

Name: David Delgado
Title: Finite Difference Methods for 1-D Conduction in a Flat Plate
Date: Wednesday, April 17

1. Abstract

The objective of this project was to become familiarized with finite-difference methods in solving heat transfer problems, as well as the accuracy associated with these methods. Convective heat transfer was an important consideration in the design of the finite difference equations for boundary and interior nodes, in addition to one-dimensional heat conduction. The approach taken in this report involves the design of these equations via the general heat equation. Then, these equations were inputted to MATLAB, along with given parameters, to iteratively generate temperature profiles at a given time and the temperature distribution within the plate over time (see Appendix 6.1). An accuracy test was constructed to explore the relationship between the grid size chosen to cover the domain of the plate and the resulting temperature distribution.

The initial grid chosen to cover the domain of the plate consisted of 10 equally spaced nodes. This design proved to be insufficiently fine in that a grid consisting of 100 nodes produced a noticeably different temperature distribution. Hence, a grid consisting of 100 nodes was used to construct the temperature profiles in the plate at 50s, 100s and at steady state. The accuracy of these results was confirmed via an additional accuracy test shown in Figure 6, which shows that the two temperature distributions converge. Furthermore, the temperature profiles (see Figure 2-4) in the plate show the overwhelming effect of the left boundary conditions compared to the right boundary conditions given h_1 approaches ∞ and $h_2 = 0$ W/m²K. At 50s, the left boundary is at 0 °C and the middle and right boundary remain at the initial 30 °C. Only after 1000s does the plate begin to approach steady state as the temperature distribution converges to a uniform 0 °C. At 1000s, the left boundary is at 0 °C, the middle node is at approximately 4.5 °C and the right boundary is at approximately 6.5 °C. This trend shows that as h_1 approaches ∞ , the plate reaches steady state quicker as the temperature distribution across the plate uniformly reaches 0 °C.

2. Equations

The calculation of the temperature distribution across the plate required three unique equations: one equation for temperatures at node 1, one for the interior nodes and one for the maximum node. The derivation of the following equations can be found in Appendix 6.2.

In order to calculate the temperature of the left boundary of the plate over time, an energy balance method was used to derive the following equation:

$$T_1^{P+1} = T_1^P - \frac{\Delta t}{\rho c_p \Delta x} [h_1(T_1^P - T_{1,\infty}) + \frac{k}{\Delta x}(T_1^P - T_2^P)] \quad (1)$$

Similarly, equation 2 was used to calculate the temperature of all interior nodes over time:

$$T_n^{P+1} = T_n^P + \frac{k\Delta t}{\rho c_p (\Delta x)^2} [T_{n-1}^P - 2T_n^P + T_{n+1}^P] \quad (2)$$

The temperature distribution at the right boundary of the flat plate was found using the following:

$$T_{nmax}^{P+1} = T_{nmax}^P + \frac{\Delta t}{\rho c_p \Delta x} \left[\frac{k}{\Delta x} (T_{nmax-1}^P - T_{nmax}^P) - h_2(T_{nmax}^P - T_{2,\infty}) \right] \quad (3)$$

In the equations above, the subscript of T refers to the node and the superscript defines time. Δt is the time step manipulated to satisfy the stability criterion, and Δx is the distance between nodes. The thermal conductivity of the plate, the density of the fluids and the specific heat of the fluids are defined by k , ρ , c_p , respectively. Lastly, h_1 and h_2 are the convection coefficients of the fluids, and $T_{1,\infty}$ and $T_{2,\infty}$ are the temperatures of those fluids (see Figure 1).

3. Experimental Data & Results

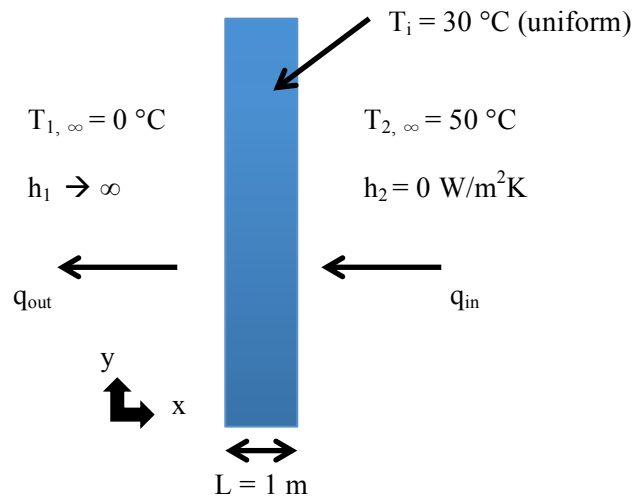


Figure 1: Project Problem Statement

Table 1: Properties of Flat Plate and Fluids

k	1 W/m*K
ρ	2 kg/m ³
c	700 J/kg*K

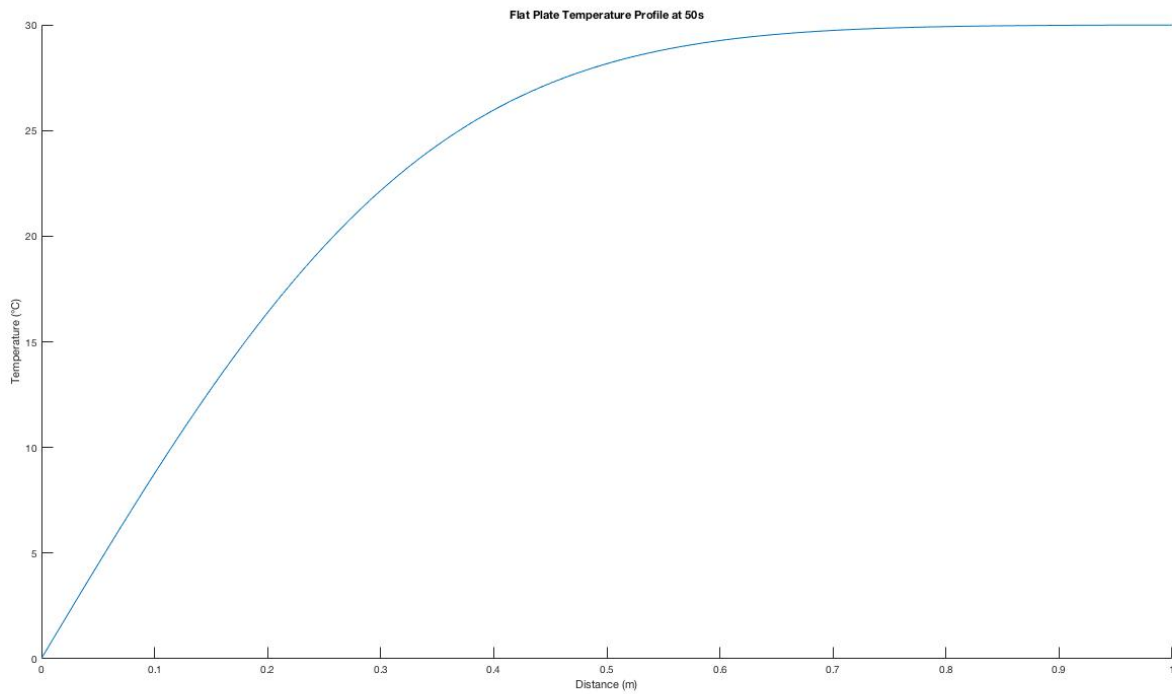


Figure 2: Temperature Profile in the Plate at 50s Using 100 Nodes

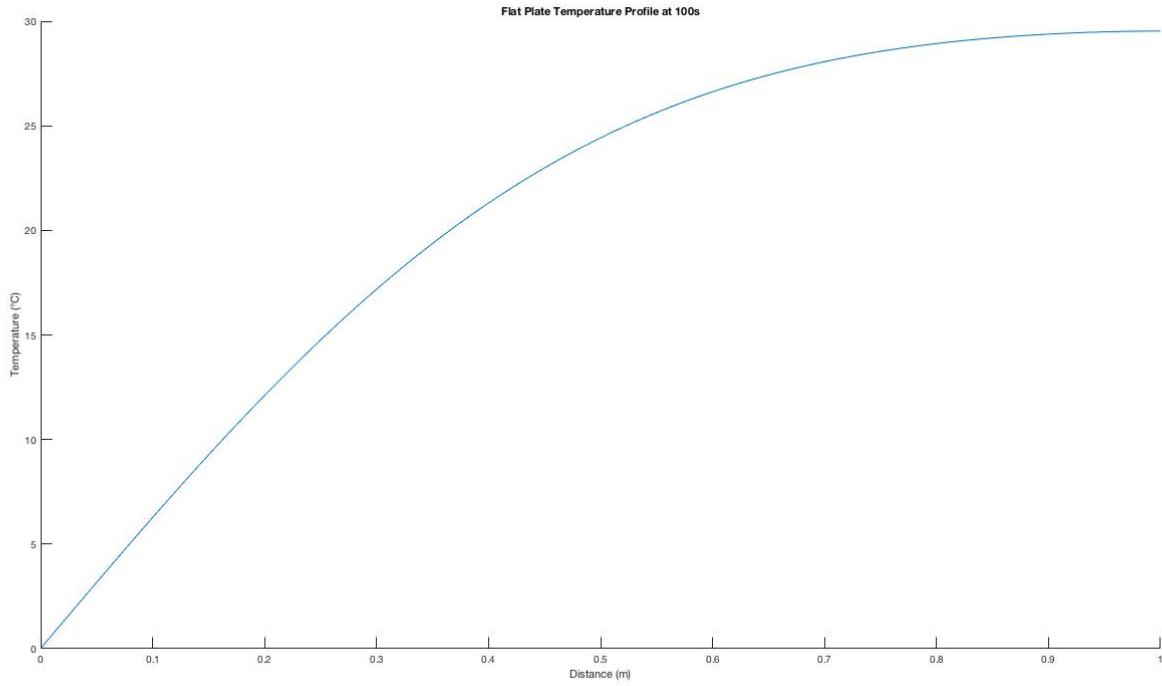


Figure 3: Temperature Profile in the Plate at 100s Using 100 Nodes

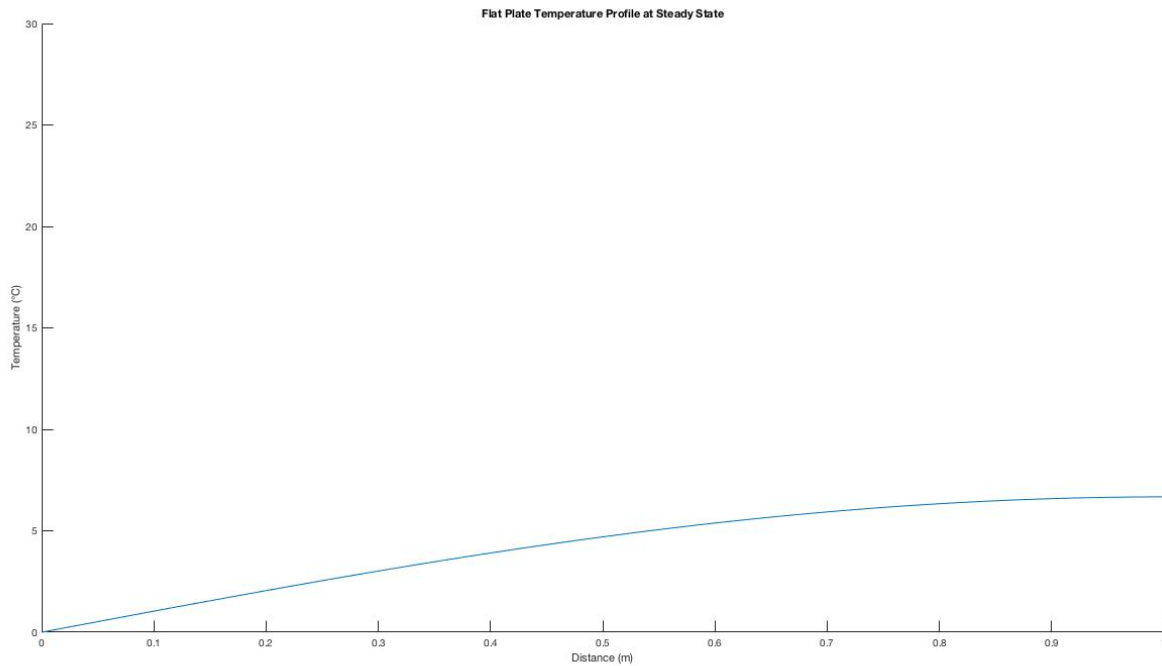


Figure 4: Temperature Profile in the Plate at Steady State Using 100 Nodes

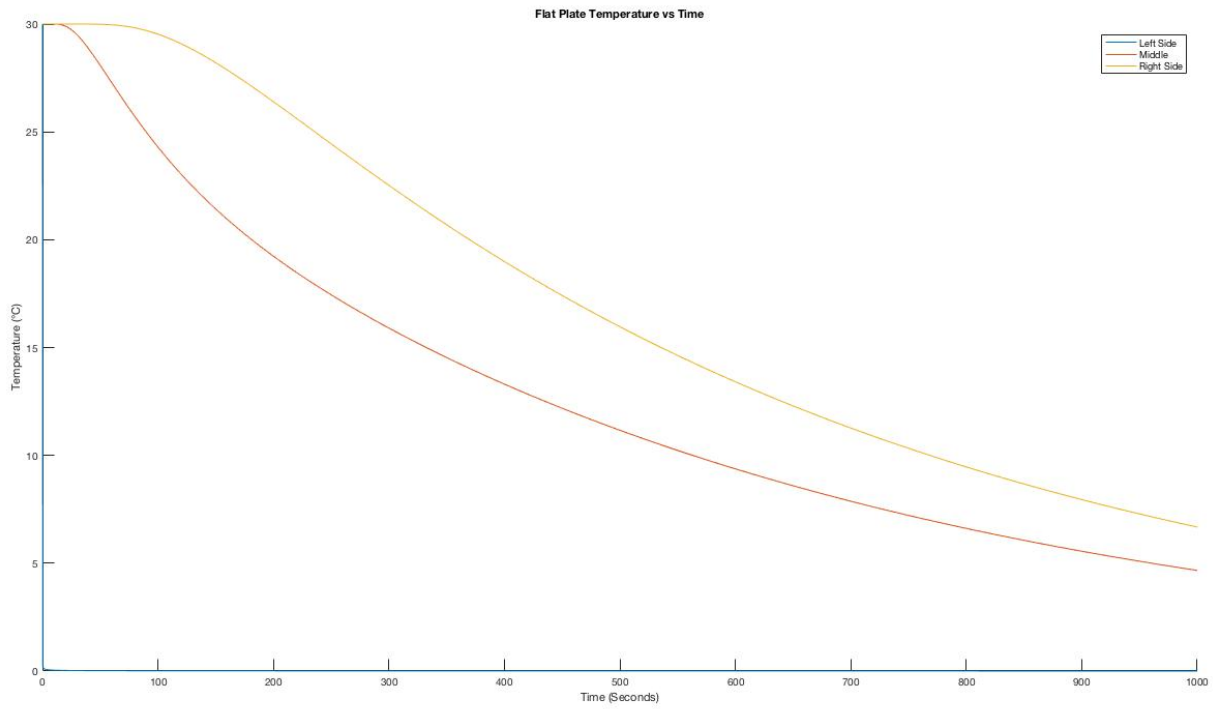


Figure 5: Temperature Distribution in Plate over Time Using 100 nodes

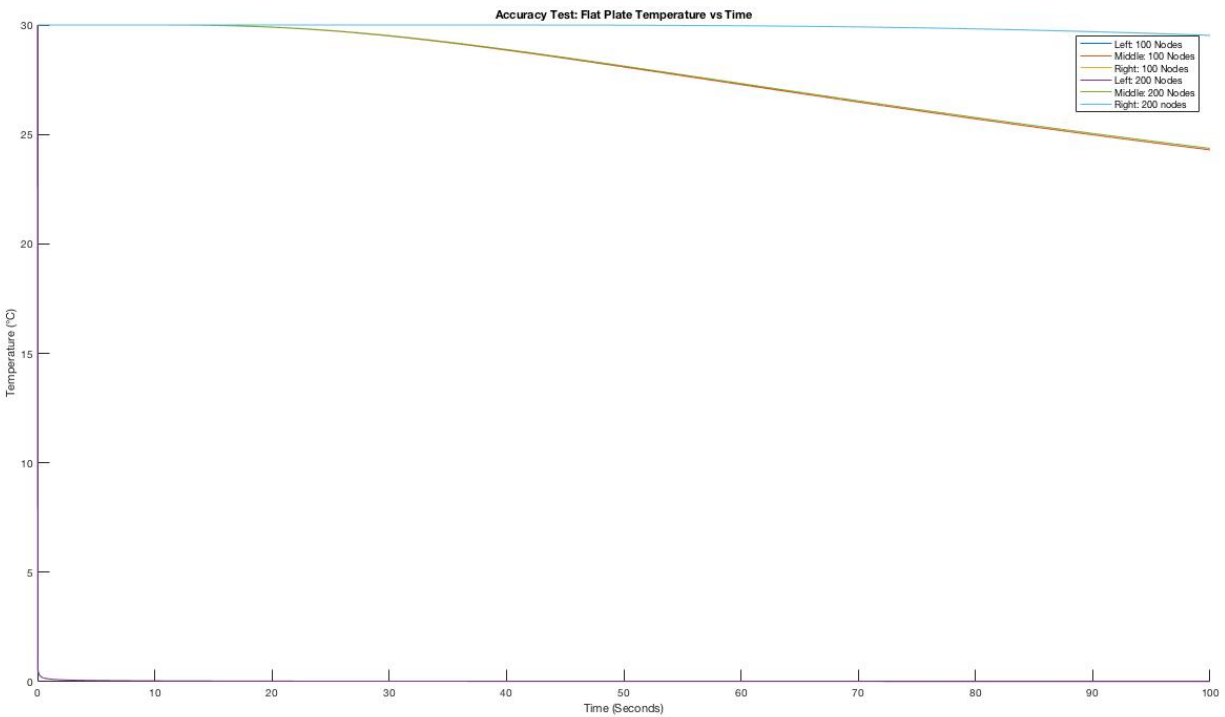


Figure 6: Accuracy Test of Chosen Grid

4. Discussion

From the results of the finite-difference method analysis on the plate, it is apparent that the left boundary conditions have an overwhelming effect on the temperature distribution along the x-axis of the plate with respect to time. This trend is noted in figure 5 where the temperature distribution was constructed using a grid with 100 nodes. It is important to note that left boundary condition stating that h_1 approaches ∞ was not interpreted literally, but rather a large enough convection coefficient was used to simulate the effect of an infinite value. In this case, $h_1 = 50,000 \text{ W/m}^2\text{K}$ was used. The reason that infinity cannot be used in the finite difference calculations is related to the stability criterion of the solution. At an infinite value of h , the time step, Δt , required to achieve a stable solution is so small that it would not be feasible to run the code for the purpose of this analysis. Nevertheless, the results demonstrate that the effect of the right boundary fluid is negligible compared to the left boundary fluid given $h_2 = 0 \text{ W/m}^2\text{K}$, as expected.

Furthermore, the temperature profile in the plate at 50s, 100s and at steady state can be seen in Figure 2,3 and 4, respectively. These three plots are based on a grid consisting of 100 nodes equidistant from each other along the x-axis of the plate. Due to the assumption of one-dimensional heat conduction along the plate, each node along the length of the plate represents a constant temperature along the y-axis. For instance, in figure 2 the temperature profile at 50s indicates that the left boundary reaches $0 \text{ }^\circ\text{C}$ while the right boundary remains at $30 \text{ }^\circ\text{C}$. Therefore, the entire left side of the plate is $0 \text{ }^\circ\text{C}$ after this time and the entire right side is $30 \text{ }^\circ\text{C}$. The plate gradually reaches steady state as can be seen in Figure 5, as the middle and right side of the plate converge to $0 \text{ }^\circ\text{C}$ after 1000s. Figure 4 shows that after 1000s the temperature profile in the plate uniformly reaches lower temperatures. Namely, the right boundary does not exceed $7 \text{ }^\circ\text{C}$. Figure 5 supports that this convergence will eventually lead to a steady state after approximately 30 minutes, and the temperature profile in the plate will no longer be time dependent. At steady state, the temperature profile along the plate will be a flat line at $0 \text{ }^\circ\text{C}$ (i.e., uniform temperature).

In order to be sure of the data presented above, and thereby the analysis drawn from it, an accuracy test was constructed to show that a grid consisting of 100 nodes is fine enough. To achieve this, a second grid consisting of 200 nodes was used to generate a new temperature distribution in the plate over 100s. The use of a finer grid to calculate the second temperature distribution required a smaller time step as nodes increased to 200, resulting in a longer run time for the code. This new temperature distribution was plotted alongside the old one generated from 100 nodes in figure 5. As figure 6 shows, the two aforementioned temperature distributions converge seamlessly according to the location of the boundary and middle nodes. Hence, it is not only accurate to use the temperature distribution and profiles produced by a grid with 100 nodes, but also more efficient.

5. Conclusion

Heat conduction in the plate was considered one-dimensional for the purpose of this project, but there are obvious flaws in the assumptions made in this report. Namely, one would expect for temperature to vary along the y-axis of the plate due to an inconsistent flow over the plate. Furthermore, the convection coefficient used in place of infinity was arbitrarily chosen and only approximates the effect of an infinite value for h_1 . Given an actual infinite convection coefficient on the left boundary, the entire plate would almost instantly reach a uniform temperature of $0 \text{ }^\circ\text{C}$. However, the value for h_1 used in the analysis above produces intuitive results in terms of the trends associated with one-sided convective heat transfer and one-dimensional conduction.

6. Appendix

[Appendix 6.1]

```
%% Finite difference analysis of 1D transient heat transfer (code)
clear
clc
close all
% Define/initialize parameters k, rho, cp, dx and q here
k = 1; % [W/m*K], thermal conductivity
rho = 2; % [kg/m^3], density
cp = 700; % [J/kg*K], specific heat
L = 1; % [m], length of flat plate
nodeNumber = 100; % number of nodes 10 (change for accuracy test)
dx = L/(nodeNumber-1); % [m], distance between nodes in m
h1 = 50000; % [W/m^2*k], left side convective coefficient
h2 = 0; % [W/m^2*k], right side convective coefficient
Ti = 30; % [deg C], initial uniform temp of plate
T1inf = 0; % [deg C], left side fluid temp
T2inf = 50; % [deg C], right side fluid temp

% Outer for loop for time integration
dt = 0.0002; % [s], time step
time = 0:dt:1000; % [s], time vector (change for 50s, 100s and steady state)
TempValues = NaN(length(time), nodeNumber); % preallocate memory (rows,
columns)
TempValues(1, :) = 30; % [deg C], initial conditions

%for loop, starting at row 2 (i.e., after initial temp values)
for p = 2:length(time) % for all time values
    % Inner for loop for calculating nodal temperatures
    for m = 1:nodeNumber % for all nodes, goes m = 1,2,3,4, etc. and repeats
back to outer loop
        if m == 1 %equation for updating time and temp at node 1
            TempValues(p,m) = TempValues(p-1,m)+(dt/(rho*dx*cp))*((-
h1*(TempValues(p-1,m)-T1inf))-((k/dx)*(TempValues(p-1,m)-TempValues(p-
1,m+1))));
        elseif m == 100 %equation for updating time and temp at max node
            TempValues(p,m) = TempValues(p-
1,m)+(dt/(rho*dx*cp))*((k/dx)*(TempValues(p-1,m-1)-TempValues(p-1,m))-
h2*(TempValues(p-1,m)-T2inf));
        else %equation for updating time and temp at all interior nodes
            TempValues(p,m) = TempValues(p-
1,m)+((k*dt)/(rho*cp*(dx)^2))*(TempValues(p-1,m-1)-2*TempValues(p-
1,m)+TempValues(p-1,m+1));
        end
    end
end

%% Plot the temperature distributions over time
figure;
hold on;
aa = plot(time,TempValues(:,1));
ab = plot(time,TempValues(:,50));
ac = plot(time,TempValues(:,100));
title('Flat Plate Temperature vs Time');
```

```
s=['Temperature (' char(176) 'C)'];  
ylabel(s);  
xlabel('Time (Seconds)');  
legend('Left Side','Middle', 'Right Side');  
hold off  
  
%% Temperature profiles at 50s, 100s, steady state  
dist = linspace(0,1,nodeNumber); %distance along the plate in the x-direction  
figure;  
hold on;  
ba = plot(dist, TempValues(250001,:));  
title('Flat Plate Temperature Profile at 50s');  
s=['Temperature (' char(176) 'C)'];  
ylabel(s);  
xlabel('Distance (m)');  
  
figure;  
hold on;  
bb = plot(dist, TempValues(500001,:));  
title('Flat Plate Temperature Profile at 100s');  
s=['Temperature (' char(176) 'C)'];  
ylabel(s);  
xlabel('Distance (m)');  
  
figure;  
hold on;  
bc = plot(dist, TempValues(5000001,:));  
title('Flat Plate Temperature Profile at Steady State');  
s=['Temperature (' char(176) 'C)'];  
ylabel(s);  
xlabel('Distance (m)');  
  
%% Accuracy Test  
nodeNumber2 = 200; %construct distribution using finer grid  
dx2 = L/(nodeNumber2-1);  
dt2 = 0.0002; %adjust time step as needed to satisfy stability criterion  
time2 = 0:dt2:100;  
TempValuesfine = NaN(length(time2), nodeNumber2); % preallocate memory (rows,  
columns)  
TempValuesfine(1, :) = 30; %[deg C], initial conditions  
  
%for loop, starting at row 2 (i.e., after initial temp values)  
for p = 2:length(time2) % for all time values  
    % Inner for loop for calculating nodal temperatures  
    for m = 1:nodeNumber2 % for all nodes, goes m = 1,2,3,4, etc. and repeats  
back to outer loop  
        if m == 1 %equation for updating time and temp at node 1  
            TempValuesfine(p,m) = TempValuesfine(p-1,m)+(dt/(rho*dx2*cp))*((-  
h1*(TempValuesfine(p-1,m)-T1inf))-((k/dx2)*(TempValuesfine(p-1,m)-  
TempValuesfine(p-1,m+1))));  
        elseif m == 200 %equation for updating time and temp at max node  
            TempValuesfine(p,m) = TempValuesfine(p-  
1,m)+(dt/(rho*dx2*cp))*((k/dx2)*(TempValuesfine(p-1,m-1)-TempValuesfine(p-  
1,m))-h2*(TempValuesfine(p-1,m)-T2inf));  
        else %equation for updating time and temp at all interior nodes  
            TempValuesfine(p,m) = TempValuesfine(p-
```



```

1,m)+(k*dt)/(rho*cp*(dx2)^2))*(TempValuesfine(p-1,m-1)-2*TempValuesfine(p-
1,m)+TempValuesfine(p-1,m+1));
    end
end
end

%% Plot the accuracy test
figure;
hold on;
aa2 = plot(time,TempValues(:,1));
ab2 = plot(time,TempValues(:,50));
ac2 = plot(time,TempValues(:,100));
ad = plot(time2,TempValuesfine(:,1));
ae = plot(time2,TempValuesfine(:,100));
af = plot(time2,TempValuesfine(:,200));
title('Accuracy Test: Flat Plate Temperature vs Time');
s=['Temperature (' char(176) 'C)'];
ylabel(s);
xlabel('Time (Seconds)');
legend('Left: 100 Nodes','Middle: 100 Nodes', 'Right: 100 Nodes', ...
'Left: 200 Nodes','Middle: 200 Nodes','Right: 200 nodes');
hold off

% End script
    
```

[Appendix 6.2]

Equation 1:

$$\rho V c_p \frac{dT}{dt} = q_{out} - \left(-kA \frac{dT}{dx} \right)$$

$$\rho V c_p \frac{T_1^{P+1} - T_1^P}{\Delta t} = -h_1 A (T_1^P - T_{1,\infty}) + \frac{kA}{\Delta x} (T_2^P - T_1^P)$$

$$T_1^{P+1} - T_1^P = \frac{\Delta t}{\rho V c_p} \left[-h_1 A (T_1^P - T_{1,\infty}) - \frac{kA}{\Delta x} (T_1^P - T_2^P) \right]$$

Where $V = A * \Delta x$; thus:

$$T_1^{P+1} = T_1^P - \frac{\Delta t}{\rho c_p \Delta x} \left[h_1 (T_1^P - T_{1,\infty}) + \frac{k}{\Delta x} (T_1^P - T_2^P) \right] \quad (1)$$

Equation 2:

$$\rho V c_p \frac{dT}{dt} = -kA \frac{dT}{dx} - \left(-kA \frac{dT}{dx} \right)$$

$$\rho V c_p \frac{T_n^{P+1} - T_n^P}{\Delta t} = -\frac{kA}{\Delta x} (T_n^P - T_{n-1}^P) + \frac{kA}{\Delta x} (T_{n+1}^P - T_n^P)$$

$$T_n^{P+1} - T_n^P = \frac{kA\Delta t}{\rho V c_p \Delta x} [T_{n-1}^P - 2T_n^P + T_{n+1}^P]$$
$$T_n^{P+1} = T_n^P + \frac{k\Delta t}{\rho c_p (\Delta x)^2} [T_{n-1}^P - 2T_n^P + T_{n+1}^P] \quad (2)$$

Equation 3:

$$\rho V c_p \frac{dT}{dt} = -kA \frac{dT}{dx} + q_{in}$$
$$\rho V c_p \frac{T_{nmax}^{P+1} - T_{nmax}^P}{\Delta t} = -\frac{kA}{\Delta x} (T_{nmax}^P - T_{nmax-1}^P) - h_2 A (T_{nmax}^P - T_{2,\infty})$$
$$T_{nmax}^{P+1} - T_{nmax}^P = \frac{\Delta t}{\rho V c_p} \left[\frac{kA}{\Delta x} (T_{nmax-1}^P - T_{nmax}^P) - h_2 A (T_{nmax}^P - T_{2,\infty}) \right]$$
$$T_{nmax}^{P+1} = T_{nmax}^P + \frac{\Delta t}{\rho c_p \Delta x} \left[\frac{k}{\Delta x} (T_{nmax-1}^P - T_{nmax}^P) - h_2 (T_{nmax}^P - T_{2,\infty}) \right] \quad (3)$$