2D Transient Thermal Analysis of a PCB

December 6, 2024

Cole Francis, Darren Hunt,
Aayush Shah, Bhavin Yardi

Abstract

This project focuses on investigating the transient heat transfer behavior on a Printed Circuit Board (PCB) heated by a resistor with uniform heat generation over time. The goal is to deepen the understanding of heat transfer in electronic components and analyze how various physical parameters—such as material properties and dimensions—affect the temperature distribution in PCBs. Through this study, the coupling of discretized transient heat transfer equations across different materials and the temperature evolution as the system reaches steady-state conditions will be examined.

Motivated by the need to better understand thermal management in electronics, this project will also focus on the numerical modeling of the heat transfer process in PCBs. Key design parameters, boundary conditions, and assumptions were applied to create the model. By implementing this model with a 2-D transient framework, the heat transfer through the PCB's thickness will be incorporated, along with the heat transfer across the width of the PCB. This will provide a more comprehensive understanding of the temperature distribution in PCBs over time.

A simplified PCB board will be modeled. This will be a 2D transient analysis consisting of a resistor providing heat generation that is attached to the FR4 board and a copper conductor. The effects of the heat generation from the resistor on the FR4 body and conductor will be analyzed. The top and left side of the board are subject to convection, and the right and bottom side of the board are subject to an adiabatic boundary condition.

Several assumptions were made to simplify and constrain the problem. The effects of radiation to and from the PCB were neglected, since it would be extremely small. Also, the material properties are considered to be isotropic to simplify the problem. Finally, the PCB was considered to be symmetrical about its center axis in order to simplify computations. These assumptions, along with the initial problem specifications, resulted in the governing equations and boundary conditions that were used in the numerical model.

There are different results this study will be analyzing. This includes the transient results after 30s of simulation. Also, this study will go through the temperature contours at various levels of heat generation, analyze the temperature at which the FR4 will reach its glass transition temperature (breaking point) of 403k. Lastly, a grid and time independence analysis will be performed for this study.

Introduction

The management of heat generation and dissipation in electronics is a major limiting factor in the design and implementation of modern electronics and data centers. Excessive heat that is not effectively dissipated can destroy electronic components, leading to large scale data outages and electronic device failures. As the temperature of the Earth has also begun to rise, it has further exacerbated the issue of heat management in data centers. For example, in 2022, record setting temperatures in London and Sacramento led to large scale data center failures for many major tech companies [1]. PCBs that safely dissipate heat are also essential to the development of compact, high-performance devices such as smartphones and medical equipment.

A 2-D transient analysis was chosen to numerically model the heat transfer in a PCB. This choice of analysis allows for heat transfer along both the width and thickness to be studied, providing a clear representation of the heat transfer across the cross section of a PCB. The chosen model will also highlight how the heat distribution in the PCB changes over time. The PCB was simplified to make the problem less numerically complex. The PCB was modelled with only one resistor in the center, and the PCB was assumed to be symmetrical over its center to a symmetric boundary condition could be used. The effects of radiation were also neglected, as the amount of heat transfer conducted by radiation would be extremely small compared to the amount of heat transfer conducted by convection around the PCB and conduction between the components of the PCB.

Problem Domain and Governing Equations

The domain defined for a PCB used in this study is composed of three different components which can be seen in Figure 1; here the resistor is in the top right corner, the FR4 component is on the bottom and the copper conductor is on the top left of the FR4 component. The blue component seen in the figure is ambient air held at a constant temperature. This 2-dimensional domain is simplified to a resistor, conductor, and FR4 material, used as the base for a PCB, although PCBs are much more complex in reality, this domain allows for the analysis of the effects of heat generation from a resistor across the conductor and FR4 component. The overall height of the PCB domain is 0.0016 meters, and the width of the domain is 0.017 meters.

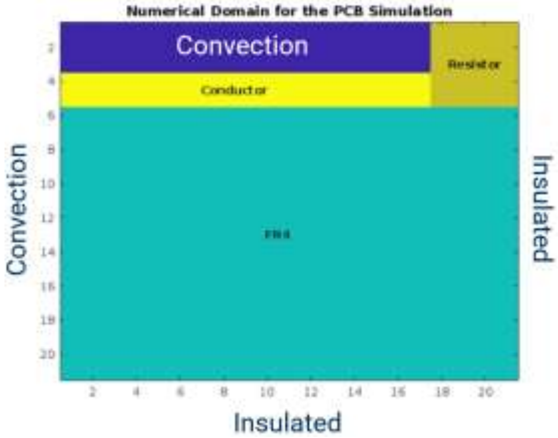

Figure 1: PCB domain with external boundary conditions

Because the domain consists of three components, there are three governing equations that define the problem. These three transient governing equations can be seen below for each component.

Resistor: $$K_R \frac{\partial^2 T}{\partial x^2} + K_R \frac{\partial^2 T}{\partial y^2} + \dot{q} = \rho_R C_{P_R} \frac{\partial T}{\partial t}$$

FR4: $$K_{FR4,x} \frac{\partial^2 T}{\partial x^2} + K_{FR4,y} \frac{\partial^2 T}{\partial y^2} = \rho_{FR4} C_{P_{FR4}} \frac{\partial T}{\partial t}$$

Conductor: $$K_C \frac{\partial^2 T}{\partial x^2} + K_C \frac{\partial^2 T}{\partial y^2} = \rho_C C_{P_C} \frac{\partial T}{\partial t}$$

The material properties used for these governing equations were obtained from previous research or commonly used values for PCBs. These values are shown in the table below:

	Thermal conductivity [W/(m*K)]	Density [Kg/m^2]	Specific heat capacity [J/(kg*K]
Resistor	5	3500	800
FR4	In width: 0.9; In height: 0.3	1850	1100
Conductor	400	8960	385

Two points worth noting are the resistor is the only component that has a heat generating term, and this study looks at the effect on the temperature across the domain for different possible heat generating values. Also, the FR4 material is an anisotropic material meaning that the thermal conductivity across the height and width directions are different. The thermal conductivity in the width is about three times larger than in the height so in this study it is expected to find higher heat dissipation across the width of the FR4 component.

Boundary Conditions and Total Equations

Next, boundary conditions must be considered to be able to formulate equations that can define every node within the domain. On the right external boundary of the resistor and FR4, there is an insulating boundary condition due to symmetry. On the bottom external boundary of the FR4, there is an insulating boundary due to the assumption that the bottom of the PCB is being attached to an insulating material. On all of the other external boundaries there are convection boundary conditions to air held at a constant 298 Kelvin. Between each of the components, there is also a contact resistance considered between the boundaries. Contact resistances were handled initially by creating a node between the boundaries to account for contact resistance, but after difficulties with node sizes, a different approach was used. This second approach consisted of components sharing the same boundary. All the points along the boundary are now being considered in only one of the components. Due to this, the FR4 took on the equations for the boundary and corner nodes of the conductor and resistor and accounted for the contact resistance. Likewise, the resistor took on the other boundary and corner node equations from the conductor that the FR4 material was not in contact with. After all of these considerations, the FR4 had ten total equations consisting of one for internal nodes, five for boundary nodes, and four for corner nodes. The resistor has seven total equations of one for internal nodes, four for boundary nodes, and two for corner nodes. Finally, the conductor had four equations: one for internal nodes, one for the corner node, and two for boundary nodes. The conductor has the least number of equations to define its domain as the FR4 and resistor accounted for all equations for nodes sharing a boundary with the conductor.

Methodology

The governing equations for the materials in the domain were discretized using the Finite Volume Approach. Euler Explicit Method was adopted to iterate through various time steps and get the transient solution. Implicit Method Crank-Nicolson Method was also explored, but owing to the large number of equations and the requirement to invert a matrix at every time was proving to be very computationally expensive. Hence, this study resorted to using the Euler Explicit Method. To maintain the stability of the explicit solver, the adequate time step was computed using the below formulae:

$$\Delta t \leq \frac{\Delta y^2}{4\alpha} \quad \text{where, } \Delta t = \text{maximum time step}$$

Δy = element size in the y-direction

α = thermal diffusivity

Based on the above equation, the maximum value of Δt is found to 10^{-5} s on assuming a grid size of 21 x 21. To achieve a grid of 21x 21, this study assumes the conductor to be 1D. The justification of this assumption lies in the fact that the thickness of the conductor is significantly less in comparison to all other dimensions and the Thermal Conductivity of the conductor is very high, reducing its Thermal resistance to a very low value and allowing a 1D element. Based on this grid size, the element size comes out to $8x10^{-5}$ X $8.5x10^{-4}$m. This study initializes all the temperatures in the domain to the ambient temperature (298k). Owing to a very small time step, the solver takes very long to reach the steady state solution. So, to keep the problem computationally efficient, this study compares all the different values after running the transient solution for 30s. This runs the solver for 3Mn iterations.

Results

Based on running the simulation for 30s, the study shows the following results:

1 – Fig.2. and Fig.3. show the contour of temperature distribution across the PCB. It can be seen that the contours for both the heat generation values match with other with the only difference lying in the temperature ranges for both the cases. Also, through the contours it can be seen that the Top Right corner of the resistor which has adiabatic wall on one side and convection boundary on the other side experiences the maximum temperature, and the bottom left corner of the FR4 which is the farthest away from the heat generating element experiences the least temperature. And the heat very evenly propagates from the resistor to the end of the domain as evident through the temperature distribution in the domain.

2 – Fig.4. shows the variation of the maximum temperature of FR4 with heat generation. This figure shows that the temperature varies linearly with the change in the heat dissipation. Also, from Fig.5. it can be seen that the Glass Transition Temperature of FR4 (403k) is achieved at 38.70W, and this temperature is industry wide considered as the breaking point of FR4.

3 – For Grid Independence, this study compares the results of 21X21 grid points with 51x51 grid points. Mean Squared Error (MSE) is used as a metric to compute the error between the two. MSE of 0.2k is observed. Through this, it can be concluded that grid independence is achieved at grid size of 21x21 as the net change is approximately 1% of our net change in temperatures.

4 – For Time Independence, this study compares the results of Δt = 10^{-5}s with Δt = 10^{-6}s. Mean Squared Error (MSE) is used as a metric to compute the error between the two. MSE of $2.3141*10^{-7}$k is attained. Through this, it can be concluded that time independence is achieved at Δt = 10^{-5}s, as the error is way lower than the net temperature difference observed across the domain.

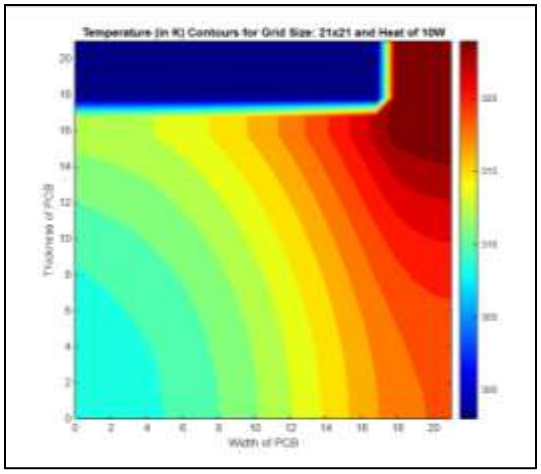

Fig. 2. Temperature Contour for Heat dissipation

Of 10W

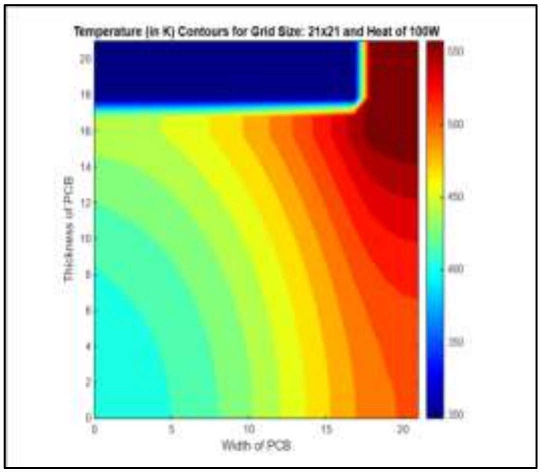

Fig. 3. Temperature Contour for Heat dissipation

of 100W

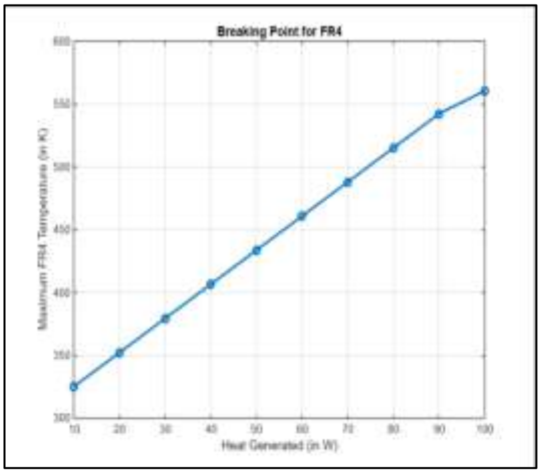

Fig. 4. Variation of Maximum FR4 Temperature

With Heat Generation

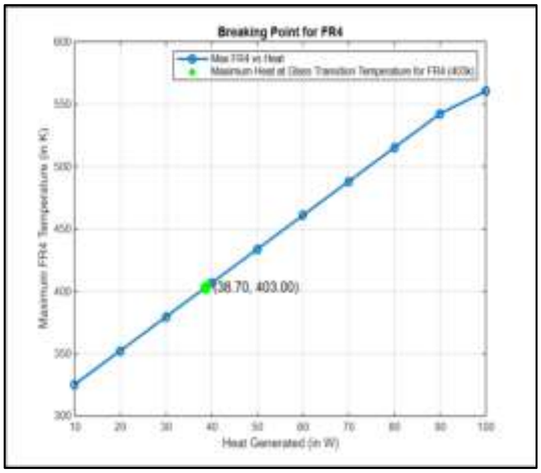

Fig. 5. Maximum Heat dissipated by FR4

Discussions

The outcomes of the simulations demonstrate a strong alignment with the theoretical framework and conceptual understanding of numerical heat transfer. This congruence is evident in several key aspects, which are detailed and analyzed in this section.

To simplify the problem by leveraging symmetrical geometry, the right wall has been designated as adiabatic, ensuring no heat transfer occurs across this boundary. As a result, the temperature gradient in the x-direction ($dT/dx=0$) must be zero along the right wall. This theoretical expectation is validated by the simulated temperature contours, which show that the contour lines are oriented perpendicular to the right boundary. This orientation confirms the absence of a temperature gradient normal to the wall, verifying its insulation and that no thermal energy is exchanged across it. The agreement between the simulation results and the theoretical boundary condition underscores the accuracy of the numerical model in representing adiabatic behavior at this boundary.

A similar phenomenon is observed for the insulated bottom wall, although this wall does not benefit from symmetrical considerations. For the bottom wall, the adiabatic condition ensures that the temperature gradient in the y-direction (dT/dy=0) is zero. This behavior is evident from the simulated temperature contours, which are perpendicular to the bottom boundary, indicating no heat transfer across it. Despite the lack of symmetry, the simulation results align with theoretical expectations, confirming that the bottom wall is effectively insulated.

The accurate representation of adiabatic conditions for both the symmetrical right wall and the asymmetrical bottom wall highlights the robustness of the numerical model. This consistency validates the model's ability to faithfully capture the thermal behavior of insulated boundaries under varying geometrical and physical constraints.

Another important aspect of model validation is the observation of heat stagnation near the right wall. In the case of the FR4 material, the primary modes of heat dissipation are through convection to the ambient fluid at the left edges and conduction to the copper conductor at the top edge. However, as FR4 is an anisotropic material, its thermal conductivity is higher along the horizontal direction than along the vertical. Consequently, the majority of heat dissipation from the FR4 occurs via convection.

The configuration of the heat-generating resistor, which is positioned just above the right section of the FR4, further contributes to this stagnation. Since the heat dissipation predominantly occurs on the left section of the material, a substantial amount of heat accumulates in the right section where there is limited dissipation. The simulated temperature contours clearly illustrate this phenomenon, with noticeably higher temperature values in the right section of the FR4. These elevated temperatures in the right section confirm the expected thermal behavior, where heat remains largely confined due to limited thermal dissipation pathways in that region.

This observation further validates the accuracy of the numerical model, as it replicates the heat stagnation behavior in line with theoretical expectations. The temperature distribution observed in the simulation reflects the material's thermal anisotropy and the configuration of the heat sources and dissipation mechanisms, providing strong evidence for the model's reliability in simulating real-world heat transfer processes.

Methods to improve Heat Transfer Rate

Understanding methods to improve the heat transfer rate in printed PCBs is critical for enhancing the reliability, performance, and longevity of electronic devices. Effective thermal management reduces the risk of component degradation and failure due to excessive heat, ensuring optimal performance and extending the operational lifespan of the PCB. Additionally, improved heat dissipation contributes to energy efficiency by minimizing power losses, allowing for higher-density designs without compromising thermal safety. This knowledge also facilitates cost-effective manufacturing by reducing the need for additional cooling mechanisms and ensuring compliance with industry standards for thermal performance. Therefore, this section of the paper explores various methods to enhance the heat transfer rate in PCBs, emphasizing their impact on overall system efficiency and reliability.

Firstly, an increased heat transfer rate can be achieved by selecting a material with higher thermal conductivity to improve the conduction of heat away from the resistor and FR4. In this study, copper was used as the conductor; however, metals such as silver, which possess even higher thermal conductivity, could lead to a noticeable enhancement in the heat transfer rate.

Another effective method to enhance conductivity between PCB components is by reducing the adverse effects of contact resistance, which occurs due to microscopic inconsistencies in material contact. These microscopic gaps or voids between materials create small air pockets that significantly increase thermal resistance, impeding the efficient transfer of heat. To mitigate this issue, several solutions can be implemented. The first solution involves applying a stronger clamping force to increase the microscopic contact area between the materials. By enhancing the pressure at the interface, the clamping force enlarges the actual contact points at the macroscopic level, resulting in a more substantial area for heat to transfer. This leads to a more efficient heat dissipation process as the enhanced contact area allows for a greater flow of thermal energy between materials. The second solution is the use of high-thermal-conductivity filler pastes, which are introduced between the contacting materials. These pastes fill in the air gaps or pockets that contribute to increased contact resistance. Since the pastes are typically made from materials with superior thermal conductivity, they serve to reduce the thermal resistance at the interface significantly. By filling the microscopic voids, the pastes improve the overall thermal conductivity between the materials, allowing heat to flow more freely. The third approach involves machining the surfaces of the materials in contact to reduce surface roughness. When materials have smoother surfaces, there is less opportunity for air pockets to form at the interface, leading to more uniform contact. This reduction in surface roughness enhances the intimate contact between the materials, which in turn reduces the contact resistance. A more even contact promotes better heat conduction, further improving the overall thermal performance of the PCB.

To consider the convective aspect of heat transfer, enhancing the convective heat transfer rate would also increase the heat dissipation from the FR4 and the resistor. The rate of convective heat transfer can be improved in two primary ways: First, by employing forced convection at the surfaces of the FR4 and the resistor. This can be achieved by using external agents such as fans, which elevate the fluid velocity. This increase in velocity raises the Reynolds number, which in turn boosts the Nusselt number, ultimately enhancing the convective heat transfer coefficient. As the convective heat transfer coefficient increases, more heat can be dissipated. A second method to improve heat transfer by convection is to lower the temperature of the ambient fluid. A reduced fluid temperature creates a greater temperature difference between the PCB and the surrounding environment, thereby increasing the temperature gradient and further enhancing the heat transfer rate.

Future Work

Now that, this study has the base solver for running for analyzing 2D PCB, the team finds the following possible extensions to this study:

1 – Considering the 3D PCB as the domain instead of 2D PCB for analyzing temperature distributions to identify any potential hotspots.

2 – Considering other heat generating elements like transistors and capacitors, to better understand the temperature distribution in PCBs.

3 – Modeling different layers in 2D PCBs to better understand the penetration of heat through these layers.

4 – Implementing Implicit method and running the model at higher Δt to run the model until steady state. This will help to understand the distribution at the steady state.

Conclusion

In conclusion, this project investigates the transient heat transfer behavior in a Printed Circuit Board (PCB) heated by a resistor with uniform heat generation. The study aims to enhance the understanding of thermal management in electronic components by analyzing how material properties and design parameters influence the temperature distribution within the PCB. Through the application of a 2-D transient numerical model, this research examines the heat transfer across different components, incorporating both the PCB's thickness and width. By modeling the effects of heat generation from the resistor on the FR4 board and copper conductor, the study provides valuable insights into the temperature evolution over time, including localized temperature gradients near the heat source during the transient phase and steady-state conditions.

The simplified assumptions made, such as neglecting radiation effects and considering isotropic material properties, allow for a focused analysis of the governing heat transfer processes. The results of this investigation highlight the progression of temperature distribution and the determination of the maximum allowable heat flux to avoid material failure. The findings contribute to a deeper understanding of how heat transfer can be managed in PCB design, ensuring safe operation and supporting the development of more efficient and reliable electronic systems.

References

[1] N. Eddy, "How Heat Waves and AI Challenges Are Piling Pressure on Data Centers," DataCenter Knowledge, (accessed Dec. 4, 2024).

[2] Zhang, Yabin. "Improved numerical-analytical thermal modeling method of the PCB with considering radiation heat transfer and calculation of components' temperature." IEEE Access 9 (2021): 92925-92940.

[3] Kearney, Daniel J., et al. "Thermal performance of a PCB embedded pulsating heat pipe for power electronics applications." Applied Thermal Engineering 98 (2016): 798-809.

[4] Eveloy, Valérie, John Lohan, and Peter Rodgers. "A benchmark study of computational fluid dynamics predictive accuracy for component-printed circuit board heat transfer." IEEE transactions on components and packaging technologies 23.3 (2000): 568-577.

[5] Shen, Yanfeng, et al. "Thermal modeling and design optimization of PCB vias and pads." IEEE Transactions on Power Electronics 35.1 (2019): 882-900.

Appendix

a) Code for Defining all the constants

```
clc
clear
N = 21;
%%Defining all the constants
%For Resistor
```

```
kR = 5;
rhoR = 3500;
cR = 800;
alphaxR = kR/(rhoR*cR);
alphayR = kR/(rhoR*cR);
lR = 0.00255;
tR = 0.0005;
%For FR4
rhoF = 1850;
kxF = 0.9;
kyF = 0.3;
CF = 1100;
alphaxF = kxF/(rhoF*CF);
alphayF = kyF/(rhoF*CF);
lF = 0.017;
tF = 0.0015;
%For Conductor
rhoC = 8960;
kC = 400;
cC = 385;
alphayC = kC/(rhoC*cC);
alphaxC = kC/(rhoC*cC);
tC = 0.0001;
%For Convection
h = 25;
T_inf = 298;
hC = h;
hF = h;
%For Contact Resistances
R1 = 2e-4;
R2 = 1e-4;
R3 = 1e-4;
R1C = R1;
R3C = R3;
R2F = R2;
R3F = R3;
%For deltax and deltay
deltax = lF/(N-1);
deltay = (tF+tR)/(N-1);
deltayC = deltay;
deltaxC = deltax;
deltayF = deltay;
deltaxF = deltax;
```

b) Code for making the grid and visualizing the PCB

```
% clc
% Define number of nodes for each element in l-direction
nlF = round(lF/deltax) + 1;
```

```
nlR = round(lR/deltax) + 1;
nlC = N - nlR;
% Define the number for each element in h-direction
nhF = round(tF / deltay) + 1;
nhR = N - nhF;
nhC = round(tC / deltay);
%Visualizing the domain
pcb = zeros(N,N); % Start with all zeros (background)
% Assign different values to each region
pcb(nhR+1:end, :) = 2; % FR4 region
pcb(1:nhR, nlC + 1:end) = 3; % Resistor region
pcb(nhR - nhC:nhR, 1:nlC) = 4; % Conductor region
% Display the matrix with different colors for each material
imagesc(pcb);
colormap([1 1 1; 0.5 1 0; 1 0.5 0; 0 1 1]); % Custom colors: white, green, orange,
cyan
colorbar;
caxis([0 5]); % Ensures color mapping corresponds to 0, 1, 3, and 5
title('Numerical Domain for the PCB Simulation', 'FontSize', 12, 'FontWeight',
'bold');
% Add text annotations
hold on;
text(N / 2, nhR + (N - nhR) / 2, 'FR4', 'HorizontalAlignment', 'center', ...
'VerticalAlignment', 'middle', 'FontSize', 10, 'Color', 'black', 'FontWeight',
'bold');
text((nlC + nlR / 2) + 0.5, (nhR / 2)+0.5, 'Resistor', 'HorizontalAlignment',
'center', ...
'VerticalAlignment', 'middle', 'FontSize', 10, 'Color', 'black', 'FontWeight',
'bold');
text(nlC / 2, nhR - nhC / 2, 'Conductor', 'HorizontalAlignment', 'center', ...
'VerticalAlignment', 'middle', 'FontSize', 10, 'Color', 'black', 'FontWeight',
'bold');
hold off;
```

c) Code for running the script:

```
clc
clear
%Running the Actual Code
%Defining some parameters and initialising matrices
dt = 1e-6;
T = zeros(N,N);
for i = 1:N
for j = 1:N
T(i,j) = 298;
end
end
T_temp = zeros(N,N);
T_temp = T;
can_continue = true;
```

```
tR = (nhR-1)*deltay;
lC = (nlC-1)*deltax;
heat = 10;
g = 9.8039e+06;
delta = 1e-6;
error = zeros(N,N);
count = 0;
%Running Loop for 30s
T = zeros(N,N);
for i = 1:N
for j = 1:N
T(i,j) = 298;
end
end
T_temp = zeros(N,N);
T_temp = T;
for z = 1: 3e7
count = count+1
for i = 1:N
for j = 1:N
T_temp =
Eqns_ConductorFn(i,j,T,T_temp,dt,nhR,nhC,nlC,alphayC,alphaxC,rhoC,cC,deltayC,hC,delta
xC,T_inf);
T_temp =
Eqns_FR4Fn(i,j,T,T_temp,N,nhR,nlC,alphayF,alphaxF,deltayF,deltaxF,T_inf,dt,rhoF,CF,hF
,kC,kR);
T_temp =
Eqns_ResistorFn(i,j,T,T_temp,N,nlC,nhR,alphayR,deltay,alphaxR,deltax,rhoR,cR,T_inf,g,
h,nhC,dt,kC,R2);
end
end
error = T_temp - T;
max(max(T_temp))
T = T_temp;
end
fprintf("Generated results for \n Grid Size: %d X %d \n Time Step: %f \n Heat Flux:
%d", N,N,dt,heat)
% Step 2: Specify the Excel file name and sheet
filename = 'T_51_1e-6_10_30s.xlsx'; % Include the '.xlsx' extension
sheet = 'Sheet1'; % Specify the sheet name (optional)
% Step 3: Write the array to the Excel sheet
writematrix(T, filename, 'Sheet', sheet);
```

d) Code for Conductor Eqns

```
function [T_temp] =
Eqns_ConductorFn(i,j,T,T_temp,dt,nhR,nhC,nlC,alphayC,alphaxC,rhoC,cC,deltayC,hC,delta
xC,T_inf)
% Top Left Corner
if (i == nhR + 1 - nhC && j ==1)
dT_dt = ((2*hC/(rhoC*cC*deltayC)) + (2*hC/(rhoC*cC*deltaxC)))*T_inf ...
```

```
+ (2*alphayC/(deltayC^2))*T(i+1, j) ...
+ (2*alphaxC/(deltaxC^2))*T(i, j+1) ...
- ((2*hC/(rhoC*cC*deltayC)) + (2*hC/(rhoC*cC*deltaxC)) ...
+ (2*alphayC/(deltayC^2)) + (2*alphaxC/(deltaxC^2))) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% Top Edge
if(i == nhR + 1 - nhC && j>=2 && j<=nlC)
dT_dt = (2*hC/(deltayC*rhoC*cC))*T_inf ...
+ (alphaxC/(deltaxC^2))*T(i, j+1) ...
+ (2*alphayC/(deltayC^2))*T(i+1, j) ...
+ (alphaxC/(deltaxC^2))*T(i, j-1) ...
- ((2*hC/(deltayC*rhoC*cC)) + (alphaxC/(deltaxC^2)) ...
+ (2*alphayC/(deltayC^2)) + (alphaxC/(deltaxC^2))) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
%Left Edge
if(i>=nhR-nhC+1 && i<=nhR && j==1)
dT_dt = (alphayC/(deltayC^2))*T(i-1, j) ...
+ (alphayC/(deltayC^2))*T(i+1, j) ...
+ (2*hC/(rhoC*cC*deltaxC))*T_inf ...
+ (2*alphaxC/(deltaxC^2))*T(i, j+1) ...
- ((alphayC/(deltayC^2)) + (alphayC/(deltayC^2)) ...
+ (2*hC/(rhoC*cC*deltaxC)) + (2*alphaxC/(deltaxC^2))) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% Internal Node
if(i >= nhR - nhC + 2 && i<=nhR && j>=2 && j<=nlC)
dT_dt = (alphayC/(deltayC^2))*T(i+1, j) ...
+ (alphaxC/(deltaxC^2))*T(i, j+1) ...
+ (alphayC/(deltayC^2))*T(i-1, j) ...
+ (alphaxC/(deltaxC^2))*T(i, j-1) ...
- ((2*alphayC/(deltayC^2)) + (2*alphaxC/(deltaxC^2))) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
end
```

e) Code for Resistor

```
function [T_temp] =
Eqns_ResistorFn(i,j,T,T_temp,N,nlC,nhR,alphayR,deltay,alphaxR,deltax,rhoR,cR,T_inf,g,
h,nhC,dt,kC,R2)
alpha_yR = alphayR/(deltay^2);
alpha_xR = alphaxR/(deltax^2);
g_flux = g/(rhoR*cR);
hR_flux_x = 2*h/(rhoR*cR*deltax);
hR_flux_y = 2*h/(rhoR*cR*deltay);
RcR = 1/(rhoR*cR*deltax*((deltax/kC)+R2));
conRes_R1 = 2/(R1*rhoR*cR*deltax);
conRes_R2 = 2/(R2*rhoR*cR*deltay);
%For Top Left Corner
```

```
if (i==1 && j == nlC + 1)
dT_dt = T(i+1,j)*2*alpha_yR + T(i,j+1)*2*alpha_xR - T(i,j)*(2*alpha_yR + 2*alpha_xR +
hR_flux_x + hR_flux_y) + (hR_flux_y + hR_flux_x)*T_inf + g_flux;
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
%For Top Boundary
if(i==1 && j>=nlC+2 && j<=N-1)
dT_dt = T(i+1,j)*alpha_xR + T(i,j-1)*alpha_xR + T(i+1,j)*2*alpha_yR -
T(i,j)*(2*alpha_xR + 2*alpha_yR + hR_flux_y) + hR_flux_y*T_inf + g_flux;
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
%For Top Right Corner
if (i==1 && j==N)
dT_dt = T(i,j-1)*2*alpha_xR + T(i+1,j)*2*alpha_yR - T(i,j)*(2*alpha_xR + 2*alpha_yR +
hR_flux_y) + hR_flux_y*T_inf + g_flux;
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
%For right boundary
if(i>=2 && i<=nhR && j==N)
dT_dt = T(i+1,j)*alpha_yR + T(i-1,j)*alpha_yR + T(i,j-1)*2*alpha_xR -
T(i,j)*(2*alpha_yR + 2*alpha_xR) + g_flux;
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
%For Left Bottom Boundary
if(i>=nhR + 1 -nhC && i<=nhR && j==nlC+1)
dT_dt = T(i-1,j)*alpha_yR + T(i+1,j)*alpha_yR + T(i,j+1)*2*alpha_xR + T(i,j-1)*RcR -
T(i,j)*(2*alpha_yR + 2*alpha_xR + RcR) + g_flux;
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
%For Left Top boundary
if(i>=2 && i<=nhR-nhC && j==nlC+1)
dT_dt = T(i-1,j)*alpha_yR + T(i+1,j)*alpha_yR + T(i,j+1)*2*alpha_xR -
T(i,j)*(2*alpha_yR + 2*alpha_xR + hR_flux_x) + hR_flux_x*T_inf + g_flux;
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
%For internal Nodes
if(i>=2 && i<=nhR && j>=nlC+2 && j<=N-1)
dT_dt = T(i+1,j)*(alpha_yR) + T(i,j+1)*(alpha_xR) + T(i-1,j)*alpha_yR + T(i,j-
1)*alpha_xR - T(i,j)*(2*alpha_xR + 2*alpha_yR) + g_flux;
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
end
```

f) Code for FR4

```
function [T_temp] =
Eqns_FR4Fn(i,j,T,T_temp,N,nhR,nlC,alphayF,alphaxF,deltayF,deltaxF,T_inf,dt,rhoF,CF,hF
,kC,kR)
% (8) Top Left Corner
if(i == nhR+1 && j==1)
```

```
dT_dt = (2 * alphayF / deltayF^2) * T(i+1, j) ...
+ (2 * hF / (rhoF * CF * deltaxF)) * T_inf ...
+ (2 * kC / (rhoF * CF * (deltayF^2))) * T(i-1, j) ...
+ (2 * alphaxF / deltaxF^2) * T(i, j+1) ...
- ((2 * alphayF / deltayF^2) + (2 * hF / (rhoF * CF * deltaxF)) + (2 * kC / (rhoF *
CF * (deltayF^2))) + (2 * alphaxF / deltaxF^2)) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% (6) External - Under Conductor
if(i == nhR+1 && j>=2 && j<=nlC)
dT_dt = (2 * alphayF / deltayF^2) * T(i+1, j) ...
+ (alphaxF / deltaxF^2) * T(i, j-1) ...
+ (2 * kC/ (rhoF * CF * (deltayF^2))) * T(i-1, j) ...
+ (alphaxF / deltaxF^2) * T(i, j+1) ...
- ((2 * alphayF / deltayF^2) + (2 * alphaxF / deltaxF^2) + (2 * kC/ (rhoF * CF *
(deltayF^2)))) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% (3) External - Under Resistor
if(i==nhR+1 && j >=nlC+1 && j<=N-1)
dT_dt = (2 * alphayF / deltayF^2) * T(i+1, j) ...
+ (alphaxF / deltaxF^2) * T(i, j-1) ...
+ (2 * kR/ (rhoF * CF * (deltayF^2))) * T(i-1, j) ...
+ (alphaxF / deltaxF^2) * T(i, j+1) ...
- ((2 * alphayF / deltayF^2) + (2 * alphaxF / deltaxF^2) + (2 * kR/ (rhoF * CF *
(deltayF^2)))) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% (4) Top Right Corner
if(i == nhR+1 && j==N)
dT_dt = (2 * alphayF / deltayF^2) * T(i+1, j) ...
+ (2 * alphaxF / deltaxF^2) * T(i, j-1) ...
+ (2 *kR/ (rhoF * CF * (deltayF^2))) * T(i-1, j) ...
- ((2 * alphayF / deltayF^2) + (2 * alphaxF / deltaxF^2) + (2 *kR/ (rhoF * CF *
(deltayF^2)))) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% (2) External - Right Side
if(i>=nhR+2 && i<=N-1 && j==N)
dT_dt = (alphayF / deltayF^2) * T(i+1, j) ...
+ (2 * alphaxF / deltaxF^2) * T(i, j-1) ...
+ (alphayF / deltayF^2) * T(i-1, j) ...
- ((2 * alphayF / deltayF^2) + (2 * alphaxF / deltaxF^2)) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% (5) Bottom Right Corner
if(i==N && j==N)
dT_dt = (2 * alphayF / deltayF^2) * T(i-1, j) ...
+ (2 * alphaxF / deltaxF^2) * T(i, j-1) ...
- ((2 * alphayF / deltayF^2) + (2 * alphaxF / deltaxF^2)) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
```

```
end
% (10) External - Bottom Edge
if(i==N && j>=2 && j<=N-1)
dT_dt = (2 * alphayF / deltayF^2) * T(i-1, j) ...
+ (alphaxF / deltaxF^2) * T(i, j+1) ...
+ (alphaxF / deltaxF^2) * T(i, j-1) ...
- ((2 * alphayF / deltayF^2) + (2* alphaxF / deltaxF^2)) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% (9) Bottom Left Corner
if(i==N && j==1)
dT_dt = (2 * hF / (rhoF * CF * deltaxF)) * T_inf ...
+ (2 * alphayF / deltayF^2) * T(i-1, j) ...
+ (2 * alphaxF / deltaxF^2) * T(i, j+1) ...
- ((2 * hF / (rhoF * CF * deltaxF)) + (2 * alphayF / deltayF^2) + (2 * alphaxF /
deltaxF^2)) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% (7) External - Left Side
if(i>=nhR+2 && i<=N-1 && j==1)
dT_dt = (alphayF / deltayF^2) * T(i+1, j) ...
+ (2 * hF / (rhoF * CF * deltaxF)) * T_inf ...
+ (alphayF / deltayF^2) * T(i-1, j) ...
+ (2 * alphaxF / deltaxF^2) * T(i, j+1) ...
- ((2 * alphayF / deltayF^2) + (2 * alphaxF / deltaxF^2) + (2 * hF / (deltaxF* rhoF *
CF))) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
% (1) Internal Nodes
if (i>=nhR+2 && i<=N-1 && j>=2 && j<=N-1)
dT_dt = (alphayF / deltayF^2) * T(i+1, j) ...
+ (alphaxF / deltaxF^2) * T(i, j+1) ...
+ (alphayF / deltayF^2) * T(i-1, j) ...
+ (alphaxF / deltaxF^2) * T(i, j-1) ...
- ((2 * alphayF / deltayF^2) + (2 * alphaxF / deltaxF^2)) * T(i, j);
T_temp(i,j) = T(i,j) + dt*dT_dt;
end
end
```