583	0.9375	46.2404
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.5 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	0.925662	0.0625	0.932824
0.125	1.90948	0.125	1.92395
0.1875	3.01326	0.1875	3.03532
0.25	4.30636	0.25	4.3364
0.3125	5.87	0.3125	5.90851
0.375	7.80244	0.375	7.8499
0.4375	10.2251	0.4375	10.2819
0.5	13.2901	0.5	13.3565
0.5625	17.1901	0.5625	17.2659
0.625	22.1701	0.625	22.2545
0.6875	28.543	0.6875	28.6339
0.75	36.7091	0.75	36.803
0.8125	47.1815	0.8125	47.2722
0.875	60.6181	0.875	60.6959
0.9375	77.8631	0.9375	77.9132
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 5.0 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	4.87636	0.0625	4.88467
0.125	10.0591	0.125	10.0746
0.1875	15.8738	0.1875	15.8942
0.25	22.6857	0.25	22.7073
0.3125	30.923	0.3125	30.9395
0.375	41.103	0.375	41.1054
0.4375	53.8654	0.4375	53.8404
0.5	70.0119	0.5	69.9405
0.5625	72.0483	0.5625	71.9876
0.625	74.5352	0.625	74.4846
0.6875	77.4882	0.6875	77.4472
0.75	80.9258	0.75	80.8939
0.8125	84.8694	0.8125	84.8461
0.875	89.3438	0.875	89.3286
0.9375	94.3768	0.9375	94.3694
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis

k1 = 50.0 Nodes = 16

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.0625	6.69103	0.0625	6.7082
0.125	13.8024	0.125	13.8357
0.1875	21.781	0.1875	21.8279
0.25	31.128	0.25	31.1843
0.3125	42.4306	0.3125	42.4897
0.375	56.399	0.375	56.4508
0.4375	73.9107	0.4375	73.94
0.5	96.0659	0.5	96.0505
0.5625	96.3451	0.5625	96.3316
0.625	96.6845	0.625	96.673
0.6875	97.0843	0.6875	97.0747
0.75	97.5448	0.75	97.5372
0.8125	98.0663	0.8125	98.0606
0.875	98.6491	0.875	98.6453
0.9375	99.2935	0.9375	99.2916
1	100	1	100

Temperature Vs. Position Tables for Exact Analysis

Nodes = 16

Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50	T, k1 = 500
0	0	0	0	0	0
0.0625	2.52455e-09	0.00583127	0.925662	4.87636	6.69103
0.125	5.2077e-09	0.0120289	1.90948	10.0591	13.8024
0.1875	8.21803e-09	0.0189822	3.01326	15.8738	21.781
0.25	1.17447e-08	0.0271282	4.30636	22.6857	31.128
0.3125	1.60092e-08	0.0369785	5.87	30.923	42.4306
0.375	2.12795e-08	0.049152	7.80244	41.103	56.399
0.4375	2.78867e-08	0.0644135	10.2251	53.8654	73.9107
0.5	3.6246e-08	0.083722	13.2901	70.0119	96.0659
0.5625	2.49706e-06	0.351725	17.1901	72.0483	96.3451
0.625	3.05891e-05	0.851247	22.1701	74.5352	96.6845
0.6875	0.000372665	1.91109	28.543	77.4882	97.0843
0.75	0.00453999	4.22889	36.7091	80.9258	97.5448
0.8125	0.0553084	9.3303	47.1815	84.8694	98.0663
0.875	0.673795	20.5733	60.6181	89.3438	98.6491
0.9375	8.2085	45.3583	77.8631	94.3768	99.2935
1	100	100	100	100	100

Temperature Vs. Position Tables for FDM Analysis

Nodes = 16

Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50	T, k1 = 500
0	0	0	0	0	0
0.0625	9.74424e-08	0.00716406	0.932824	4.88467	6.7082
0.125	2.00975e-07	0.0147759	1.92395	10.0746	13.8357
0.1875	3.17069e-07	0.0233112	3.03532	15.8942	21.8279
0.25	4.52979e-07	0.0333034	4.3364	22.7073	31.1843
0.3125	6.172e-07	0.0453771	5.90851	30.9395	42.4897
0.375	8.19997e-07	0.0602869	7.8499	41.1054	56.4508
0.4375	1.07404e-06	0.0789646	10.2819	53.8404	73.94
0.5	1.39522e-06	0.102578	13.3565	69.9405	96.0505
0.5625	4.22327e-05	0.402819	17.2659	71.9876	96.3316
0.625	0.000347025	0.954821	22.2545	74.4846	96.673
0.6875	0.00282072	2.10359	28.6339	77.4472	97.0747
0.75	0.0229239	4.5671	36.803	80.8939	97.5372
0.8125	0.186302	9.88504	47.2722	84.8461	98.0606
0.875	1.51407	21.3811	60.6959	89.3286	98.6453
0.9375	12.3047	46.2404	77.9132	94.3694	99.2916
1	100	100	100	100	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.05 Nodes = 32

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.03125	0.002893	0.03125	0.00305429
0.0625	0.00583127	0.0625	0.00615631
0.09375	0.00886077	0.09375	0.00935452
0.125	0.0120289	0.125	0.0126989
0.15625	0.0153852	0.15625	0.0162417
0.1875	0.0189822	0.1875	0.0200383
0.21875	0.0228763	0.21875	0.0241479
0.25	0.0271282	0.25	0.0286349
0.28125	0.0318045	0.28125	0.0335693
0.3125	0.0369785	0.3125	0.0390282
0.34375	0.042731	0.34375	0.045097
0.375	0.049152	0.375	0.0518703
0.40625	0.056342	0.40625	0.0594542
0.4375	0.0644135	0.4375	0.067967
0.46875	0.0734928	0.46875	0.0775418
0.5	0.083722	0.5	0.0883282
0.53125	0.201755	0.53125	0.209993
0.5625	0.351725	0.5625	0.36447
0.59375	0.557371	0.59375	0.575895
0.625	0.851247	0.625	0.877304
0.65625	1.27987	0.65625	1.31579
0.6875	1.91109	0.6875	1.95987
0.71875	2.84483	0.71875	2.91018
0.75	4.22889	0.75	4.31521
0.78125	6.28236	0.78125	6.39448
0.8125	9.3303	0.8125	9.4729
0.84375	13.8552	0.84375	14.0315
0.875	20.5733	0.875	20.7824
0.90625	30.548	0.90625	30.7807
0.9375	45.3583	0.9375	45.5884
0.96875	67.3487	0.96875	67.5192
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 0.5 Nodes = 32

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.03125	0.459238	0.03125	0.460134
0.0625	0.925662	0.0625	0.927457
0.09375	1.40657	0.09375	1.40927
0.125	1.90948	0.125	1.91311
0.15625	2.44227	0.15625	2.44683
0.1875	3.01326	0.1875	3.01879
0.21875	3.6314	0.21875	3.63792
0.25	4.30636	0.25	4.31389
0.28125	5.04869	0.28125	5.05726
0.3125	5.87	0.3125	5.87966
0.34375	6.78316	0.34375	6.79392
0.375	7.80244	0.375	7.81434
0.40625	8.94379	0.40625	8.95686
0.4375	10.2251	0.4375	10.2393
0.46875	11.6663	0.46875	11.6818
0.5	13.2901	0.5	13.3068
0.53125	15.1218	0.53125	15.1397
0.5625	17.1901	0.5625	17.2091
0.59375	19.5274	0.59375	19.5475
0.625	22.1701	0.625	22.1913
0.65625	25.1597	0.65625	25.1818
0.6875	28.543	0.6875	28.5658
0.71875	32.3728	0.71875	32.3961
0.75	36.7091	0.75	36.7326
0.78125	41.6197	0.78125	41.6431
0.8125	47.1815	0.8125	47.2043
0.84375	53.4814	0.84375	53.503
0.875	60.6181	0.875	60.6376
0.90625	68.7032	0.90625	68.7198
0.9375	77.8631	0.9375	77.8757
0.96875	88.2413	0.96875	88.2484
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 5.0 Nodes = 32

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.03125	2.41925	0.03125	2.42031
0.0625	4.87636	0.0625	4.87845
0.09375	7.40975	0.09375	7.4128
0.125	10.0591	0.125	10.063
0.15625	12.8658	0.15625	12.8704
0.1875	15.8738	0.1875	15.8789
0.21875	19.1301	0.21875	19.1355
0.25	22.6857	0.25	22.6912
0.28125	26.5963	0.28125	26.6013
0.3125	30.923	0.3125	30.9271
0.34375	35.7335	0.34375	35.7362
0.375	41.103	0.375	41.1036
0.40625	47.1156	0.40625	47.1133
0.4375	53.8654	0.4375	53.8591
0.46875	61.4578	0.46875	61.4465
0.5	70.0119	0.5	69.9939
0.53125	70.9746	0.53125	70.958
0.5625	72.0483	0.5625	72.033
0.59375	73.2345	0.59375	73.2206
0.625	74.5352	0.625	74.5225
0.65625	75.9524	0.65625	75.9409
0.6875	77.4882	0.6875	77.4779
0.71875	79.1452	0.71875	79.136
0.75	80.9258	0.75	80.9178
0.78125	82.8329	0.78125	82.826
0.8125	84.8694	0.8125	84.8636
0.84375	87.0386	0.84375	87.0338
0.875	89.3438	0.875	89.34
0.90625	91.7886	0.90625	91.7857
0.9375	94.3768	0.9375	94.3749
0.96875	97.1125	0.96875	97.1116
1	100	1	100

Temperature Vs. Position Tables for Exact and FDM Analysis
k1 = 50.0 Nodes = 32

Position (cm)	T Exact	Position (cm)	T FDM
0	0	0	0
0.03125	3.31955	0.03125	3.32172
0.0625	6.69103	0.0625	6.69535
0.09375	10.1672	0.09375	10.1736
0.125	13.8024	0.125	13.8108
0.15625	17.6536	0.15625	17.6638
0.1875	21.781	0.1875	21.7928
0.21875	26.2491	0.21875	26.2623
0.25	31.128	0.25	31.1421
0.28125	36.4938	0.28125	36.5086
0.3125	42.4306	0.3125	42.4455
0.34375	49.0312	0.34375	49.0456
0.375	56.399	0.375	56.412
0.40625	64.6491	0.40625	64.6599
0.4375	73.9107	0.4375	73.9181
0.46875	84.3286	0.46875	84.3312
0.5	96.0659	0.5	96.0621
0.53125	96.198	0.53125	96.1944
0.5625	96.3451	0.5625	96.3417
0.59375	96.5073	0.59375	96.5041
0.625	96.6845	0.625	96.6816
0.65625	96.8769	0.65625	96.8742
0.6875	97.0843	0.6875	97.0819
0.71875	97.307	0.71875	97.3048
0.75	97.5448	0.75	97.5429
0.78125	97.7979	0.78125	97.7963
0.8125	98.0663	0.8125	98.0649
0.84375	98.35	0.84375	98.3488
0.875	98.6491	0.875	98.6481
0.90625	98.9636	0.90625	98.9628
0.9375	99.2935	0.9375	99.293
0.96875	99.639	0.96875	99.6387
1	100	1	100

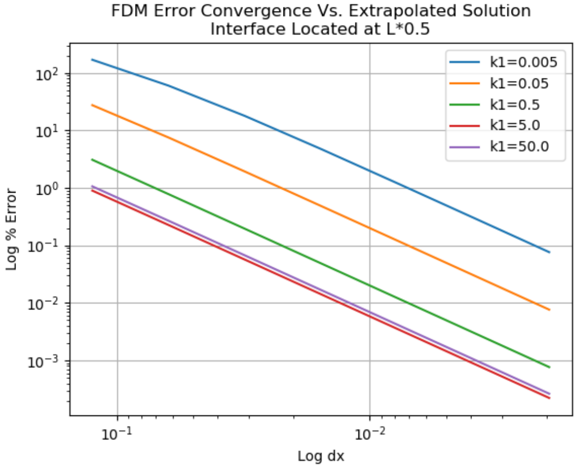
Temperature Vs. Position Tables for Exact Analysis
Nodes = 32

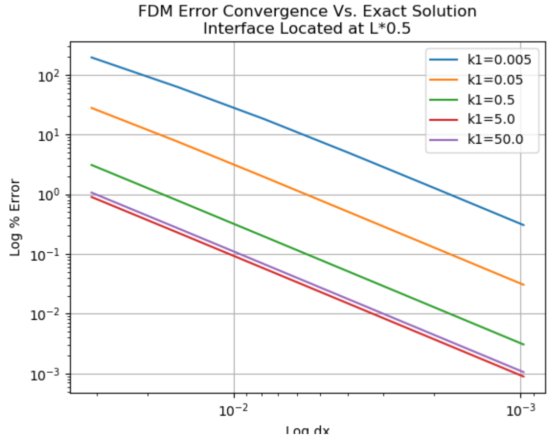
Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50
0	0	0	0	0
0.03125	0.002893	0.459238	2.41925	3.31955
0.0625	0.00583127	0.925662	4.87636	6.69103
0.09375	0.00886077	1.40657	7.40975	10.1672
0.125	0.0120289	1.90948	10.0591	13.8024
0.15625	0.0153852	2.44227	12.8658	17.6536
0.1875	0.0189822	3.01326	15.8738	21.781
0.21875	0.0228763	3.6314	19.1301	26.2491
0.25	0.0271282	4.30636	22.6857	31.128
0.28125	0.0318045	5.04869	26.5963	36.4938
0.3125	0.0369785	5.87	30.923	42.4306
0.34375	0.042731	6.78316	35.7335	49.0312
0.375	0.049152	7.80244	41.103	56.399
0.40625	0.056342	8.94379	47.1156	64.6491
0.4375	0.0644135	10.2251	53.8654	73.9107
0.46875	0.0734928	11.6663	61.4578	84.3286
0.5	0.083722	13.2901	70.0119	96.0659
0.53125	0.201755	15.1218	70.9746	96.198
0.5625	0.351725	17.1901	72.0483	96.3451
0.59375	0.557371	19.5274	73.2345	96.5073
0.625	0.851247	22.1701	74.5352	96.6845
0.65625	1.27987	25.1597	75.9524	96.8769
0.6875	1.91109	28.543	77.4882	97.0843
0.71875	2.84483	32.3728	79.1452	97.307
0.75	4.22889	36.7091	80.9258	97.5448
0.78125	6.28236	41.6197	82.8329	97.7979
0.8125	9.3303	47.1815	84.8694	98.0663
0.84375	13.8552	53.4814	87.0386	98.35
0.875	20.5733	60.6181	89.3438	98.6491
0.90625	30.548	68.7032	91.7886	98.9636
0.9375	45.3583	77.8631	94.3768	99.2935
0.96875	67.3487	88.2413	97.1125	99.639
1	100	100	100	100

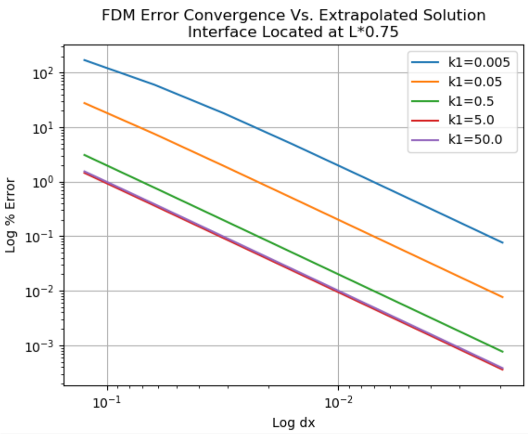
Temperature Vs. Position Tables for FDM Analysis
Nodes = 32

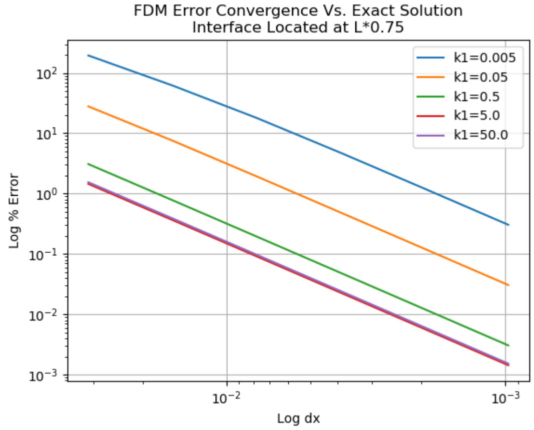
Position (cm)	T, k1 = 0.05	T, k1 = 0.5	T, k1 = 5	T, k1 = 50
0	0	0	0	0
0.03125	0.00305429	0.460134	2.42031	3.32172
0.0625	0.00615631	0.927457	4.87845	6.69535
0.09375	0.00935452	1.40927	7.4128	10.1736
0.125	0.0126989	1.91311	10.063	13.8108
0.15625	0.0162417	2.44683	12.8704	17.6638
0.1875	0.0200383	3.01879	15.8789	21.7928
0.21875	0.0241479	3.63792	19.1355	26.2623
0.25	0.0286349	4.31389	22.6912	31.1421
0.28125	0.0335693	5.05726	26.6013	36.5086
0.3125	0.0390282	5.87966	30.9271	42.4455
0.34375	0.045097	6.79392	35.7362	49.0456
0.375	0.0518703	7.81434	41.1036	56.412
0.40625	0.0594542	8.95686	47.1133	64.6599
0.4375	0.067967	10.2393	53.8591	73.9181
0.46875	0.0775418	11.6818	61.4465	84.3312
0.5	0.0883282	13.3068	69.9939	96.0621
0.53125	0.209993	15.1397	70.958	96.1944
0.5625	0.36447	17.2091	72.033	96.3417
0.59375	0.575895	19.5475	73.2206	96.5041
0.625	0.877304	22.1913	74.5225	96.6816
0.65625	1.31579	25.1818	75.9409	96.8742
0.6875	1.95987	28.5658	77.4779	97.0819
0.71875	2.91018	32.3961	79.136	97.3048
0.75	4.31521	36.7326	80.9178	97.5429
0.78125	6.39448	41.6431	82.826	97.7963
0.8125	9.4729	47.2043	84.8636	98.0649
0.84375	14.0315	53.503	87.0338	98.3488
0.875	20.7824	60.6376	89.34	98.6481
0.90625	30.7807	68.7198	91.7857	98.9628
0.9375	45.5884	77.8757	94.3749	99.293
0.96875	67.5192	88.2484	97.1116	99.6387
1	100	100	100	100

9 Appendix C – Convergence Graphs and Tables









Convergence using Richardson extrapolation with k1 = 0.005

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-1.6918	-0.574374	194.547	1.37299
0.015625	-1.0058	-0.620783	62.0214	1.67996
0.0078125	-0.740943	-0.627667	18.0472	1.88746
0.00390625	-0.658284	-0.628273	4.77661	1.96839
0.00195312	-0.635942	-0.628316	1.21381	1.99184
0.000976562	-0.630233	-0.628318	0.304741	1.99794

Convergence using exact solution with k1 = 0.005

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-1.6918	-0.628319	169.258	nan
0.125	-1.6918	-0.628319	169.258	1.49431
0.0625	-1.0058	-0.628319	60.0781	1.74489
0.03125	-0.740943	-0.628319	17.9248	1.91017
0.015625	-0.658284	-0.628319	4.76909	1.97473
0.0078125	-0.635942	-0.628319	1.21334	1.99347
0.00390625	-0.630233	-0.628319	0.304712	1.99835
0.00195312	-0.628798	-0.628319	0.0762649	1.99959

Convergence using Richardson extrapolation with k1 = 0.05

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-2.53285	-1.98203	27.7911	1.83402
0.015625	-2.13652	-1.98657	7.54852	1.95067
0.0078125	-2.02536	-1.9869	1.93564	1.98703
0.00390625	-1.9966	-1.98692	0.487176	1.99671
0.00195312	-1.98935	-1.98692	0.122002	1.99918
0.000976562	-1.98753	-1.98692	0.0305136	1.99979

Convergence using exact solution with k1 = 0.05

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-2.53285	-1.98692	27.4761	nan
0.125	-2.53285	-1.98692	27.4761	1.86761
0.0625	-2.13652	-1.98692	7.52918	1.96058
0.03125	-2.02536	-1.98692	1.93444	1.98962
0.015625	-1.9966	-1.98692	0.487101	1.99737
0.0078125	-1.98935	-1.98692	0.121997	1.99934
0.00390625	-1.98753	-1.98692	0.0305133	1.99983
0.00195312	-1.98708	-1.98692	0.0076292	1.99996

Convergence using Richardson extrapolation with k1 = 0.5

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-6.48127	-6.28722	3.08654	1.9796
0.015625	-6.33642	-6.28739	0.779825	1.99479
0.0078125	-6.29969	-6.2874	0.195484	1.99869
0.00390625	-6.29048	-6.2874	0.0489044	1.99967
0.00195312	-6.28817	-6.2874	0.0122282	1.99992
0.000976562	-6.28759	-6.2874	0.00305717	1.99998

Convergence using exact solution with k1 = 0.5

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-6.48127	-6.2874	3.08348	nan
0.125	-6.48127	-6.2874	3.08348	1.98369
0.0625	-6.33642	-6.2874	0.779635	1.99583
0.03125	-6.29969	-6.2874	0.195473	1.99895
0.015625	-6.29048	-6.2874	0.0489036	1.99974
0.0078125	-6.28817	-6.2874	0.0122281	1.99993
0.00390625	-6.28759	-6.2874	0.00305717	1.99998
0.00195312	-6.28745	-6.2874	0.000764302	2

Convergence using Richardson extrapolation with k1 = 5.0

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-15.0367	-14.9028	0.898506	1.98219
0.015625	-14.9367	-14.9029	0.226712	1.99544
0.0078125	-14.9114	-14.9029	0.0568125	1.99885
0.00390625	-14.905	-14.9029	0.0142116	1.99971
0.00195312	-14.9034	-14.9029	0.00355343	1.99992
0.000976562	-14.903	-14.9029	0.000888387	1.99998

Convergence using exact solution with k1 = 5.0

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-15.0367	-14.9029	0.897749	nan
0.125	-15.0367	-14.9029	0.897749	1.98576
0.0625	-14.9367	-14.9029	0.226664	1.99635
0.03125	-14.9114	-14.9029	0.0568095	1.99908
0.015625	-14.905	-14.9029	0.0142114	1.99977
0.0078125	-14.9034	-14.9029	0.00355342	1.99994
0.00390625	-14.903	-14.9029	0.00088839	1.99999
0.00195312	-14.9029	-14.9029	0.000222099	1.99991

Convergence using Richardson extrapolation with k1 = 50.0

dx	q_dot(1)	q_dot(1)_extra	% Error	Beta
0.03125	-18.7366	-18.5393	1.06418	1.98138
0.015625	-18.5893	-18.5395	0.268628	1.99524
0.0078125	-18.552	-18.5395	0.0673233	1.9988
0.00390625	-18.5426	-18.5395	0.0168413	1.9997
0.00195312	-18.5403	-18.5395	0.00421096	1.99994
0.000976562	-18.5397	-18.5395	0.00105269	2.00032

Convergence using exact solution with k1 = 50.0

dx	q_dot(1)	q_dot(1)_exact	% Error	Beta
0.125	-18.7366	-18.5395	1.06324	nan
0.125	-18.7366	-18.5395	1.06324	1.98511
0.0625	-18.5893	-18.5395	0.268569	1.99619
0.03125	-18.552	-18.5395	0.0673196	1.99904
0.015625	-18.5426	-18.5395	0.0168411	1.99976
0.0078125	-18.5403	-18.5395	0.00421097	1.99994
0.00390625	-18.5397	-18.5395	0.00105279	1.99992
0.00195312	-18.5396	-18.5395	0.000263212	1.99878

10 Appendix D – Code

```
from numpy import *
import matplotlib.pyplot as plt
import tabulate as tbl

"""
AERO 430 Homework Assignment 2

Andrew Hollister
UIN: 127008398
Date: 09/14/21

Assignment Description:

Repeat assignment 1 for a bi-material bar as in the report of Caleb Bryan

Assignment 1:
Write a program which does the computations in the Valentina Musu report
which solves all 3 cases of boundary conditions at x = 0
"""

with open('output_txt.txt', 'w') as file:

    # Defining bar properties
    T_0 = 0  # temperature of bar at x=0 in deg C
    T_L = 100  # temperature of bar at x=L deg C
    L = 1  # length of bar cm
    radius = 0.1  # radius of bar cm
    area = pi*radius**2  # cm^2
    P = 2*pi*radius
    k_0 = 0.5  # Thermal conductivity of the rod
    h = 0.4

    inter_locs = [L/2, 3*L/4]
    Alpha = [4]
    C = [0.01, 0.1, 1, 10, 100]


    def richardsons(Q):
        i = len(Q) - 1
        Qe = (Q[i-1]**2 - Q[i-2]*Q[i]) / (2*Q[i-1]-Q[i-2]-Q[i])
        beta = (np.log((Qe-Q[i-2])/(Qe-Q[i-1])))/np.log(2)
        return [Qe, beta]


    def rich_extra(q, q2, q4):
        q_extr = (q2**2 - q*q4)/(2*q2-q-q4)
        beta = log((q_extr - q2)/(q_extr-q4))/log(2)
        return q_extr, beta

    def get_error(exact, approx):
        return abs((exact-approx)/exact)*100


    def get_beta(exact, approx, approx_2, h, h_2):
        A = log(exact-approx)
        B = log(exact-approx_2)
        C = log(h) - log(h_2)
        return -(A-B)/C


    # For 0 <= x < 1/2
    def get_T_left(x, T_inter, a, inter):
        A = -0*sinh(a*(x-inter)) / sinh(a*inter)
        B = (T_inter*sinh(a*x)/sinh(a*inter))
        return A + B


    # For 1/2 < x <= 1
    def get_T_right(x, T_inter, a, inter):
        A = -T_inter*sinh(a*(x-1))/sinh(a*(1-inter))
        B = 100*sinh(a*(x-inter))/sinh(a*(1-inter))
        return A + B


    # Function for T at x = 1/2
```

```
    def get_T_inter(a_0, k_0, a_1, k_1, inter):
        A = (a_0*cosh(a_0*inter)/sinh(a_0*inter))*k_0
        B = (a_1*cosh(a_1*(inter-1))/sinh(a_1*(1-inter)))*k_1
        C = (100*a_1/sinh(a_1*(1-inter)))*k_1
        return C/(A+B)


    # Function for determining alpha 1
    def get_a_1(k_1):
        return sqrt(2*0.4/(radius*k_1))


    def get_q_dot_exact(a_1, k_1, area, T_inter, inter):
        A = -T_inter*a_1/sinh(a_1*(1-inter))
        B = 100*a_1*cosh(a_1*(1-inter))/sinh(a_1*(1-inter))
        return -k_1*area*(A+B)


    def get_q_dot_FDM(k_1, area, T_N, T_N_1, h, P, dx):
        return -((-k_1*area*T_N_1)+(k_1*area+((h*P)*dx**2)/2)*T_N)/dx


    # Analytical Solution
    Q_exact_data = []
    exact_data = []
    for interface in inter_locs:
        # print(interface)
        for a in Alpha:
            for k in C:
                k_1 = 0.5*k
                a_1 = get_a_1(k_1)
                T_inter = get_T_inter(a, k_0, a_1, k_1, interface)
                for i in range(3, 11):
                    n_elem = 2 ** i
                    dx = L / n_elem

                    for node in range(n_elem+1):
                        x = dx*node
                        if x < interface:
                            T = get_T_left(x, T_inter, a, interface)
                            exact_data.append({"interface": interface, "alpha": a, "x": x, "Temperature": T,
"Nodes": n_elem, "k": k})
                        elif x > interface:
                            T = get_T_right(x, T_inter, a_1, interface)
                            exact_data.append({"interface": interface, "alpha": a, "x": x, "Temperature": T,
"Nodes": n_elem, "k": k})
                        elif x == interface:
                            exact_data.append({"interface": interface, "alpha": a, "x": x, "Temperature":
T_inter, "Nodes": n_elem, "k": k})

                    q_dot = get_q_dot_exact(a_1, k_1, area, T_inter, interface)
                    Q_exact_data.append({"interface": interface, "alpha": a, "Q dot": q_dot, "k": k, "Nodes":
n_elem})
    # FDM
    Q_FDM_data = []
    FDM_data = []
    for interface in inter_locs:
        for i in range(3, 11):
            n_elem = 2 ** i
            for a in Alpha:
                for k in C:
                    dx = L / n_elem
                    k_1 = 0.5 * k

                    omega_0 = h*P/(k_0*area)
                    omega_1 = h*P/(k_1*area)
                    kappa_0 = 2 + omega_0 * dx ** 2
                    kappa_1 = 2 + omega_1 * dx ** 2

                    # Creating A matrix
                    A = zeros((n_elem-1, n_elem-1))-eye(n_elem-1, k=-1)-eye(n_elem-1, k=1)
                    for node in range(len(A)):
                        if (node+1)*dx < interface:
                            A[node][node] = kappa_0
                        elif (node+1)*dx == interface:
                            A[node][node-1] = -k_0*area/dx
                            A[node][node] = (k_0*area/dx) + k_1*area/dx + h*P*dx
                            A[node][node+1] = -k_1*area/dx
                        elif (1+node)*dx > interface:
```

```
                            A[node][node] = kappa_1

                    # Creating B matrix
                    B = zeros(n_elem-1)
                    B[0] = 0  # deg C
                    B[-1] = 100  # deg C

                    # Solving
                    T = linalg.solve(A, B)
                    T = concatenate([[0], T])
                    T = append(T, 100)
                    for i in range(len(T)):
                        x = dx*i
                        FDM_data.append({"interface": interface, "alpha": a, "x": x, "Temperature": T[i],
"Nodes": n_elem, "k": k})

                    q_dot_FDM = get_q_dot_FDM(k_1, area, T[-1], T[-2], h, P, dx)
                    Q_FDM_data.append({"interface": interface, "alpha": a, "Q dot": q_dot_FDM, "k": k, "Nodes":
n_elem})

    # Convergence
    for interface in inter_locs:
        for a in Alpha:
            for k in C:

                exact_con = []
                rich_con = []
                q_exact = [item["Q dot"] for item in Q_exact_data if item["alpha"] == a
                           and item["interface"] == interface and item["k"] == k]
                q_approx = [item["Q dot"] for item in Q_FDM_data if item["alpha"] == a
                            and item["interface"] == interface and item["k"] == k]
                nodes = [item["Nodes"] for item in Q_exact_data if item["alpha"] == a
                         and item["interface"] == interface and item["k"] == k]

                dx = L / nodes[0]
                error = get_error(q_exact[0], q_approx[0])
                exact_con.append([dx, q_approx[0], q_exact[0], error, 'Nan'])

                for i in range(len(nodes)-1):
                    dx = L / nodes[i]
                    error = get_error(q_exact[i], q_approx[i])
                    B_exact = get_beta(q_exact[i], q_approx[i], q_approx[i+1], nodes[i], nodes[i+1])
                    exact_con.append([dx, q_approx[i], q_exact[i], error, B_exact])

                for i in range(len(nodes)-2):
                    dx = L/nodes[i+2]
                    q_extr, beta = rich_extra(q_approx[i], q_approx[i+1], q_approx[i+2])
                    error = get_error(q_extr, q_approx[i])
                    rich_con.append([dx, q_approx[i], q_extr, error, beta])

                print('\nConvergence using Richardson extrapolation with k1 = '+str(k_0*k))
                print(tbl.tabulate(rich_con, headers=['dx', 'q_dot(1)', 'q_dot(1)_extra', '% Error', 'Beta']))
                print('\nConvergence using exact solution with k1 = '+str(k_0*k))
                print(tbl.tabulate(exact_con, headers=['dx', 'q_dot(1)', 'q_dot(1)_exact', '% Error', 'Beta']))

                file.write('\nConvergence using Richardson extrapolation with k1 = '+str(k_0*k))
                file.write(tbl.tabulate(rich_con, headers=['dx', 'q_dot(1)', 'q_dot(1)_extra', '% Error',
'Beta']))
                file.write('\nConvergence using exact solution with k1 = '+str(k_0*k))
                file.write(tbl.tabulate(exact_con, headers=['dx', 'q_dot(1)', 'q_dot(1)_exact', '% Error',
'Beta']))

                plt.figure(1)
                x_data = [item[0] for item in rich_con]
                y_data = [item[3] for item in rich_con]
                plt.plot(x_data, y_data, label='k1='+str(k_0*k))

                plt.figure(2)
                x_data = [item[0] for item in exact_con]
                y_data = [item[3] for item in exact_con]
                plt.plot(x_data, y_data, label='k1='+str(k_0*k))

            for i in range(2):
                if i == 0:
                    plt.title('FDM Error Convergence Vs. Extrapolated Solution\nInterface Located at
L*'+str(interface))
                else:
                    plt.title('FDM Error Convergence Vs. Exact Solution\nInterface Located at
L*'+str(interface))
```

```
                plt.legend()
                plt.figure(i+1)
                plt.grid()
                plt.ylabel('Log % Error')
                plt.xlabel('Log dx')
                plt.xscale('log')
                plt.yscale('log')
                plt.gca().invert_xaxis()
            plt.show()

    # Plotting data
    table_data = []
    for interface in inter_locs:
        for i in range(3, 11):
            n_elem = 2 ** i
            for a in Alpha:
                k_table1 = []
                k_table0 = []
                x_data_exact = []
                x_data_FDM = []
                k_headers = ['Position (cm)', 'T, k1 = 0.05', 'T, k1 = 0.5', 'T, k1 = 5', 'T, k1 = 50', 'T, k1
= 500']
                for k in C:

                    # Collecting Data
                    y_data_exact = [item["Temperature"] for item in exact_data if item["alpha"] == a
                                    and item["interface"] == interface and item["k"] == k and
                                    item["Nodes"] == n_elem]

                    x_data_exact = [item["x"] for item in exact_data if item["alpha"] == a
                                    and item["interface"] == interface and item["k"] == k and
                                    item["Nodes"] == n_elem]

                    y_data_FDM = [item["Temperature"] for item in FDM_data if item["alpha"] == a
                                  and item["interface"] == interface and item["k"] == k and
                                  item["Nodes"] == n_elem]

                    x_data_FDM = [item["x"] for item in FDM_data if item["alpha"] == a
                                  and item["interface"] == interface and item["k"] == k and
                                  item["Nodes"] == n_elem]

                    # Printing Tables
                    if i < 5:

                        k_table0.append(y_data_exact)
                        k_table1.append(y_data_FDM)

                        print('\nTemperature Vs. Position Tables for Exact and FDM Analysis')
                        print('k1 = '+str(k_0*k)+' Nodes = '+str(n_elem))
                        print(tbl.tabulate(transpose(array([x_data_exact, y_data_exact, x_data_FDM,
y_data_FDM])),
                                           headers=['Position (cm)', 'T Exact', 'Position (cm)', 'T FDM']))
                    file.write('\nTemperature Vs. Position Tables for Exact and FDM Analysis\nk1 = '+str(k_0*k)
                               + ' Nodes = '+str(n_elem)+tbl.tabulate(transpose(array([x_data_exact,
y_data_exact, x_data_FDM, y_data_FDM])),
                                                                      headers=['Position (cm)', 'T Exact',
'Position (cm)', 'T FDM']))

                    # Plotting Data
                    for i in range(1, 3):
                        plt.figure(i)
                        plt.plot(x_data_FDM, y_data_FDM, '-o', label="FDM: k1 = " + str(0.5 * k))
                        plt.legend()
                    for i in range(2, 4):
                        plt.figure(i)
                        plt.plot(x_data_exact, y_data_exact, '-o', label="Exact: k1 = " + str(0.5 * k))
                        plt.legend()

                k_table0.insert(0, x_data_exact)
                k_table1.insert(0, x_data_FDM)
                if i < 5:
                    print('\nTemperature Vs. Position Tables for Exact Analysis')
                    print('Nodes = ' + str(n_elem))
                    print(tbl.tabulate(transpose(array(k_table0)), headers=k_headers))

                    print('\nTemperature Vs. Position Tables for FDM Analysis')
                    print('Nodes = ' + str(n_elem))
                    print(tbl.tabulate(transpose(array(k_table1)), headers=k_headers))
```

```
                for i in range(1, 4):
                    plt.figure(i)
                    plt.axvline(x=interface, color='grey', linestyle='--')
                    plt.title('Temperature Vs. Position of Bi-metallic Bar\nAlpha='+str(a)+',
Interface='+str(interface)
                              + ', Nodes='+str(n_elem))
                    plt.xlabel('Position (cm)')
                    plt.ylabel('Temperature (\u00B0C)')

                plt.show()
```