

3D THERMAL ANALYSIS OF PASSIVE COOLING METHODS OF PV PANELS

Group 9

Hari Bharadwaj Narayanan, Anand Tripathi, Yi-Chen Chen, Mingjun Li

3D Thermal Analysis of Passive cooling methods of PV panels

Group 9: Hari Bharadwaj Narayanan, Anand Tripathi, Yi-Chen Chen, Mingjun Li

1. Abstract

Photovoltaic (PV) panels suffer efficiency losses with increasing operating temperatures. This study investigates passive cooling strategies using heatsink designs to mitigate thermal effects. A 3D steady-state heat transfer model evaluates the influence of fin thickness, thermal conductivity, and ambient temperature on cooling performance. Findings identify optimal design parameters for enhanced heat dissipation, offering valuable insights into material and structural trade-offs. Additionally, preliminary efforts to analyze a 3D transient fin provide a further understanding of dynamic thermal behavior. These results contribute to the development of efficient, cost-effective cooling solutions for PV systems, promoting improved performance and longevity

2. Introduction

2.a. Background and Schematic

PV panels lose efficiency at high temperatures, but fins in heatsinks provide an effective passive cooling solution [1]. They increase surface area, enhancing heat dissipation without requiring external energy, making them a cost-efficient way to maintain optimal panel performance. In this study, we are only analyzing a part of a panel-fin system that involves only a single fin. Symmetric conditions have been considered to reduce the computational load. The schematic of the domain considered is shown below in Figure 1.

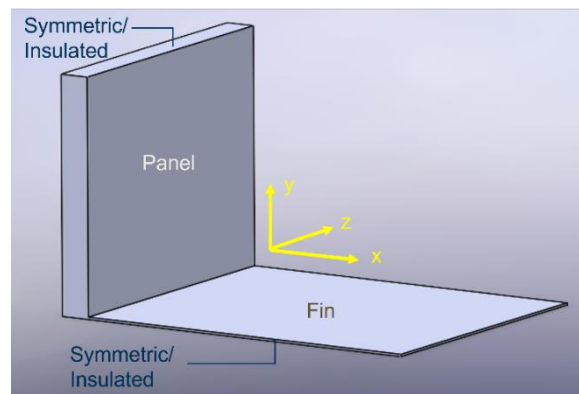


Figure 1. Schematic

2. b. Governing equations and Steady-state discretization

The governing equation involved is the general 3D steady-state heat conduction equation (with no heat generation):

General Conduction equation

Since the problem predominantly involves incoming heat flux and convection, the energy balance would look like this:

$$\dot{q}_{conduction} + \dot{q}_{convection} + \dot{q}_{radiation} = \frac{dU}{dt}$$

Simple energy balance

In this study, the heat transfer has been discretized using the central difference method, which involves averaging the temperature of the surrounding nodes to find the temperature of the node under consideration. The actual discretization of each node varies depending on the location and the boundary conditions. For example, a corner node in a 3D setting with incoming heat flux and convective boundary conditions would be discretized as:

$$k \cdot \left[\Delta y \Delta z \cdot \left(\frac{T_{i+1,j,k} - 2T_{i,j,k} + T_{i-1,j,k}}{\Delta x} \right) + \Delta x \Delta z \cdot \left(\frac{T_{i,j+1,k} - 2T_{i,j,k} + T_{i,j-1,k}}{\Delta y} \right) + \Delta x \Delta y \cdot \left(\frac{T_{i,j,k+1} - 2T_{i,j,k} + T_{i,j,k-1}}{\Delta z} \right) \right] + A_{conv} q_{convection} + A_{rad} q_{radiation} = 0$$

One steady state discretization for corner node (1,1,1)

Similar discretization has been done for other unique and relevant nodes. To be exact, in this study, 50 unique nodes (8 surfaces, 23 edges, 18 corners and one interior node) were identified in the domain and appropriately discretized. These discretized equations were then used to analyse the heat transfer phenomena happening through the system.

$$\Delta y \Delta z \cdot \left[h \cdot (T_{\infty} - T_{1,1,1}) + k \cdot \left(\frac{T_{2,1,1} - T_{1,1,1}}{\Delta x} \right) \right] + \Delta x \Delta z \cdot k \cdot \left(\frac{T_{1,2,1} + T_{1,1,1}}{\Delta y} \right) + \Delta x \Delta y \cdot \left[h \cdot (T_{\infty} - T_{1,1,1}) + k \cdot \left(\frac{T_{1,1,2} - T_{1,1,1}}{\Delta z} \right) \right] + q_{radiation} \cdot \frac{\Delta y \Delta z}{4} = 0$$

2. c. Boundary conditions

The computational domain comprises a PV panel attached to a fin. The upper and lower surfaces are assumed to be symmetric and thus insulated. Incoming heat flux from solar radiation is applied uniformly to one of the panel surfaces. Heat flux applied to the panel surface: $Q_{rad} = 800 \text{ W/m}^2$ (with <18% conversion efficiency considered). It was taken that natural convection occurs at an ambient temperature of 293 K (varied for a few cases) for all exposed surfaces, with the heat transfer coefficient to be taken as $10 \text{ W/m}^2 \cdot \text{K}$. A 2D schematic of the boundary conditions taken is shown below in Fig.

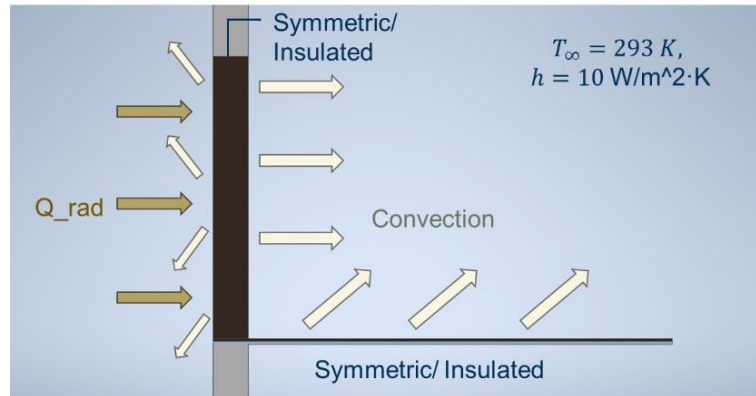


Figure 2. Boundary conditions

3. Methodology

3. a Solution Methodology

The study utilized a structured methodology to analyze the thermal performance of fins in cooling photovoltaic panels. Firstly, the body was discretized using the central difference method to approximate the 3D steady-state heat conduction equation. Next, a grid independence test was done to determine an optimal grid size of 1 mm, ensuring accurate results with computational efficiency. The Gauss-Seidel iterative method was then employed to solve the discretized equations to obtain the required results, with a convergence criterion set at a tolerance of $1e-4$. The main output result that was used to assess the fin performance was the temperature of the panel at the base of the fin, which is referred to as the interface temperature. The study varied fin thickness and thermal conductivity to identify optimal configurations for minimizing interface temperature. Additionally, the effect of ambient temperature variations was analyzed for selected cases to assess its influence on thermal performance. Results in general, were compared against a baseline scenario without fins to quantify their impact on heat dissipation.

3. b. Assumptions

Several assumptions were made to simplify the problem statement considered. First, the radiation heat flux was set at a constant value of 800 W/m^2 , which already accounts for a conversion efficiency of less than 18%. The thermal conductivity of the panel was assumed to be equivalent to that of glass, as the panel material is primarily glass based on the figure 3 given below. The ambient temperature was assumed to remain unchanged over time. The convection

coefficient was treated as constant, while the radiation emitted from the panel was deemed negligible. Additionally, thermal contact resistance was considered negligible to simplify the calculations.

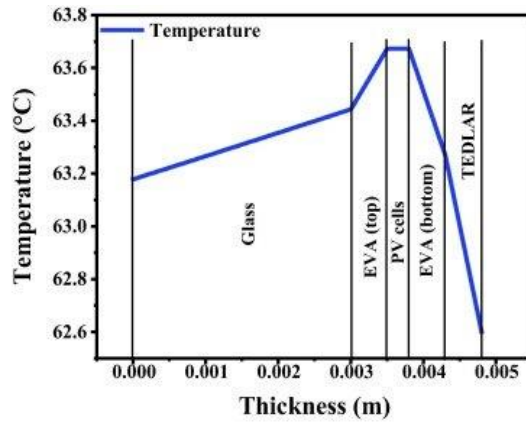


Figure 3 Temperature distribution of PV panel

3. c. System parameters

The table below shows the parameters for the heat sink design, which were developed based on the reference [2].

Table 1: Properties of the system

Parameters		
Air	Convection coefficient h [$W/m^2 \cdot K$]	80
Panel	Thermal conductivity k [$W/m \cdot K$]	1
	Density [kg/m^3]	3000
	Specific heat capacity [$J/kg \cdot K$]	500
Fin	Thermal conductivity k [$W/m \cdot K$]	1/10/100
	Density [kg/m^3]	3000
	Specific heat capacity [$J/kg \cdot K$]	500
	Grid size [mm]	0.1

3.d Grid Independence

A particular case was considered and its interface temperature was checked for various grid sizes to find the optimum grid size. In this study, the grid independence test was done for a fin

thermal conductivity of 10 W/m.K, fin thickness of 4 mm and an ambient temperature of 293 K for the grid sizes of 2mm, 1mm, 0.5mm and 0.25mm as shown below in Figure 4. From the results, 1mm was chosen as the optimum grid to reduce the computational time while increasing accuracy.

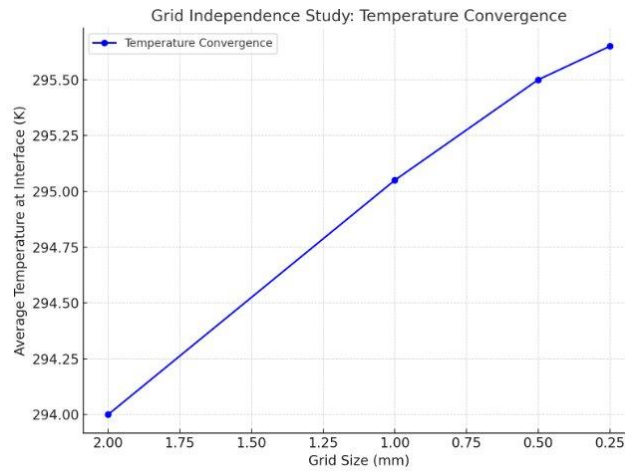


Figure 4: Grid Independence

4. Results and Discussion

4.a. Effect of Thermal Conductivity and Thickness:

The results of the simulation are tabulated in the table given below. The temperature predominantly varied only in the X direction, while it was negligible in the other directions. Thus, the analysis has been done with respect to temperature variation in the X-direction (shown in Fig 5).

Table 2: Simulation Results

	Thermal Conductivity (W/m.K)	Fin Thickness (mm)	Interface Temperature (K)
Unfinned PV panel	N/A	N/A	301.87
Finned PV panel	1	2	299.48
	1	4	299.15
	1	6	298.96
	10	2	295.66
	10	4	295.05

	10	6	294.81
	100	2	293.38
	100	4	293.26
	100	6	293.23

From the table, it can be seen that the interface temperature has indeed been reduced due to the addition of fins to the PV panel. Now, some trends could be observed from the results. Firstly, it must be established that the lower the interface temperature, the better the performance of the fin system. From that perspective, more cooling performance was achieved with greater fin thermal conductivity and thickness. To further elaborate on this, the average interface temperature reduction was about 2.5 K, 6.5 K and 8.5 K for a fin thermal conductivity of 1, 10 and 100 respectively. Meaning, that the greater the thermal conductivity of the fin, the greater the cooling performance of the system. The results are also quite pronounced when it comes to the length of fin taken for the fin temperature to reach equilibrium with the ambient temperature as can be seen in Fig 5. From the figure, it could be observed that the lower the thermal conductivity, the greater is the fin length (beyond the interface (0.02 m)) where temperature variance can be seen. In fact, in the case of the thermal conductivity of the fin is 100 W/m.K, the fin temperature almost reaches ambient immediately beyond the interface. The results are to be expected, as the greater the thermal conductivity, the greater the thermal dissipation along the fin. However, the results were not as pronounced while varying the thickness of the fin. The influence of fin thickness on thermal performance is generally minimal, particularly for cases where the thermal conductivity (k) is 1 W/m.K or 100 W/m.K. A more noticeable effect was observed at fin thermal conductivity = 10 W/m.K, indicating some sensitivity to thickness in this range. Nevertheless, the difference in interface temperature for a particular fin thermal conductivity, while varying the fin thickness was within 1 K, even for a fin thermal conductivity of 10 where maximum variation could be seen. This can be observed from the table provided above but is more clearly illustrated below in Fig 5. In conclusion, for a fin thermal conductivity of 100 W/m.K and a fin thickness (t) of 6 mm, the interface exhibited the lowest average temperature, representing the optimal thermal outcome, but, the temperature reduction with increasing thickness beyond 2 mm was marginal. Considering material efficiency, a fin thickness of 2 mm provides a favorable balance between performance and cost, while having a thermal conductivity of 100 W/m.K.

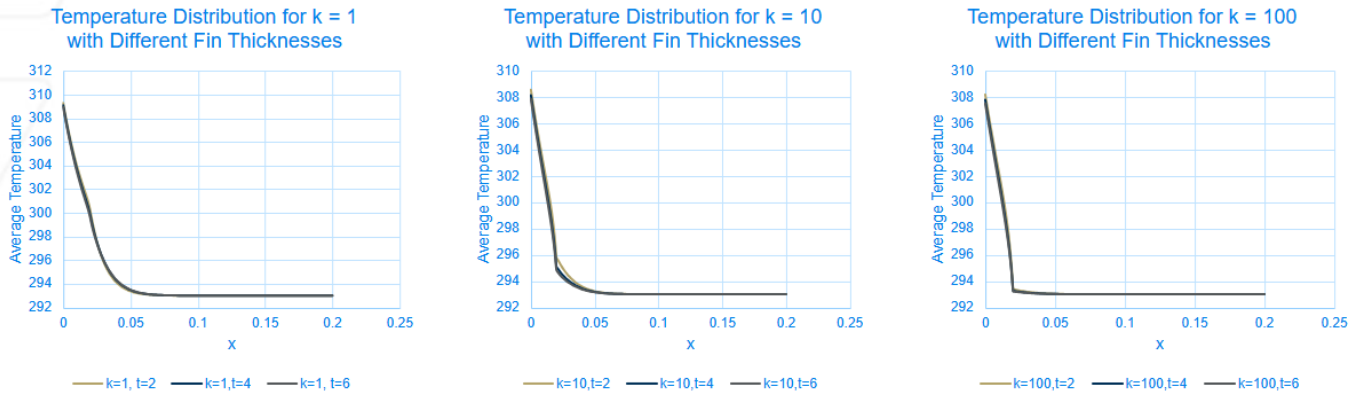


Figure 5 Temperature Distribution with varying k

4. b. Ambient Temperature analysis

The unfinned system and the system with a fin thermal conductivity of 10 W/mK and fin thickness of 2 mm, have been tested for 3 different ambient temperatures of 288 K, 293 K and 300 K, to see the effects of ambient temperature. The results from this analysis are tabulated in the table below. From the table, it could be gathered that the greater the ambient temperature, the greater the interface temperature. Also, the decrease in interface temperature due to the addition of fin varied between 5 to 6 K for all three ambient temperatures considered.

Table 3: Simulation Results for Varying Ambient Temperature

Ambient Temperature (K)	Case	Thermal Conductivity (W/m.K)	Fin Thickness (mm)	Interface Temperature (K)
288 K	Finned	10	2	292.01
	Un-finned	N/A	N/A	298.38
293 K	Finned	10	2	295.66
	Un-finned	N/A	N/A	301.87
298 K	Finned	10	2	301.4
	Un-finned	N/A	N/A	306.13

4. c. Transient Analysis

Time Independence:

- The choosing of time step is crucial to the stability of a transient numerical model using explicit method. The critical time step could be derived from Courant-Friedrichs-Lewy (CFL) condition [3], which is 0.75s and 0.075s for unfinned and finned model respectively. A time independence verification is shown below. A smaller time step will undoubtedly stabilize the results, but it also imposes a greater computational burden. Therefore, 0.1s and 0.01s were chosen as the time step of the model.

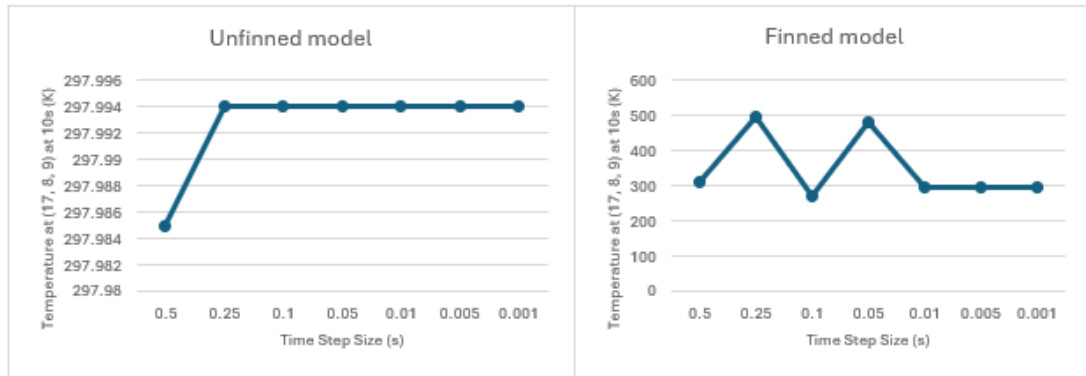


Figure 6: Time Independence

Time Evolution:

- The transient simulations of unfinned heatsink showed rapid temperature stabilization within 200 seconds. Additionally, the fin system consistently outperforming the unfinned case.

5. Conclusion

This study analyzed the thermal performance of fins as passive cooling solutions for photovoltaic panels using a 3D steady-state heat transfer model. Results confirm that the addition of fins significantly reduces interface temperature, improving panel efficiency. The thermal conductivity of the fins had a pronounced impact, with higher conductivities (e.g., 100 W/m·K) achieving superior cooling performance, including rapid temperature equilibrium with the ambient. Fin thickness had a minimal effect, with temperature reductions beyond 2 mm being negligible, suggesting that a thickness of 2 mm offers a cost-effective design. Ambient temperature variations demonstrated a proportional influence on interface temperature, but the effectiveness of the fins in reducing temperature remained consistent across conditions. The transient analysis further supported these findings by illustrating dynamic thermal behaviour. Overall, fins with high thermal conductivity and optimized thickness present a practical, material-efficient solution to mitigate thermal losses in PV systems, promoting their performance and durability.

References:

- [1]: Mahdavi, Arash, et al. "A Review of Passive Cooling of Photovoltaic Devices." *Cleaner Engineering and Technology*, vol. 11, 1 Dec. 2022, p. 100579, www.sciencedirect.com/science/article/pii/S2666790822001847, <https://doi.org/10.1016/j.clet.2022.100579>. Accessed 29 July 2023.
- [2]: Lamaamar, Ibtissam, et al. "Thermal Performance Analysis of a Poly C-Si PV Module under Semi-Arid Conditions." *Materials Science for Energy Technologies*, vol. 5, 2022, pp. 243–251, <https://doi.org/10.1016/j.mset.2022.03.001>. Accessed 13 Feb. 2023.
- [3]: Griffiths, D. V., & Smith, I. M. (2006). *Numerical Methods for Engineers* (2nd ed.). Chapman & Hall/CRC. ISBN: 978-1584884019.

Appendix 1 unfinned steady state:

```
clear;
clc;
% Define physical constants and panel dimensions
L = 0.200; % Length of the panel (m, Y-axis)
W = 0.150; % Width of the panel (m, X-axis)
Z = 0.250; % Height of the panel and fin (m, Z-axis)
nx = 151; % Number of grid points in x
ny = 201; % Number of grid points in y
nz = 251; % Number of grid points in z
dx = W / (nx - 1);
dy = L / (ny - 1);
dz = Z / (nz - 1);
% Material properties for panel
k_pv = 1; % Thermal conductivity (W/m.K)
% Material properties for fin
k_fin = 10; % Thermal conductivity (W/m.K)
% Heat transfer parameters
h = 10; % Convective heat transfer coefficient (W/m^2.K)
Qrad = 800; % Radiative heat input (W/m^2)
% Initial and boundary conditions
T = 298 * ones(nx, ny, nz); % Initial temperature (K)
T_inf_t = 288; % Ambient temperature function
max_iter=10000;
tol=5e-3;
u=((nx-1)*20/(1000*W))+1;
t_fin=4;%in mm
v=t_fin*(ny-1)/(1000*L)+1;
error = 1;
iter=0;
while error > tol && iter < max_iter
    T_new = T; % Store the previous temperature field

    % Update temperature at node T(1,1,1)
    T(1, 1, 1) = ((k_pv *(T_new(2,1,1)+T_new(1,2,1)+T_new(1,1,2)))+... %Convection
    (Qrad * (dx))+... % Radiation term on left YZ
    (h * (T_inf_t) * (2*dx)))/(h*(2*dx)+k_pv*3); % Convection
```

```

%fprintf ("The Temperature is %g\n",T(1,1,1))
% Update temperature at node T(1,201,1)
T(1, ny, 1) = (k_pv * ((T_new(2, ny, 1)) + (T_new(1, ny-1, 1)) + (T_new(1, ny, 2))) + ... % Conduction in X
Qrad * (dx) + ... % Radiation term on left YZ
h * (T_inf_t) * (2*dx)) / (h*(2*dx)+k_pv*3); % Convection
% fprintf ("The Temperature is %g\n",T(1,201,1))

% Update temperature at node T(1,201,251)
T(1, ny, nz) = (k_pv * ((T_new(2, ny, nz)) + (T_new(1, ny-1, nz)) + (T_new(1, ny, nz-1)))) + ... % Conduction in X
Qrad * (dx) + ... % Radiation term on left YZ
h * (T_inf_t) * (2*dx)) / (h*(2*dx)+k_pv*3); % Convection
% Update temperature at node T(1,1,251)
T(1, 1, nz) = (k_pv * ((T_new(2, 1, nz)) + (T_new(1, 2, nz)) + (T_new(1, 1, nz-1)))) + ... % Conduction in X
Qrad * (dx) + ... % Radiation term on left YZ
h * (T_inf_t) * (2*dx)) / (h*(2*dx)+k_pv*3); % Convection

% Loop for nodes on the YZ plane for i=1, j=2:200, k=1
for j = 2:ny-1
    k = 1; % Z = 0 plane (k = 0)
    % Discretized equation for node T(1,j,1)
    T(1, j, k) = ( k_pv * ( 2*T_new(2, j, k) + 2*T_new(1, j, k+1)+T_new(1,j+1,k)+T_new(1,j-1,k))+ ... % Conduction
        Qrad * (2*dx) + ... % Radiation term
        (h * 4*(T_inf_t)*dx)) / (h*4*dx+6*k_pv); % Convection term
end

% Loop for
% nodes on the YZ plane for i=1, j=201, k=2:250
for k = 2:nz-1
    j = ny;
    % Discretized equation for node T(1,j,1)
    T(1, j, k) = ( k_pv * ( 2*T_new(2, j, k) + 2*T_new(1, j-1,k)+T_new(1,j,k+1)+T_new(1,j,k-1))+ ... % Conduction
        Qrad * (2*dx) + ... % Radiation term
        (h * 2*(T_inf_t)*dx)) / (h*2*dx+6*k_pv); % Convection term
end

% Loop for nodes on the YZ plane for i=1, j=2:200, k=251
for j = 2:ny-1
    k = nz;
    % Discretized equation for node T(1,j,1)
    T(1, j, k) = (k_pv*( 2*T_new(2, j, k) + 2*T_new(1, j, k-1)+T_new(1,j+1,k)+T_new(1,j-1,k))+ ... % Conduction
        Qrad * (2*dx) + ... % Radiation term
        (h * 4*(T_inf_t)*dx)) / (h*4*dx+6*k_pv); % Convection term
end

% Loop for nodes on the YZ plane for i=1, j=1, k=2:250
for k = 2:nz-1
    j = 1;
    % Discretized equation for node T(1,j,1)
    T(1, j, k) = ( k_pv * (2*T_new(2, j, k) + 2*T_new(1, j+1,k)+T_new(1,j,k+1)+T_new(1,j,k-1))+ ... % Conduction
        Qrad * (2*dx) + ... % Radiation term
        (h * 2*(T_inf_t)*dx)) / (h*2*dx+6*k_pv); % Convection term
end

% Loop for nodes on the YZ plane for i=1, j=2:200, k=2:250
for k = 2:nz-1
    for j = 2:ny-1
        % Discretized equation for node T(1,j,1)
        T(1, j, k) = (k_pv*(2*T_new(2, j, k)+T_new(1,j+1,k)+T_new(1,j-1,k)+T_new(1,j,k+1)+T_new(1,j,k-1))+ ... %
        Qrad * (2*dx) + ... % Radiation term
        (h * 2*(T_inf_t)*dx)) / (h*2*dx+6*k_pv); % Convection term
    end
end

% Update temperature at node T(26,201,1)
T(u, ny, 1) = ( k_pv * (T_new(u-1, ny, 1) +T_new(u, ny-1, 1)+T_new(u, ny, 2)) + ... % Conduction in Z
h * 2*dx * (T_inf_t)) / (h*(2*dx)+k_pv*3); % Convection

% Update temperature at node T(26,201,251)
T(u, ny, nz) = ( k_pv * (T_new(u-1, ny, nz) +T_new(u, ny-1, nz)+T_new(u, ny, nz-1)) + ... % Conduction in Z
h * 2*dx * (T_inf_t)) / (h*(2*dx)+k_pv*3); % Convection

% Loop for nodes on the YZ plane for i=26, j=201, k=2:250
for k = 2:nz-1
    j = ny;
    % Discretized equation for node T(1,j,1)

```

```

j = ny;
% Discretized equation for node T(1,j,1)
T(u, j, k) = ( k_pv * ( 2*T_new(u-1, j, k) + 2*T_new(1, j-1,k)+T_new(1,j,k+1)+T_new(1,j,k-1))+ ... % Conduction
(h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end
% Loop for nodes on the YZ plane for i=2:25, j=201, k=1
for i = 2:u-1
j = ny;
% Discretized equation for node T(1,j,1)
T(i, j, 1) = ( k_pv *(T_new(i-1, j,1)+T_new(i+1,j,1)+2*T_new(i,j-1,1)+2*T_new(i,j,2))+ ... % Conduction
(h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end
% Loop for nodes on the YZ plane for i=2:25, j=201, k=251
for i = 2:u-1
j = ny;
% Discretized equation for node T(1,j,1)
T(i, j, nz) = ( k_pv *(T_new(i-1, j, nz)+T_new(i+1,j,nz)+2*T_new(i,j-1,nz)+2*T_new(i,j,nz-1))+ ... % Conduction
(h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end
% Loop for nodes on the YZ plane for i=2:25, j=201, k=2:250
for k = 2:nz-1
for i = 2:u-1
% Discretized equation for node T(1,j,1)
T(i, ny, k) = ( k_pv *(T_new(i-1, ny, k)+T_new(i+1,ny,k)+2*T_new(i,ny-1,k)+T_new(i,ny,k-1)+T_new(i,ny,k+1))... % Conduction
)/(6*k_pv); % Convection term
end
end
% Loop for nodes on the YZ plane for i=26, j=4:200, k=1
for j = 2:ny-1
k = 1; % Z = 0 plane (k = 0)
% Discretized equation for node T(1,j,1)
T(u, j, k) = ( k_pv *(2*T_new(u-1, j, k) + 2*T_new(u, j, k+1)+T_new(u,j+1,k)+T_new(u,j-1,k))+ ... % Conduction
(h * 4*(T_inf_t)*dx))/(h*4*dx+6*k_pv); % Convection term
end
% Loop for nodes on the YZ plane for i=26, j=4:200, k=251
for j = 2:ny-1
k = nz; % Z = 0 plane (k = 0)
% Discretized equation for node T(1,j,1)

T(u, j, k) = ( k_pv *(2*T_new(u-1, j, k) + 2*T_new(u, j, k-1)+T_new(u,j+1,k)+T_new(u,j-1,k))+ ... % Conduction
(h * 4*(T_inf_t)*dx))/(h*4*dx+6*k_pv); % Convection term
end
% Loop for nodes on the YZ plane for i=26, j=4:200, k=2:250
for k = 2:nz-1
for j = 2:ny-1
% Discretized equation for node T(1,j,1)
T(u, j, k) = ( k_pv *(2*T_new(u-1, j, k) + T_new(u, j, k-1)+T_new(u,j,k+1)+T_new(u,j+1,k)+T_new(u,j-1,k))+ ... % Conduction
(h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end
end
% Loop over XZ plane for j = 1 k=1
for i = 2:u-1
for k = 1 % Loop over Z direction (top to bottom)
T(i, 1, k) = ( k_pv *(T_new(i-1, 1, k) + T_new(i+1, 1,k)+2*T_new(i,1,k+1)+2*T_new(i,2,k))+ ... % Conduction
(h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end
end
% Loop over XZ plane for j = 1 k=251
for i = 2:nx-1
for k = nz % Loop over Z direction (top to bottom)
T(i, 1, k) = ( k_pv *(T_new(i-1, 1, k) + T_new(i+1, 1,k)+2*T_new(i,1,k-1)+2*T_new(i,2,k))+ ... % Conduction
(h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end
end
% Update temperature at node T(326,1,1)
T(u, 1, 1) = (k_pv *((T_new(u-1, 1, 1)) +(T_new(u, 2, 1)) + (T_new(u, 1, 2))) + ... % Conduction in X
h * (T_inf_t) * (2*dx))/(h*(2*dx+k_pv*3)); % Convection
% Update temperature at node T(nx,1,251)
T(u, 1, nz) = (k_pv *((T_new(u-1, 1, nz)) +(T_new(u, 2, nz)) + (T_new(u, 1, nz-1))) + ... % Conduction in X
h * (T_inf_t) * (2*dx))/(h*(2*dx+k_pv*3)); % Convection
% Loop for nodes on the YZ plane for i=nx, j=1, k=2:250
for k = 2:nz-1
j = 1;
% Discretized equation for node T(1,j,1)
T(u, j, k) = (k_pv *(2*T_new(u-1, j, k)+2*T_new(u, j+1, k)+T_new(u, j, k+1)+T_new(u,j,k-1)) + ... % Conduction in +X

```

```

        (h * (T_inf_t)*2*dx) / (h*2*dx+k_pv*6); % Convection term
    end
    % Loop over XZ plane for j = 1 k=2:250 i=2:235
    for i = 2:u-1
        for k = 2:nz-1 % Loop over Z direction (top to bottom)
            T(i, 1, k) = (k_pv * (T_new(i-1, 1, k)+T_new(i+1,1,k)+2*T_new(i,2,k)+T_new(i,1,k-1)+T_new(i,1,k+1))... % Conduction
                )/(6*k_pv);
        end
    end
    % Loop for nodes on the YZ plane for i=2:25, j=2:200, k=1
    for j = 2:ny-1
        for i=2:u-1
            % Discretized equation for node T(1,j,1)
            T(i, j, 1) = (k_pv * (T_new(i-1, j, 1)+T_new(i+1, j, 1)+T_new(i,j-1,1)+T_new(i, j+1, 1)+2*T_new(i,j,2)) + ... % Conduction in +X
                (h * (T_inf_t)*2*dx) / (h*2*dx+k_pv*6); % Convection termion term
        end
    end
    % Loop for nodes on the YZ plane for i=2:25, j=2:200, k=251
    for j = 2:ny-1
        for i=2:u-1
            % Discretized equation for node T(1,j,1)
            T(i, j, nz) = (k_pv * (T_new(i-1, j, nz)+T_new(i+1, j, nz)+T_new(i,j-1,nz)+T_new(i, j+1, nz)+2*T_new(i,j,nz-1)) + ... % Conduction
                (h * (T_inf_t)*2*dx) / (h*2*dx+k_pv*6); % Convection termion term
        end
    end
end

% Solve for the interior nodes of the panel and fin
for i = 2:(nx - 1)
    for j = 2:(ny - 1)
        for k = 2:(nz - 1)
            if (i < u && j < ny && k < nz) % Panel interior
                T(i, j, k) = ( ((T_new(i+1, j, k) + T_new(i-1, j, k)) + ...
                    (T_new(i, j+1, k) + T_new(i, j-1, k))+ ...
                    (T_new(i, j, k+1) + T_new(i, j, k-1))))/(6);
            elseif (i >= u+1 && i < nx && j < v && k < nz) % Fin interior
                T(i, j, k) = ( ((T_new(i+1, j, k) + T_new(i-1, j, k)) + ...
                    (T_new(i, j+1, k) + T_new(i, j-1, k))+ ...
                    (T_new(i, j, k+1) + T_new(i, j, k-1))))/(6);
            end
        end
    end
end
% Calculate the error (maximum change in temperature)
error = max(abs(T(:) - T_new(:)));
iter = iter + 1;

% Print results every 500 steps
if mod(iter, 10) == 0
    fprintf('Iteration: %d, Error: %.6e\n', iter, error);
end
end

% Extract the temperature data for the YZ plane at x = u
T_YZ_plane = squeeze(T(u, :, :)); % Extract the 2D slice for YZ plane
T_YZ_plane1 = squeeze(T(1, :, :)); % Extract the 2D slice for YZ plane
% Create a meshgrid for the YZ plane
y = linspace(0, v*dx, ny); % Y-axis
z = linspace(0, Z, nz); % Z-axis
[Y, Z2] = meshgrid(y, z);
% Create a meshgrid for the YZ plane
y1 = linspace(0, L, ny); % Y-axis
z1 = linspace(0, Z, nz); % Z-axis
[Y1, Z1] = meshgrid(y1, z1);
for i=u
    for j=1:v
        for k=1:nz
            T_YZ_region = T(i,j,k); % Extract region for Y and full Z
        end
    end
end
for i=1
    for j=1:ny
        for k=1:nz
            T_YZ_region1 = T(i,j,k); % Extract region for Y and full Z
        end
    end
end
% Compute the average temperature in this region
average_temperature = mean(T_YZ_region(:)); % Average of the region
% Print the average temperature with explicit x-coordinate
fprintf('Average Temperature on YZ Plane (X = %.3f m, Y = 0 to %.3f m, Z = 0 to %.3f m): %.2f K\n', ...
    u * dx, v * dy, nz, average_temperature);
% Compute the average temperature in this region
average_temperature = mean(T_YZ_region1(:)); % Average of the region
% Print the average temperature with explicit x-coordinate
fprintf('Average Temperature on YZ Plane (X = 0 m, Y = 0 to %.3f m, Z = 0 to %.3f m): %.2f K\n', ...
    nz, nz, average_temperature);
% Plot the contour of temperature for the full YZ plane
figure(1);
contourf(Z2, Y, T_YZ_plane', 20, 'LineColor', 'none'); % Transpose for correct orientation
colorbar;

```

```

xlabel('Z (m)');
ylabel('Y (m)');
title(sprintf('Temperature Contour on YZ Plane at X = 0.02 m'));
figure(2);
contourf(Z1, Y1, T_YZ_plane1', 20, 'LineColor', 'none'); % Transpose for correct orientation
colorbar;
xlabel('Z (m)');
ylabel('Y (m)');
title(sprintf('Temperature Contour on YZ Plane at X = 0 m'));

```

Appendix 2 finned steady state:

```

clear;
clc;
% Define physical constants and panel dimensions
L = 0.200; % Length of the panel (m, Y-axis)
W = 0.200; % Width of the panel (m, X-axis)
Z = 0.250; % Height of the panel and fin (m, Z-axis)
nx = 201; % Number of grid points in x
ny = 201; % Number of grid points in y
nz = 251; % Number of grid points in z
dx = W / (nx - 1);
dy = L / (ny - 1);
dz = Z / (nz - 1);
% Material properties for panel
k_pv = 1; % Thermal conductivity (W/m.K)
% Material properties for fin
k_fin = 10; % Thermal conductivity (W/m.K)
% Heat transfer parameters
h = 10; % Convective heat transfer coefficient (W/m^2.K)
Qrad = 800; % Radiative heat input (W/m^2)
% Initial and boundary conditions
T = 293 * ones(nx, ny, nz); % Initial temperature (K)
T_inf_t = 293; % Ambient temperature function
max_iter=10000;
tol=5e-3;
u=((nx-1)*20/(1000*W))+1;
t_fin=6;%in mm
v=t_fin*(ny-1)/(1000*L)+1;
error = 1;
iter=0;
while error > tol && iter < max_iter
    T_new = T; % Store the previous temperature field

    % Update temperature at node T(1,1,1)
    T(1, 1, 1) = ((k_pv * (T_new(2,1,1)+T_new(1,2,1)+T_new(1,1,2)))+... %Convection
    (Qrad * (dx))+... % Radiation term on left YZ
    (h * (T_inf_t) * (2*dx)))/(h*(2*dx)+k_pv*3); % Convection

```

```

%fprintf ("The Temperature is %g\n",T(1,1,1))
% Update temperature at node T(1,201,1)
T(1, ny, 1) = (k_pv *((T_new(2, ny, 1)) +(T_new(1, ny-1, 1)))+ (T_new(1, ny, 2))) + ... % Conduction in X
Qrad *(dx) + ... % Radiation term on left YZ
h * (T_inf_t) * (2*dx))/(h*(2*dx)+k_pv*3); % Convection
% fprintf ("The Temperature is %g\n",T(1,201,1))

% Update temperature at node T(1,201,251)
T(1, ny, nz) =(k_pv *((T_new(2, ny, nz)) +(T_new(1, ny-1, nz))+( T_new(1, ny, nz-1))) + ... % Conduction in X
Qrad *(dx) + ... % Radiation term on left YZ
h * (T_inf_t) * (2*dx))/(h*(2*dx)+k_pv*3); % Convection
% Update temperature at node T(1,1,251)
T(1, 1, nz) = (k_pv *((T_new(2, 1, nz)) +(T_new(1, 2, nz))+( T_new(1, 1, nz-1))) + ... % Conduction in X
Qrad *(dx) + ... % Radiation term on left YZ
h * (T_inf_t) * (2*dx))/(h*(2*dx)+k_pv*3); % Convection

% Loop for nodes on the YZ plane for i=1, j=2:200, k=1
for j = 2:ny-1
    k = 1; % Z = 0 plane (k = 0)
    % Discretized equation for node T(1,j,1)
    T(1, j, k) = ( k_pv *( 2*T_new(2, j, k) + 2*T_new(1, j, k+1)+T_new(1,j+1,k)+T_new(1,j-1,k))+ ... % Conduction
    Qrad * (2*dx) + ... % Radiation term
    (h * 4*(T_inf_t)*dx))/(h*4*dx+6*k_pv); % Convection term
end

% Loop for nodes on the YZ plane for i=1, j=201, k=2:250
for k = 2:nz-1
    j = ny;
    % Discretized equation for node T(1,j,1)
    T(1, j, k) = ( k_pv *( 2*T_new(2, j, k) + 2*T_new(1, j-1,k)+T_new(1,j,k+1)+T_new(1,j,k-1))+ ... % Conduction
    Qrad * (2*dx) + ... % Radiation term
    (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end

% Loop for nodes on the YZ plane for i=1, j=2:200, k=251
for j = 2:ny-1
    k = nz;
    % Discretized equation for node T(1,j,1)
    T(1, j, k) = (k_pv*( 2*T_new(2, j, k) + 2*T_new(1, j, k-1)+T_new(1,j+1,k)+T_new(1,j-1,k))+ ... % Conduction
    Qrad * (2*dx) + ... % Radiation term
    (h * 4*(T_inf_t)*dx))/(h*4*dx+6*k_pv); % Convection term
end

% Loop for nodes on the YZ plane for i=1, j=1, k=2:250
for k = 2:nz-1
    j = 1;
    % Discretized equation for node T(1,j,1)
    T(1, j, k) = ( k_pv *(2*T_new(2, j, k) + 2*T_new(1, j+1,k)+T_new(1,j,k+1)+T_new(1,j,k-1))+ ... % Conduction
    Qrad * (2*dx) + ... % Radiation term
    (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end

% Loop for nodes on the YZ plane for i=1, j=2:200, k=2:250
for k = 2:nz-1
    for j = 2:ny-1
        % Discretized equation for node T(1,j,1)
        T(1, j, k)=(k_pv*(2*T_new(2, j, k)+T_new(1,j+1,k)+T_new(1,j-1,k)+T_new(1,j,k+1)+T_new(1,j,k-1))+ ... % Conduction
        Qrad * (2*dx) + ... % Radiation term
        (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
    end
end

% Update temperature at node T(26,201,1)
T(u, ny, 1) = ( k_pv * (T_new(u-1, ...
ny, 1) +T_new(u, ny-1, 1)+T_new(u, ny, 2)) + ... % Conduction in Z
h *2*dx *(T_inf_t))/ (h*(2*dx)+k_pv*3); % Convection

% Update temperature at node T(26,201,251)
T(u, ny, nz) = ( k_pv * (T_new(u-1, ny, nz) +T_new(u, ny-1, nz)+T_new(u, ny, nz-1)) + ... % Conduction in Z
h *2*dx *(T_inf_t))/ (h*(2*dx)+k_pv*3); % Convection

```

```

% Loop for nodes on the YZ plane for i=26, j=201, k=2:250
for k = 2:nz-1
    j = ny;
    % Discretized equation for node T(1,j,1)
    T(u, j, k) = ( k_pv *( 2*T_new(u-1, j, k) + 2*T_new(1, j-1,k)+T_new(1,j,k+1)+T_new(1,j,k-1))+ ... % Conduction
        (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end
% Loop for nodes on the YZ plane for i=2:25, j=201, k=1
for i = 2:u-1
    j = ny;
    % Discretized equation for node T(1,j,1)
    T(i, j, 1) = ( k_pv *(T_new(i-1, j, 1)+T_new(i+1,j,1)+2*T_new(i,j-1,1)+2*T_new(i,j,2))+ ... % Conduction
        (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end
% Loop for nodes on the YZ plane for i=2:25, j=201, k=251
for i = 2:u-1
    j = ny;
    % Discretized equation for node T(1,j,1)
    T(i, j, nz) = ( k_pv *(T_new(i-1, j, nz)+T_new(i+1,j,nz)+2*T_new(i,j-1,nz)+2*T_new(i,j,nz-1))+ ... % Conduction
        (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
end
% Loop for nodes on the YZ plane for i=2:25, j=201, k=2:250
for k = 2:nz-1
    for i = 2:u-1
        % Discretized equation for node T(1,j,1)
        T(i, ny, k) = ( k_pv *(T_new(i-1, ny, k)+T_new(i+1,ny,k)+2*T_new(i,ny-1,k)+T_new(i,ny,k-1)+T_new(i,ny,k+1))... % Conduction
            )/(6*k_pv); % Convection term
    end
end
% Loop for nodes on the YZ plane for i=26, j=4:200, k=1
for j = v+1:ny-1
    k = 1; % Z = 0 plane (k = 0)
    % Discretized equation for node T(1,j,1)
    T(u, j, k) = ( k_pv *(2*T_new(u-1, j, k) + 2*T_new(u, j, k+1)+T_new(u,j+1,k)+T_new(u,j-1,k))+ ... % Conduction
        (h * 4*(T_inf_t)*dx))/(h*4*dx+6*k_pv); % Convection term
end

% Loop for nodes on the YZ plane for i=26, j=4:200, k=251
for j = v+1:ny-1
    k = nz; % Z = 0 plane (k = 0)
    % Discretized equation for node T(1,j,1)
    T(u, j, k) = ( k_pv *(2*T_new(u-1, j, k) + 2*T_new(u, j, k-1)+T_new(u,j+1,k)+T_new(u,j-1,k))+ ... % Conduction
        (h * 4*(T_inf_t)*dx))/(h*4*dx+6*k_pv); % Convection term
end
% Loop for nodes on the YZ plane for i=26, j=4:200, k=2:250
for k = 2:nz-1
    for j = v+1:ny-1
        % Discretized equation for node T(1,j,1)
        T(u, j, k) = ( k_pv *(2*T_new(u-1, j, k) + T_new(u, j, k-1)+T_new(u,j,k+1)+T_new(u,j+1,k)+T_new(u,j-1,k))+ ... % Conduction
            (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
    end
end
% Loop over XZ plane for j = 1 k=1
for i = 2:nx-1
    for k = 1 % Loop over Z direction (top to bottom)
        if i >= 2 && i <= u-1 % PV panel region
            T(i, 1, k) = ( k_pv *(T_new(i-1, 1, k) + T_new(i+1, 1,k)+2*T_new(i,1,k+1)+2*T_new(i,2,k))+ ... % Conduction
                (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
        elseif i >= u+1 && i <= nx-1 % Fin region
            T(i, 1, k) = ( k_fin *(T_new(i-1, 1, k) + T_new(i+1, 1,k)+2*T_new(i,1,k+1)+2*T_new(i,2,k))+ ... % Conduction
                (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_fin); % Convection term
        end
    end
end
% Loop over XZ plane for j = 1 k=251
for i = 2:nx-1
    for k = nz % Loop over Z direction (top to bottom)
        if i >= 2 && i <= u-1 % PV panel region
            T(i, 1, k) = ( k_pv *(T_new(i-1, 1, k) + T_new(i+1, 1,k)+2*T_new(i,1,k-1)+2*T_new(i,2,k))+ ... % Conduction
                (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_pv); % Convection term
        elseif i >= u+1 && i <= nx-1 % Fin region
            T(i, 1, k) = ( k_fin *(T_new(i-1, 1, k) + T_new(i+1, 1,k)+2*T_new(i,1,k-1)+2*T_new(i,2,k))+ ... % Conduction
                (h * 2*(T_inf_t)*dx))/(h*2*dx+6*k_fin); % Convection term
        end
    end
end
end

```

```

% Update temperature at node T(326,1,1)
T(nx, 1, 1) = (k_fin * ((T_new(nx-1, 1, 1)) + (T_new(nx, 2, 1)) + (T_new(nx, 1, 2))) + ... % Conduction in X
h * (T_inf_t) * (2*dx)) / (h*(2*dx)+k_fin*3); % Convection

% Update temperature at node T(326,3,1)
T(nx,v, 1) = (k_fin * ((T_new(nx-1, v, 1)) + (T_new(nx, v-1, 1)) + (T_new(nx, v, 2))) + ... % Conduction in X
h * (T_inf_t) * (3*dx)) / (h*(3*dx)+k_fin*3); % Convection

% Update temperature at node T(326,3,251)
T(nx,v, nz) = (k_fin * ((T_new(nx-1, v, nz)) + (T_new(nx, v-1, nz)) + (T_new(nx, v, nz-1))) + ... % Conduction in X
h * (T_inf_t) * (3*dx)) / (h*(3*dx)+k_fin*3); % Convection

% Update temperature at node T(nx,1,251)
T(nx, 1, nz) = (k_fin * ((T_new(nx-1, 1, nz)) + (T_new(nx, 2, nz)) + (T_new(nx, 1, nz-1))) + ... % Conduction in X
h * (T_inf_t) * (2*dx)) / (h*(2*dx)+k_fin*3); % Convection

% Loop for nodes on the YZ plane for i=nx, j=1, k=2:250
for k = 2:nz-1
    j = 1;
    % Discretized equation for node T(1,j,1)
    T(nx, j, k) = (k_fin * (2*T_new(nx-1, j, k) + 2*T_new(nx, j+1, k) + T_new(nx, j, k+1) + T_new(nx, j, k-1)) + ... % Conduction in +X
    (h * (T_inf_t)*2*dx)) / (h*2*dx+k_fin*6); % Convection term
end
% Loop over XZ plane for j = 1 k=2:250 i=2:235
for i = 2:nx-1
    for k = 2:nz-1 % Loop over Z direction (top to bottom)
        if i >= 2 && i <= u-1 % PV panel region
            T(i, 1, k) = (k_pv * (T_new(i-1, 1, k) + T_new(i+1, 1, k) + 2*T_new(i, 2, k) + T_new(i, 1, k-1) + T_new(i, 1, k+1)) ... % Conduction
            ) / (6*k_pv);
        elseif i >= u+1 && i <= nx-1 % Fin region
            T(i, 1, k) = (k_fin * (T_new(i-1, 1, k) + T_new(i+1, 1, k) + 2*T_new(i, 2, k) + T_new(i, 1, k-1) + T_new(i, 1, k+1)) ... % Conduction
            ) / (6*k_fin);
        end
    end
end

% Loop for nodes on the YZ plane for i=326, j=3, k=2:250
for k = 2:nz-1
    j = v;
    % Discretized equation for node T(1,j,1)
    T(nx, j, k) = (k_fin * (2*T_new(nx-1, j, k) + 2*T_new(nx, j-1, k) + T_new(nx, j, k+1) + T_new(nx, j, k-1)) + ... % Conduction in +X
    (h * (T_inf_t)*4*dx)) / (h*4*dx+k_fin*6); % Convection term
end
% Loop for nodes on the YZ plane for i=326, j=2, k=251
for j = 2:v-1
    k = nz;
    % Discretized equation for node T(1,j,1)
    T(nx, j, k) = (k_fin * (2*T_new(nx-1, j, k) + 2*T_new(nx, j, k-1) + T_new(nx, j+1, k) + T_new(nx, j-1, k)) + ... % Conduction in +X
    (h * (T_inf_t)*4*dx)) / (h*4*dx+k_fin*6); % Convection term
end
% Loop for nodes on the YZ plane for i=nx, j=2, k=1
for j = 2:v-1
    k = 1;
    % Discretized equation for node T(1,j,1)
    T(nx, j, k) = (k_fin * (2*T_new(nx-1, j, k) + 2*T_new(nx, j, k+1) + T_new(nx, j+1, k) + T_new(nx, j-1, k)) + ... % Conduction in +X
    (h * (T_inf_t)*4*dx)) / (h*4*dx+k_fin*6); % Convection term
end
% Loop for nodes on the YZ plane for i=326, j=2, k=2:250
for k = 2:nz-1
    for j = 2:v-1
        % Discretized equation for node T(1,j,1)
        T(nx, j, k) = (k_fin * (2*T_new(nx-1, j, k) + T_new(nx, j, k+1) + T_new(nx, j, k-1) + T_new(nx, j+1, k) + T_new(nx, j-1, k)) + ... % Conduction
        (h * (T_inf_t)*2*dx)) / (h*2*dx+k_fin*6); % Convection term
    end
end
% Loop for nodes on the YZ plane for i=27:325, j=3, k=251
for i = u+1:nx-1
    j = v;
    % Discretized equation for node T(1,j,1)
    T(i, j, nz) = (k_fin * (T_new(i-1, j, nz) + T_new(i+1, j, nz) + 2*T_new(i, j, nz-1) + 2*T_new(i, j-1, nz)) + ... % Conduction in +X
    (h * (T_inf_t)*4*dx)) / (h*4*dx+k_fin*6); % Convection term
end

```

```

% Loop for nodes on the XZ plane for i=27:325, j=3, k=2
for i = u+1:nx-1
    j = v;
    % Discretized equation for node T(1,j,1)
    T(i, j, 1) = (k_fin *(T_new(i-1, j, 1)+T_new(i+1, j, 1)+2*T_new(i,j,2)+2*T_new(i, j-1, 1)) + ... % Conduction in +X
        (h * (T_inf_t)*4*dx)) / (h*4*dx+k_fin*6); % Convection termion term
end
% Loop for nodes on the YZ plane for i=27:325, j=3, k=2:250
for k = 2:nz-1
    for i=u+1:nx-1
        % Discretized equation for node T(1,j,1)
        T(i, v, k) = (k_fin *(T_new(i-1, v, k)+T_new(i+1, v, k)+T_new(i,v,k-1)+T_new(i, v, k+1)+2*T_new(i,v-1,k)) + ... % Conduction in +X
            (h * (T_inf_t)*2*dx)) / (h*2*dx+k_fin*6); % Convection termion term
    end
end
% Loop for nodes on the YZ plane for i=2:25, j=2:200, k=1
for j = 2:ny-1
    for i=2:u-1
        % Discretized equation for node T(1,j,1)
        T(i, j, 1) = (k_pv *(T_new(i-1, j, 1)+T_new(i+1, j, 1)+T_new(i,j-1,1)+T_new(i, j+1, 1)+2*T_new(i,j,2)) + ... % Conduction in +X
            (h * (T_inf_t)*2*dx)) / (h*2*dx+k_pv*6); % Convection termion term
    end
end
% Loop for nodes on the YZ plane for i=2:25, j=2:200, k=251
for j = 2:ny-1
    for i=2:u-1
        % Discretized equation for node T(1,j,1)
        T(i, j, nz) = (k_pv *(T_new(i-1, j, nz)+T_new(i+1, j, nz)+T_new(i,j-1,nz)+T_new(i, j+1, nz)+2*T_new(i,j,nz-1)) + ... % Conduction in
            (h * (T_inf_t)*2*dx)) / (h*2*dx+k_pv*6); % Convection termion term
    end
end
% Loop for nodes on the YZ plane for i=27:325, j=2, k=1
for j = 2:v-1
    for i=u+1:nx-1
        % Discretized equation for node T(1,j,1)
        T(i, j, 1) = (k_fin *(T_new(i-1, j, 1)+T_new(i+1, j, 1)+T_new(i,j-1,1)+T_new(i, j+1, 1)+2*T_new(i,j,2)) + ... % Conduction in +X
            (h * (T_inf_t)*2*dx)) / (h*2*dx+k_fin*6); % Convection term
    end
end
end

% Loop for nodes on the YZ plane for i=27:325, j=2, k=251
for j = 2:v-1
    for i=u+1:nx-1
        % Discretized equation for node T(1,j,1)
        T(i, j, nz) = (k_fin *(T_new(i-1, j, nz)+T_new(i+1, j, nz)+T_new(i,j-1,nz)+T_new(i, j+1, nz)+2*T_new(i,j,nz-1)) + ... % Conduction
            (h * (T_inf_t)*2*dx)) / (h*2*dx+k_fin*6); % Convection termion term
    end
end
% Loop for nodes for i=26, j=2, k=1
for i=u
    for j=2:v-1
        for k=1
            T(i,j,k) = (2*h*dx*T_inf_t + (k_pv+k_fin)*T_new(i,j,k+1) + k_pv*T_new(i-1,j,k) + k_fin*T_new(i+1,j,k) ...
                + ((k_pv+k_fin)/2)*(T_new(i,j-1,k) + T_new(i,j+1,k)))/(2*h*dx + 3*(k_pv+k_fin));
        end
    end
end
% Loop for nodes for i=26, j=2, k=251
for i=u
    for j=2:v-1
        for k=nz
            T(i,j,k) = (2*h*dx*T_inf_t + (k_pv+k_fin)*T_new(i,j,k-1) + k_pv*T_new(i-1,j,k) + k_fin*T_new(i+1,j,k) ...
                + ((k_pv+k_fin)/2)*(T_new(i,j-1,k) + T_new(i,j+1,k)))/(2*h*dx + 3*(k_pv+k_fin));
        end
    end
end
% Loop for nodes for i=26, j=2, k=2:250
for i=u
    for j=2:v-1
        for k=2:nz-1
            T(i,j,k) = (((k_pv+k_fin)/2)*(T_new(i,j-1,k) + T_new(i,j+1,k)) + ((k_pv+k_fin)/2)*(T_new(i,j,k+1) + T_new(i,j,k-1)) ...
                + k_pv*T_new(i-1,j,k) + k_fin*T_new(i+1,j,k))/(3*(k_pv+k_fin));
        end
    end
end
end

```

```

% Loop for nodes for i=26, j=1, k=1
for i=u
    for j=1
        for k=1
            T(i,j,k) = (2*h*dx*T_inf_t + (k_pv + k_fin)*T_new(i,j,k+1) + k_pv*T_new(i-1,j,k) + k_fin*T_new(i+1,j,k) ...
                + (k_pv + k_fin)*T_new(i,j+1,k))/(2*h*dx + 3*(k_pv + k_fin));
        end
    end
end
% Loop for nodes for i=26, j=1, k=251
for i=u
    for j=1
        for k=nz
            T(i,j,k) = (2*h*dx*T_inf_t + (k_pv + k_fin)*T_new(i,j,k-1) + k_pv*T_new(i-1,j,k) + k_fin*T_new(i+1,j,k) ...
                + (k_pv + k_fin)*T_new(i,j+1,k))/(2*h*dx + 3*(k_pv + k_fin));
        end
    end
end
% Loop for nodes for i=26, j=1, k=2:250
for i=u
    for j=1
        for k=2:nz-1
            T(i,j,k) = (((k_pv+k_fin)/2)*(T_new(i,j,k+1)+T_new(i,j,k-1)) + (k_pv+k_fin)*T_new(i,j+1,k) + k_pv*T_new(i-1,j,k) ...
                + k_fin*T_new(i+1,j,k))/(3*(k_pv + k_fin));
        end
    end
end
% Loop for nodes for i=26, j=3, k=1
for i=u
    for j=v
        for k=1
            T(i,j,k) = (k_fin*T_new(i+1,j,k) + 2*k_pv*T_new(i-1,j,k) + 5*h*dx*T_inf_t + (k_fin + 2*k_pv)*T_new(i,j,k+1) ...
                + k_pv*T_new(i,j+1,k) + (k_pv + k_fin)*T_new(i,j-1,k))/(5*h*dx + 6*k_pv + 3*k_fin);
        end
    end
end
end

% Loop for nodes for i=26, j=3, k=1
for i=u
    for j=v
        for k=1
            T(i,j,k) = (k_fin*T_new(i+1,j,k) + 2*k_pv*T_new(i-1,j,k) + 5*h*dx*T_inf_t + (k_fin + 2*k_pv)*T_new(i,j,k+1) ...
                + k_pv*T_new(i,j+1,k) + (k_pv + k_fin)*T_new(i,j-1,k))/(5*h*dx + 6*k_pv + 3*k_fin);
        end
    end
end
% Loop for nodes for i=26, j=3, k=251
for i=u
    for j=v
        for k=nz
            T(i,j,k) = (k_fin*T_new(i+1,j,k) + 2*k_pv*T_new(i-1,j,k) + 5*h*dx*T_inf_t + (k_fin + 2*k_pv)*T_new(i,j,k-1) ...
                + k_pv*T_new(i,j+1,k) + (k_pv + k_fin)*T_new(i,j-1,k))/(5*h*dx + 6*k_pv + 3*k_fin);
        end
    end
end
% Loop for nodes for i=26, j=3, k=251
for i=u
    for j=v
        for k=2:nz-1
            T(i,j,k) = (k_fin*T_new(i+1,j,k) + 2*k_pv*T_new(i-1,j,k) + ((k_fin/2)+k_pv)*(T_new(i,j,k+1)+T_new(i,j,k-1)) ...
                + k_pv*T_new(i,j+1,k) + (k_pv + k_fin)*T_new(i,j-1,k) + 2*h*dx*T_inf_t)/(2*h*dx + 6*k_pv + 3*k_fin);
        end
    end
end
end

% Solve for the interior nodes of the panel and fin
for i = 2:(nx - 1)
    for j = 2:(ny - 1)
        for k = 2:(nz - 1)
            if (i < u && j < ny && k < nz) % Panel interior
                T(i, j, k) = ((T_new(i+1, j, k) + T_new(i-1, j, k)) + ...
                    (T_new(i, j+1, k) + T_new(i, j-1, k)) + ...
                    (T_new(i, j, k+1) + T_new(i, j, k-1)))/(6);
            elseif (i >= u+1 && i < nx && j < v && k < nz) % Fin interior
                T(i, j, k) = ((T_new(i+1, j, k) + T_new(i-1, j, k)) + ...
                    (T_new(i, j+1, k) + T_new(i, j-1, k)) + ...
                    (T_new(i, j, k+1) + T_new(i, j, k-1)))/(6);
            end
        end
    end
end
% Calculate the error (maximum change in temperature)
error = max(abs(T(:) - T_new(:)));
iter = iter + 1;

% Print results every 500 steps
if mod(iter, 500) == 0
    fprintf('Iteration: %d, Error: %.6e\n', iter, error);
end
end

```

```

% Case 1:  $x = 1$ ,  $y \in [0, L]$ ,  $z \in [0, Z]$ 
x_index_1 = 1; % Index for  $x = 1$ 
T_YZ_plane_1 = squeeze(T(1, :, :)); % Extract the 2D slice for the YZ plane
% Create a meshgrid for the full YZ plane at  $x = 1$ 
y_1 = linspace(0, L, ny); % Full Y-axis
z_1 = linspace(0, Z, nz); % Full Z-axis
[Z_1, Y_1] = meshgrid(z_1, y_1);
% Compute the average temperature for the full YZ plane
average_temperature_1 = mean(T_YZ_plane_1(:)); % Average of the region
% Plot the contour for the full YZ plane
figure(1);
contourf(Z_1, Y_1, T_YZ_plane_1, 20, 'LineColor', 'none'); % Transpose for correct orientation
colorbar;
xlabel('Y (m)');
ylabel('Z (m)');
title(sprintf('Temperature Contour on YZ Plane at X = %.3f m', x_index_1 * dx));
fprintf('Case 1: Average Temperature on YZ Plane (X = %.3f m, Y = 0 to %.3f m, Z = 0 to %.3f m): %.2f K\n', ...
    x_index_1 * dx, L, Z, average_temperature_1);
% Case 2:  $x = u$ ,  $y \in [0, t_{fin}]$ ,  $z \in [0, Z]$ 
x_index_u = round(u); % Index for  $x = u$ 
v_index = round(v); % Index for  $y = t_{fin}$ 
T_YZ_plane_u = squeeze(T(u, 1:v, :)); % Extract the 2D slice for the YZ plane
% Limit the region in Y to  $y \in [0, t_{fin}]$ 
% Create a meshgrid for the limited YZ plane at  $x = u$ 
y_u = linspace(0, t_fin, v); % Limited Y-axis
z_u = linspace(0, Z, nz); % Full Z-axis
[Z_u, Y_u] = meshgrid(z_u, y_u);
% Compute the average temperature for the limited YZ plane
average_temperature_u = mean(T_YZ_plane_u(:)); % Average of the region
% Plot the contour for the limited YZ plane
figure(2);
contourf(Z_u, Y_u, T_YZ_plane_u, 20, 'LineColor', 'none'); % Transpose for correct orientation
colorbar;
xlabel('Z (m)');
ylabel('Y (mm)');
title(sprintf('Temperature Contour on YZ Plane at X = 0.2m'));
fprintf('Case 2: Average Temperature on YZ Plane (X = %.3f m, Y = 0 to %.3f m, Z = 0 to %.3f m): %.2f K\n', ...
    x_index_u * dx, t_fin, Z, average_temperature_u);
T1=T(:,v-1,(nz-1)/2);
x=linspace(0,W,nx);
figure(3);
plot(x,T1)
xlabel('x (m)');
ylabel('T(K)');
title('Temperature variation with X-Axis');

```

Appendix 3 unfinned transient:

```

clear;
clc;
% Define physical constants and panel dimensions
L = 0.200; % Length of the panel (m, Y-axis)
W = 0.025; % Width of the panel (m, X-axis)
Z = 0.250; % Height of the panel and fin (m, Z-axis)
nx = 26; % Number of grid points in x
ny = 201; % Number of grid points in y
nz = 251; % Number of grid points in z
dx = W / (nx - 1);
dy = L / (ny - 1);
dz = Z / (nz - 1);

% Material properties for panel
k_pv = 1; % Thermal conductivity (W/m.K)
rho_pv = 3000; % Density (kg/m^3)
cp_pv = 500; % Specific heat capacity (J/kg.K)

% Time-stepping parameters
dt = 0.05; % Time step (s)
t_start = 0; % Start time: 9 AM (seconds)
t_end = 200; % End time: 5 PM (seconds)
time = t_start:dt:t_end;
% Heat transfer parameters
h = 40; % Convective heat transfer coefficient (W/m^2.K)
Qrad = 800; % Radiative heat input (W/m^2)
% Initial and boundary conditions
T = 298 * ones(nx, ny, nz); % Initial temperature (K)
Tinf = 295; % Ambient temperature function
|
dt_panel = rho_pv * cp_pv * (min([dx, dy, dz])^2) / (2 * k_pv);
plot_interval = 10;

videoFilename = 'TemperatureDistributionWithoutFin3.avi';
videoWriter = VideoWriter(videoFilename, 'Motion JPEG AVI');
videoWriter.Quality = 100;
videoWriter.FrameRate = 30;
open(videoWriter);

for t_idx = 1:length(time)

    T_new = T;
    T_inf_t = 295;
    % Update temperature at node T(1,1,1)
    T_new(1, 1, 1) = T(1, 1, 1) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * 2*((T(2, 1, 1) - T(1, 1, 1)) / (dx)^2) + ... % Conduction in X
    k_pv * 2*((T(1, 2, 1) - T(1, 1, 1)) / (dy)^2) + ... % Conduction in Y
    k_pv * 2*((T(1, 1, 2) - T(1, 1, 1)) / (dz)^2) + ... % Conduction in Z
    (Qrad / (dx/2) + ... % Radiation term on left YZ
    h * (T_inf_t - T(1, 1, 1)) * ((2/dx)+(2/dz))); % Convection

    % Update temperature at node T(1,201,251)
    T_new(1, 201, 251) = T(1, 201, 251) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * 2*((T(2, 201, 251) - T(1, 201, 251)) / (dx)^2) + ... % Conduction in X
    k_pv * 2*((T(1, 200, 251) - T(1, 201, 251)) / (dy)^2) + ... % Conduction in Y
    k_pv * 2*((T(1, 201, 250) - T(1, 201, 251)) / (dz)^2) + ... % Conduction in Z
    Qrad / (dx/2) + ... % Radiation term on left YZ
    h * (T_inf_t - T(1, 201, 251)) * ((2/dx)+(2/dz))); % Convection

    % Update temperature at node T(1,201,1)
    T_new(1, 201, 1) = T(1, 201, 1) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * 2*((T(2, 201, 1) - T(1, 201, 1)) / (dx)^2) + ... % Conduction in X
    k_pv * 2*((T(1, 200, 1) - T(1, 201, 1)) / (dy)^2) + ... % Conduction in Y
    k_pv * 2*((T(1, 201, 2) - T(1, 201, 1)) / (dz)^2) + ... % Conduction in Z
    Qrad / (dx/2) + ... % Radiation term on left YZ
    h * (T_inf_t - T(1, 201, 1)) * ((2/dx)+(2/dz))); % Convection

    % Update temperature at node T(1,1,251)
    T_new(1, 1, 251) = T(1, 1, 251) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * 2*((T(2, 1, 251) - T(1, 1, 251)) / (dx)^2) + ... % Conduction in X
    k_pv * 2*((T(1, 2, 251) - T(1, 1, 251)) / (dy)^2) + ... % Conduction in Y
    k_pv * 2*((T(1, 1, 250) - T(1, 1, 251)) / (dz)^2) + ... % Conduction in Z
    Qrad / (dx/2) + ... % Radiation term on left YZ
    h * (T_inf_t - T(1, 1, 251)) * ((2/dx)+(2/dz))); % Convection

    % Update temperature at node T(26,201,1)
    T_new(26, 201, 1) = T(26, 201, 1) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * 2*((T(25, 201, 1) - T(26, 201, 1)) / (dx)^2) + ... % Conduction in X
    k_pv * 2*((T(26, 200, 1) - T(26, 201, 1)) / (dy)^2) + ... % Conduction in Y
    k_pv * 2*((T(26, 201, 2) - T(26, 201, 1)) / (dz)^2) + ... % Conduction in Z
    h * (T_inf_t - T(26, 201, 1)) * ((2/dx)+(2/dz))); % Convection

    % Update temperature at node T(26,201,251)
    T_new(26, 201, 251) = T(26, 201, 251) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * 2*((T(25, 201, 251) - T(26, 201, 251)) / (dx)^2) + ... % Conduction in X
    k_pv * 2*((T(26, 200, 251) - T(26, 201, 251)) / (dy)^2) + ... % Conduction in Y
    k_pv * 2*((T(26, 201, 250) - T(26, 201, 251)) / (dz)^2) + ... % Conduction in Z
    h * (T_inf_t - T(26, 201, 251)) * ((2/dx)+(2/dz))); % Convection

```

```

% Update temperature at node T(26,1,1)
T_new(26, 1, 1) = T(26, 1, 1) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * 2*((T(25, 1, 1) - T(26, 1, 1)) / (dx)^2) + ... % Conduction in X
k_pv * 2*((T(26, 2, 1) - T(26, 1, 1)) / (dy)^2) + ... % Conduction in Y
k_pv * 2*((T(26, 1, 2) - T(26, 1, 1)) / (dz)^2) + ... % Conduction in Z
h * (T_inf_t - T(26, 1, 1)) * ((2/dx)+(2/dz));% Convection

% Update temperature at node T(26,1,251)
T_new(26, 1, 251) = T(26, 1, 251) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * 2*((T(25, 1, 251) - T(26, 1, 251)) / (dx)^2) + ... % Conduction in X
k_pv * 2*((T(26, 2, 251) - T(26, 1, 251)) / (dy)^2) + ... % Conduction in Y
k_pv * 2*((T(26, 1, 250) - T(26, 1, 251)) / (dz)^2) + ... % Conduction in Z
h * (T_inf_t - T(26, 1, 251)) * ((2/dx)+(2/dz));% Convection

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loop for nodes on the edge for i=1, j=2:200, k=1
for j = 2:200
k = 1; % Z = 0 plane (k = 0)
% Discretized equation for node T(1,j,1)
T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
k_pv * ((T(1, j+1, k) - 2*T(1, j, k) + T(1, j-1, k)) / (dy)^2) + ... % Conduction in Y
k_pv * 2*((T(1, j, k+1) - T(1, j, k)) / (dz)^2) + ... % Conduction in Z
Qrad / (dx/2) + ... % Radiation term
(h * (T_inf_t - T(1, j, k)) / (dz/2))+(h*(T_inf_t-T(1,j,k))/(dx/2)); % Convection term
end

% Loop for nodes on the edge for i=1, j=201, k=2:250
for k = 2:250
j = 201;
% Discretized equation for node T(1,j,1)
T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
k_pv * 2*((- T(1, j, k) + T(1, j-1, k)) / (dy)^2) + ... % Conduction in Y
k_pv * ((T(1, j, k+1) - 2*T(1, j, k)+T(1,j,k-1)) / (dz)^2) + ... % Conduction in Z
Qrad / (dx/2) + ... % Radiation term
(h * (T_inf_t - T(1, j, k)) / (dx/2)); % Convection term
end

% Loop for nodes on the edge for i=1, j=2:200, k=251
for j = 2:200
k = 251;
% Discretized equation for node T(1,j,1)
T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
k_pv * ((T(1, j+1, k) - 2*T(1, j, k) + T(1, j-1, k)) / (dy)^2) + ... % Conduction in Y
k_pv * 2*((T(1, j, k-1) - T(1, j, k)) / (dz)^2) + ... % Conduction in Z
Qrad / (dx/2) + ... % Radiation term
(h * (T_inf_t - T(1, j, k)) / (dz/2))+(h*(T_inf_t-T(1,j,k))/(dx/2)); % Convection term
end

% Loop for nodes on the edge for i=1, j=1, k=2:250
for k = 2:250
j = 1;
% Discretized equation for node T(1,j,1)
T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
k_pv * 2*((- T(1, j, k) + T(1, j+1, k)) / (dy)^2) + ... % Conduction in Y
k_pv * ((T(1, j, k+1) - 2*T(1, j, k)+T(1,j,k-1)) / (dz)^2) + ... % Conduction in Z
Qrad / (dx/2) + ... % Radiation term
(h * (T_inf_t - T(1, j, k)) / (dx/2)); % Convection term
end

% Loop for nodes on the edge for i=26, j=201, k=2:250
for k = 2:250
j = 201;
% Discretized equation for node T(1,j,1)
T_new(26, j, k) = T(26, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * 2*((T(25, j, k) - T(26, j, k)) / (dx)^2) + ... % Conduction in +X
k_pv * 2*((- T(26, j, k) + T(26, j-1, k)) / (dy)^2) + ... % Conduction in Y
k_pv * ((T(26, j, k+1) - 2*T(26, j, k)+T(26,j,k-1)) / (dz)^2) + ... % Conduction in Z
(h * (T_inf_t - T(26, j, k)) / (dx/2)); % Convection term
end

% Loop for nodes on the edge for i=2:25, j=201, k=1
for i = 2:25
j = 201;
% Discretized equation for node T(1,j,1)
T_new(i, j, 1) = T(i, j, 1) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * ((T(i+1, j, 1) - 2*T(i, j, 1) + T(i-1,j,1)) / (dx)^2) + ... % Conduction in +X
k_pv * 2*((- T(i, j, 1) + T(i, j-1, 1)) / (dy)^2) + ... % Conduction in Y
k_pv * 2*((T(i, j, 1) - T(i, j, 2))) / (dz)^2) + ... % Conduction in Z
(h * (T_inf_t - T(i, j, 1)) / (dz/2)); % Convection term
end

% Loop for nodes on the edge for i=2:25, j=201, k=251
for i = 2:25
j = 201;
% Discretized equation for node T(1,j,1)
T_new(i, j, 251) = T(i, j, 251) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * ((T(i+1, j, 251) - 2*T(i, j, 251) + T(i-1,j,251)) / (dx)^2) + ... % Conduction in +X
k_pv * 2*((- T(i, j, 251) + T(i, j-1, 251)) / (dy)^2) + ... % Conduction in Y
k_pv * 2*((T(i, j, 251) - T(i, j, 250))) / (dz)^2) + ... % Conduction in Z
(h * (T_inf_t - T(i, j, 251)) / (dz/2)); % Convection term
end

% Loop for nodes on the edge for i=26, j=2:200, k=1
for j = 2:200
k = 1; % Z = 0 plane (k = 0)
% Discretized equation for node T(1,j,1)
T_new(26, j, k) = T(26, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
k_pv * 2*((T(25, j, k) - T(26, j, k)) / (dx)^2) + ... % Conduction in +X
k_pv * ((T(26, j+1, k) - 2*T(26, j, k) + T(26, j-1, k)) / (dy)^2) + ... % Conduction in Y
k_pv * 2*((T(26, j, k+1) - T(26, j, k)) / (dz)^2) + ... % Conduction in Z
(h * (T_inf_t - T(26, j, k)) / (dz/2))+(h*(T_inf_t-T(26,j,k))/(dx/2)); % Convection term
end

```

```

% Loop for nodes on the edge for i=26, j=2:200, k=251
for j = 2:200
    k = 251; % Z = 0 plane (k = 0)
    % Discretized equation for node T(1,j,1)
    T_new(26, j, k) = T_new(26, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * 2*((T(25, j, k) - T(26, j, k)) / (dx/2)^2) + ... % Conduction in +X
        k_pv * ((T(26, j+1, k) - 2*T(26, j, k) + T(26, j-1, k)) / (dy)^2) + ... % Conduction in Y
        k_pv * 2*((T(26, j, k-1) - T(26, j, k)) / (dz)^2) + ... % Conduction in Z
        (h * (T_inf_t - T(26, j, k)) / (dz/2)) + (h*(T_inf_t-T(26,j,k))/(dx/2))); % Convection term
end

% Loop for nodes on the edge for j = 1 k=1
for i = 2:25
    k=1;
    T_new(i, 1, k) = T(i, 1, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * ((T(i+1, 1, k) - 2*T(i, 1, k)+T(i-1,j,k)) / (dx)^2) + ... % Conduction in X
        k_pv * 2*((T(i,2,k)-T(i,1,k))/(dy)^2)+... % Conduction in Y
        k_pv * 2*((T(i, 1, k+1) - T(i, 1, k)) / (dz)^2) + ... % Conduction in Z
        h * (T_inf_t - T(i, 1, k)) / (dz/2)); % Convection at the front
end

% Loop for nodes on the edge for j = 1 k=251
for i = 2:25
    k=251;
    T_new(i, 1, k) = T(i, 1, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * ((T(i+1, 1, k) - 2*T(i, 1, k)+T(i-1,j,k)) / (dx)^2) + ... % Conduction in X
        k_pv * 2*((T(i,2,k)-T(i,1,k))/(dy)^2)+... % Conduction in Y
        k_pv * 2*((T(i, 1, k-1) - T(i, 1, k)) / (dz)^2) + ... % Conduction in Z
        h * (T_inf_t - T(i, 1, k)) / (dz/2)); % Convection at the front
end

% Loop for nodes on the edge for i=26, j=1, k=2:250
for k = 2:250
    j = 1;
    % Discretized equation for node T(1,j,1)
    T_new(26, j, k) = T(26, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * 2*((T(25, j, k) - T(26, j, k)) / (dx)^2) + ... % Conduction in +X
        k_pv * 2*((T(26, j, k) - T(26, j+1, k)) / (dy)^2) + ... % Conduction in Y
        k_pv * ((T(26, j, k+1) - 2*T(26, j, k)+T(26,j,k-1)) / (dz)^2) + ... % Conduction in Z
        (h * (T_inf_t - T(26, j, k)) / (dx/2))); % Convection term
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Loop over XZ plane for j = 1 k=2:250 i=2:25
for i = 1:26
    for k = 1:251 % Loop over Z direction (top to bottom)
        T_new(i, 1, k) = T(i, 2, k);
    end
end
% Loop over XZ plane for j=201 k=2:250 i=2:25
for i = 1:26
    for k = 1:251 % Loop over Z direction (top to bottom)
        T_new(i, 201, k) = T(i, 200, k);
    end
end

% Loop for nodes on the YZ plane for i=2:25, j=2:200, k=1
for j = 2:200
    for i=2:25
        % Discretized equation for node T(1,j,1)
        T_new(i, j, 1) = T(i, j, 1) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * ((T(i+1, j, 1) - 2*T(i, j, 1)+T(i-1,j,1)) / (dx)^2) + ... % Conduction in +X
            k_pv * ((T(i,j-1,1) - 2*T(i, j, 1) + T(i, j+1, 1)) / (dy)^2) + ... % Conduction in Y
            k_pv * 2*((T(i, j, 2) - T(i, j, 1)) / (dz)^2) + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, 1)) / (dz/2))); % Convection term
    end
end

% Loop for nodes on the YZ plane for i=2:25, j=2:200, k=251
for j = 2:200
    for i=2:25
        % Discretized equation for node T(1,j,1)
        T_new(i, j, 251) = T(i, j, 251) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * ((T(i+1, j, 251) - 2*T(i, j, 251)+T(i-1,j,251)) / (dx)^2) + ... % Conduction in +X
            k_pv * ((T(i,j-1,251) - 2*T(i, j, 251) + T(i, j+1, 251)) / (dy)^2) + ... % Conduction in Y
            k_pv * 2*((T(i, j, 250) - T(i, j, 251)) / (dz)^2) + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, 251)) / (dz/2))); % Convection term
    end
end

Loop for nodes on the YZ plane for i=1, j=2:200, k=2:250
for j = 2:200
    for k=2:250
        % Discretized equation for node T(1,j,1)
        T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
            k_pv * ((T(1,j-1,k) - 2*T(1, j, k) + T(1, j+1, k)) / (dy)^2) + ... % Conduction in Y
            k_pv * ((T(1,j,k+1) - 2*T(1, j, k) + T(1, j, k-1)) / (dz)^2) + ... % Conduction in Z
            Qrad / (dx/2) + ... % Radiation term
            (h * (T_inf_t - T(1, j, k)) / (dx/2))); % Convection term
    end
end

% Loop for nodes on the YZ plane for i=26, j=2:200, k=2:250
for j = 2:200
    for k=2:250
        % Discretized equation for node T(1,j,1)
        T_new(26, j, k) = T(26, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * 2*((T(25, j, k) - T(26, j, k)) / (dx)^2) + ... % Conduction in +X
            k_pv * ((T(26,j-1,k) - 2*T(26, j, k) + T(26, j+1, k)) / (dy)^2) + ... % Conduction in Y
            k_pv * ((T(26,j,k+1) - 2*T(26, j, k) + T(26, j, k-1)) / (dz)^2) + ... % Conduction in Z
            (h * (T_inf_t - T(26, j, k)) / (dx/2))); % Convection term
    end
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 2:(nx-1)
    for j = 2:(ny-1)
        for k = 2:(nz-1)
            T_new(i, j, k) = T(i, j, k) + (dt / (rho_pv * cp_pv)) * ...
                (k_pv * ((T(i+1, j, k) - 2*T(i, j, k) + T(i-1, j, k)) / dx^2 + ...
                    (T(i, j+1, k) - 2*T(i, j, k) + T(i, j-1, k)) / dy^2 + ...
                    (T(i, j, k+1) - 2*T(i, j, k) + T(i, j, k-1)) / dz^2));
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Update for next iteration
T = T_new;
if time(t_idx) == 10
    fprintf("At t = 10s, the Temperature at (17,8,9) is %g\n", T(17,8,9));
    break; %
end
fprintf('Iteration %d/%d completed\n', t_idx, length(time));
fprintf("Temperature at (1,1,1): %g\n", T(1,1,1));
fprintf("The Temperature at (7,8,9) is %g\n", T(7,8,9));
fprintf("The Temperature at (17,8,9) is %g\n", T(17,8,9));
fprintf("The Temperature at (1,201,1) is %g\n", T(1,201,1));

% Visualization Section
if mod(t_idx, plot_interval) == 0 || t_idx == length(time)
    % Define grid points
    x = linspace(0, W, nx);
    y = linspace(0, L, ny);
    z = linspace(0, 0.25, nz);

    % Create or clear figure
    figure(1); % Use the same figure
    clf; % Clear the figure for the next frame
    colormap jet;
    hold on; % Hold on to plot multiple surfaces

    % Front Face (y = y(1))
    [Xf, Zf] = meshgrid(x, z);
    Yf = y(1) * ones(size(Xf));
    T_front = squeeze(T(:, 1, :));
    surf(Xf, Yf, Zf, T_front, 'EdgeColor', 'none');

    % Back Face (y = y(end))
    [Xb, Zb] = meshgrid(x, z);
    Yb = y(end) * ones(size(Xb));
    T_back = squeeze(T(:, end, :));
    surf(Xb, Yb, Zb, T_back, 'EdgeColor', 'none');

    % Left Face (x = x(1))
    [Yl, Zl] = meshgrid(y, z);
    Xl = x(1) * ones(size(Yl));
    T_left = squeeze(T(1, :, :));
    surf(Xl, Yl, Zl, T_left, 'EdgeColor', 'none');

    % Right Face (x = x(end))
    [Yr, Zr] = meshgrid(y, z);
    Xr = x(end) * ones(size(Yr));
    T_right = squeeze(T(end, :, :));
    surf(Xr, Yr, Zr, T_right, 'EdgeColor', 'none');

    % Bottom Face (z = z(1))
    [Xbott, Ybott] = meshgrid(x, y);
    Zbott = z(1) * ones(size(Xbott));
    T_bottom = squeeze(T(:, :, 1));
    surf(Xbott, Ybott, Zbott, T_bottom, 'EdgeColor', 'none');

    % Top Face (z = z(end))
    [Xtop, Ytop] = meshgrid(x, y);
    Ztop = z(end) * ones(size(Xtop));
    T_top = squeeze(T(:, :, end));
    surf(Xtop, Ytop, Ztop, T_top, 'EdgeColor', 'none');

    % Customize plot
    hold off;
    colorbar;
    xlabel('X-axis (m)');
    ylabel('Y-axis (m)');
    zlabel('Z-axis (m)');
    title(sprintf('Temperature Distribution at time %.2f s', time(t_idx)));
    view(3); % Set to 3D view
    axis equal; % Equal scaling for all axes

    frame = getframe(gcf);
    writeVideo(videoWriter, frame);
    % Pause for animation effect
    pause(0.1);
end
end

%
close(videoWriter);
disp('Finish! ');

```

Appendix 4 finned transient:

```

clear;
clc;
% Define physical constants and panel dimensions
L = 0.200; % Length of the panel (m, Y-axis)
W = 0.325; % Width of the panel (m, X-axis)
Z = 0.250; % Height of the panel and fin (m, Z-axis)
nx = 326; % Number of grid points in x
ny = 201; % Number of grid points in y
nz = 251; % Number of grid points in z
dx = W / (nx - 1);
dy = L / (ny - 1);
dz = Z / (nz - 1);
% Material properties for panel
k_pv = 1; % Thermal conductivity (W/m.K)
rho_pv = 3000; % Density (kg/m^3)
cp_pv = 500; % Specific heat capacity (J/kg.K)
% Material properties for fin
k_fin = 10; % Thermal conductivity (W/m.K)
rho_fin = 3000; % Density (kg/m^3)
cp_fin = 500; % Specific heat capacity (J/kg.K)
% Time-stepping parameters
dt = 0.005; % Time step (s)
t_start = 0; % Start time: 9 AM (seconds)
t_end = 200; %* 3600; % End time: 5 PM (seconds)
time = t_start:dt:t_end;
% Heat transfer parameters
h = 80; % Convective heat transfer coefficient (W/m^2.K)
Qrad = 900; % Radiative heat input (W/m^2)
% Initial and boundary conditions
T = 298 * ones(nx, ny, nz); % Initial temperature (K)
Tinf = 293; % Ambient temperature function
% Fin and panel node ranges
x_panel = 1:26; % X: panel
y_panel = 1:201; % Y: panel
z_panel = 1:251; % Z: panel
x_fin = 27:326; % X: fin
y_fin = 1:3; % Y: fin
z_fin = 1:251; % Z: fin
max_iter=10000;
u=((nx-1)*25/(1000*W))+1;
t_fin=2;%in mm
v=t_fin*(ny-1)/(1000*L)+1;

% Critical Time-Step
dt_panel = rho_pv * cp_pv * (min([dx, dy, dz])^2) / (2 * k_pv);
dt_fin = rho_fin * cp_fin * (min([dx, dy, dz])^2) / (2 * k_fin);

% Time-stepping loop
plot_interval = 10;

% videoFilename = 'TemperatureDistributionWithFin4.avi';
% videoWriter = VideoWriter(videoFilename, 'Motion JPEG AVI');
% videoWriter.Quality = 100;
% videoWriter.FrameRate = 30;
% open(videoWriter);

for t_idx = 1:length(time)

    T_new = T;
    T_inf_t = 293;

    % Update temperature at node T(1,1,1)
    T_new(1, 1, 1) = T(1, 1, 1) + (dt / (rho_pv * cp_pv)) * ( ...
    (k_pv * ((T(2, 1, 1) - T(1, 1, 1)) / (dx)^2 * 2)) + ... % Conduction in X
    (k_pv * ((T(1, 2, 1) - T(1, 1, 1)) / (dy)^2 * 2)) + ... % Conduction in Y
    (k_pv * ((T(1, 1, 2) - T(1, 1, 1)) / (dz)^2 * 2)) + ... % Conduction in Z
    (Qrad / (dx/2)) + ... % Radiation term on left YZ
    (h * (T_inf_t - T(1, 1, 1)) * ((2/dx)+(2/dz)))); % Convection

    fprintf('The Temperature is %g\n',T(1,1,1))
    % Update temperature at node T(1,201,1)
    T_new(1, ny, 1) = T(1, ny, 1) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * ((T(2, ny, 1) - T(1, ny, 1)) / (dx)^2 * 2) + ... % Conduction in X
    k_pv * ((T(1, ny-1, 1) - T(1, ny, 1)) / (dy)^2 * 2) + ... % Conduction in Y
    k_pv * ((T(1, ny, 2) - T(1, ny, 1)) / (dz)^2 * 2) + ... % Conduction in Z
    Qrad / (dx/2) + ... % Radiation term on left YZ
    h * (T_inf_t - T(1, ny, 1)) * ((2/dx)+(2/dz)))); % Convection
    % fprintf('The Temperature is %g\n',T(1,201,1))

    % Update temperature at node T(1,201,251)
    T_new(1, ny, nz) = T(1, ny, nz) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * ((T(2, ny, nz) - T(1, ny, nz)) / (dx)^2 * 2) + ... % Conduction in X
    k_pv * ((T(1, ny-1, nz) - T(1, ny, nz)) / (dy)^2 * 2) + ... % Conduction in Y
    k_pv * ((T(1, ny, nz-1) - T(1, ny, nz)) / (dz)^2 * 2) + ... % Conduction in Z
    Qrad / (dx/2) + ... % Radiation term on left YZ
    h * (T_inf_t - T(1, ny, nz)) * ((2/dx)+(2/dz)))); % Convection

    % Update temperature at node T(1,1,251)
    T_new(1, 1, nz) = T(1, 1, nz) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * ((T(2, 1, nz) - T(1, 1, nz)) / (dx)^2 * 2) + ... % Conduction in X
    k_pv * ((T(1, 2, nz) - T(1, 1, nz)) / (dy)^2 * 2) + ... % Conduction in Y
    k_pv * ((T(1, 1, nz-1) - T(1, 1, nz)) / (dz)^2 * 2) + ... % Conduction in Z
    Qrad / (dx/2) + ... % Radiation term on left YZ
    h * (T_inf_t - T(1, 1, nz)) * ((2/dx)+(2/dz)))); % Convection

    % Update temperature at node T(26,201,1)
    T_new(u, ny, 1) = T(u, ny, 1) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * ((T(u-1, ny, 1) - T(u, ny, 1)) / (dx)^2 * 2) + ... % Conduction in X
    k_pv * ((T(u, ny-1, 1) - T(u, ny, 1)) / (dy)^2 * 2) + ... % Conduction in Y
    k_pv * ((T(u, ny, 2) - T(u, ny, 1)) / (dz)^2 * 2) + ... % Conduction in Z
    h * (T_inf_t - T(u, ny, 1)) * ((2/dx)+(2/dz)))); % Convection

    % Update temperature at node T(26,201,251)
    T_new(u, ny, nz) = T(u, ny, nz) + (dt / (rho_pv * cp_pv)) * ( ...
    k_pv * ((T(u-1, ny, nz) - T(u, ny, nz)) / (dx)^2 * 2) + ... % Conduction in X
    k_pv * ((T(u, ny-1, nz) - T(u, ny, nz)) / (dy)^2 * 2) + ... % Conduction in Y
    k_pv * ((T(u, ny, nz-1) - T(u, ny, nz)) / (dz)^2 * 2) + ... % Conduction in Z
    h * (T_inf_t - T(u, ny, nz)) * ((2/dx)+(2/dz)))); % Convection

```

```

% Update temperature at node T(326,1,1)
T_new(nx, 1, 1) = T(nx, 1, 1) + (dt / (rho_fin * cp_fin)) * ( ...
    k_fin * ((T(nx-1, 1, 1) - T(nx, 1, 1)) / (dx)^2) + ... % Conduction in X
    k_fin * ((T(nx, 2, 1) - T(nx, 1, 1)) / (dy)^2) + ... % Conduction in Y
    k_fin * ((T(nx, 1, 2) - T(nx, 1, 1)) / (dz)^2) + ... % Conduction in Z
    h * (T_inf_t - T(nx, 1, 1)) * ((2/dx)+(2/dz))); % Convection

% Update temperature at node T(326,3,1)
T_new(nx,v, 1) = T(nx, v, 1) + (dt / (rho_fin * cp_fin)) * ( ...
    k_fin * ((T(nx-1, v, 1) - T(nx, v, 1)) / (dx)^2) + ... % Conduction in X
    k_fin * ((T(nx, v-1, 1) - T(nx,v, 1)) / (dy)^2) + ... % Conduction in Y
    k_fin * ((T(nx,v, 2) - T(nx,v, 1)) / (dz)^2) + ... % Conduction in Z
    h * (T_inf_t - T(nx,v,1)) * ((2/dx)+(2/dy)+(2/dz))); % Convection

% Update temperature at node T(326,3,251)
T_new(nx,v, nz) = T(nx, v,nz) + (dt / (rho_fin * cp_fin)) * ( ...
    k_fin * ((T(nx-1, v, nz) - T(nx, v, nz)) / (dx)^2) + ... % Conduction in X
    k_fin * ((T(nx, v-1, nz) - T(nx,v, nz)) / (dy)^2) + ... % Conduction in Y
    k_fin * ((T(nx,v, nz-1) - T(nx,v, nz)) / (dz)^2) + ... % Conduction in Z
    h * (T_inf_t - T(nx,v,nz)) * ((2/dx)+(2/dy)+(2/dz))); % Convection

% Update temperature at node T(326,2,1)
T_new(nx,2, 1) = T(nx, 2,1) + (dt / (rho_fin * cp_fin)) * ( ...
    k_fin * ((T(nx-1, 2, 1) - T(nx, 2, 1)) / (dx)^2) + ... % Conduction in X
    k_fin * ((T(nx, 1, 1)+T(nx, 3, 1) - 2*T(nx,2, 1)) / (dy)^2) + ... % Conduction in Y
    k_fin * ((T(nx,2, 2) - T(nx,2, 1)) / (dz)^2) + ... % Conduction in Z
    h * (T_inf_t - T(nx,2,1)) * ((2/dx)+(2/dz))); % Convection

% Update temperature at node T(326,2,251)
T_new(nx,2, nz) = T(nx, 2,nz) + (dt / (rho_fin * cp_fin)) * ( ...
    k_fin * ((T(nx-1, 2, nz) - T(nx, 2, nz)) / (dx)^2) + ... % Conduction in X
    k_fin * ((T(nx, 1, nz)+T(nx, 3, nz) - 2*T(nx,2, nz)) / (dy)^2) + ... % Conduction in Y
    k_fin * ((T(nx,2, nz-1) - T(nx,2, nz)) / (dz)^2) + ... % Conduction in Z
    h * (T_inf_t - T(nx,2,nz)) * ((2/dx)+(2/dz))); % Convection

% Update temperature at node T(nx,1,251)
T_new(nx, 1, nz) = T(nx, 1, nz) + (dt / (rho_fin * cp_fin)) * ( ...
    k_fin * ((T(nx-1, 1, nz) - T(nx, 1, nz)) / (dx)^2) + ... % Conduction in X
    k_fin * ((T(nx, 2, nz) - T(nx, 1, nz)) / (dy)^2) + ... % Conduction in Y
    k_fin * ((T(nx, 1, nz-1) - T(nx, 1, nz)) / (dz)^2) + ... % Conduction in Z
    h * (T_inf_t - T(nx, 1, nz)) * ((2/dx)+(2/dz))); % Convection

% Loop for nodes for i=26, j=2, k=1
for i=u
    for j=2:v-1
        for k=1
            T_new(i,j,k) = T(i,j,k) + dt/((rho_fin*cp_fin)+(rho_pv*cp_pv))*( ...
                (k_fin*(T(i+1,j,k)-T(i,j,k))) + (k_pv*(T(i-1,j,k)-T(i,j,k))))*(2/dx^2) + ...
                ((k_fin+k_pv)*(T(i,j+1,k)-2*T(i,j,k)+T(i,j-1,k)))/(dy^2) + ...
                ((k_fin+k_pv)*(T(i,j,k+1)-T(i,j,k)))/(2/dz^2)+...
                (h*(T_inf_t - T(i,j,k))*(4/dz)));
        end
    end
end

% Loop for nodes for i=26, j=2, k=251
for i=u
    for j=2:v-1
        for k=nz
            T_new(i,j,k) = T(i,j,k) + dt/((rho_fin*cp_fin)+(rho_pv*cp_pv))*( ...
                (k_fin*(T(i+1,j,k)-T(i,j,k))) + (k_pv*(T(i-1,j,k)-T(i,j,k))))*(2/dx^2) + ...
                ((k_fin+k_pv)*(T(i,j+1,k)-2*T(i,j,k)+T(i,j-1,k)))/(dy^2) + ...
                ((k_fin+k_pv)*(T(i,j,k-1)-T(i,j,k)))/(2/dz^2)+...
                (h*(T_inf_t - T(i,j,k))*(4/dz)));
        end
    end
end

% Loop for nodes for i=26, j=1, k=1
for i=u
    for j=1
        for k=1
            T_new(i,j,k) = T(i,j,k) + dt/((rho_fin*cp_fin)+(rho_pv*cp_pv))*( ...
                (k_fin*(T(i+1,j,k)-T(i,j,k))) + (k_pv*(T(i-1,j,k)-T(i,j,k))))*(2/dx^2) + ...
                ((k_fin+k_pv)*(T(i,j+1,k)-T(i,j,k)))/(dy^2/2) + ...
                ((k_fin+k_pv)*(T(i,j,k-1)-T(i,j,k)))/(2/dz^2) + ...
                (h*(T_inf_t - T(i,j,k))*(4/dz)));
        end
    end
end

% Loop for nodes for i=26, j=1, k=251
for i=u
    for j=1
        for k=nz
            T_new(i,j,k) = T(i,j,k) + dt/((rho_fin*cp_fin)+(rho_pv*cp_pv))*( ...
                (k_fin*(T(i+1,j,k)-T(i,j,k))) + (k_pv*(T(i-1,j,k)-T(i,j,k))))*(2/dx^2) + ...
                ((k_fin+k_pv)*(T(i,j+1,k)-T(i,j,k)))/(dy^2/2) + ...
                ((k_fin+k_pv)*(T(i,j,k-1)-T(i,j,k)))/(2/dz^2) + ...
                (h*(T_inf_t - T(i,j,k))*(4/dz)));
        end
    end
end
end

```

```

% Loop for nodes for i=26, j=3, k=1
for i=u
    for j=v
        for k=1
            T_new(i,j,k)= T(i,j,k) +dt/((rho_fin*cp_fin/2)+(rho_pv*cp_pv))*( ...
                (k_fin*(T(i+1,j,k)-(T(i,j,k))) + (2*k_pv*(T(i-1,j,k)-T(i,j,k))))*(1/dx^2) +...
                ((k_fin+k_pv)*(T(i,j-1,k)-T(i,j,k))/(dy^2) + ((k_pv)*(T(i,j+1,k)-T(i,j,k))/(dy^2) +...
                ((k_fin+2*k_pv)*(T(i,j,k+1)-T(i,j,k))*(1/dz^2) +...
                (h*(T_inf_t - T(i,j,k))*(3/dz + 1/dx + 1/dy)));
        end
    end
end
% Loop for nodes for i=26, j=3, k=251
for i=u
    for j=v
        for k=nz
            T_new(i,j,k)= T(i,j,k) +dt/((rho_fin*cp_fin/2)+(rho_pv*cp_pv))*( ...
                (k_fin*(T(i+1,j,k)-(T(i,j,k))) + (2*k_pv*(T(i-1,j,k)-T(i,j,k))))*(1/dx^2) +...
                ((k_fin+k_pv)*(T(i,j-1,k)-T(i,j,k))/(dy^2) + ((k_pv)*(T(i,j+1,k)-T(i,j,k))/(dy^2) +...
                ((k_fin+2*k_pv)*(T(i,j,k-1)-T(i,j,k))*(1/dz^2) +...
                (h*(T_inf_t - T(i,j,k))*(3/dz + 1/dx + 1/dy)));
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% edge
% Loop for nodes for i=26, j=1, k=2:250
for i=u
    for j=1
        for k=2:nz-1
            T_new(i,j,k)= T(i,j,k) +dt/((rho_fin*cp_fin)+(rho_pv*cp_pv))*( ...
                (k_fin*(T(i+1,j,k)-(T(i,j,k))) + (k_pv*(T(i-1,j,k)-T(i,j,k))))*(2/dx^2) +...
                ((k_fin+k_pv)*(T(i,j+1,k)-T(i,j,k))/(dy^2/2) +...
                ((k_fin+k_pv)*(T(i,j,k-1)-2*T(i,j,k)+T(i,j,k+1)))/(dz^2));
        end
    end
end
% Loop for nodes for i=26, j=3, k=2:250
for i=u
    for j=v
        for k=2:nz-1
            T_new(i,j,k)= T(i,j,k) +dt/((rho_fin*cp_fin/2)+(rho_pv*cp_pv))*( ...
                (k_fin*(T(i+1,j,k)-(T(i,j,k))) + (2*k_pv*(T(i-1,j,k)-T(i,j,k))))*(1/dx^2) +...
                ((k_fin+k_pv)*(T(i,j-1,k)-T(i,j,k))/(dy^2) + ((k_pv)*(T(i,j+1,k)-T(i,j,k))/(dy^2) +...
                ((k_fin/2)+k_pv)*(T(i,j,k-1)-2*T(i,j,k)+T(i,j,k+1))*(1/dz^2) +...
                (h*(T_inf_t - T(i,j,k))*( 1/dx + 1/dy)));
        end
    end
end

% Loop for nodes on the YZ plane for i=nx, j=1, k=2:250
for k = 2:nz-1
    j = 1;
    % Discretized equation for node T(1,j,1)
    T_new(nx, j, k) = T(nx, j, k) + (dt / (rho_fin * cp_fin)) * ( ...
        k_fin * ((T(nx-1, j, k) - T(nx, j, k)) / (dx)^2) + ... % Conduction in +X
        k_fin * ( - T(nx, j, k) + T(nx, j+1, k)) / (dy)^2 + ... % Conduction in Y
        k_fin * ((T(nx, j, k+1) - 2*T(nx, j, k)+T(nx,j,k-1)))/( dz)^2 + ... % Conduction in Z
        (h * (T_inf_t - T(nx, j, k)) / (dx/2))); % Convection term
end
% Loop for nodes on the YZ plane for i=326, j=3, k=2:250
for k = 2:nz-1
    j = v;
    % Discretized equation for node T(1,j,1)
    T_new(nx, j, k) = T(nx, j, k) + (dt / (rho_fin * cp_fin)) * ( ...
        k_fin * ((T(nx-1, j, k) - T(nx, j, k)) / (dx)^2) + ... % Conduction in +X
        k_fin * ( - T(nx, j, k) + T(nx, j-1, k)) / (dy)^2 + ... % Conduction in Y
        k_fin * ((T(nx, j, k+1) - 2*T(nx, j, k)+T(nx,j,k-1)))/( dz)^2 + ... % Conduction in Z
        (h * (T_inf_t - T(nx, j, k)) * (2/dx+2/dy))); % Convection term
end
% Loop for nodes on the YZ plane for i=326, j=2, k=2:250
for k = 2:nz-1
    for j = 2:v-1
        % Discretized equation for node T(1,j,1)
        T_new(nx, j, k) = T(nx, j, k) + (dt / (rho_fin * cp_fin)) * ( ...
            k_pv * ((T(i+1, j, k) - 2*T(i, j, k)) / (dx)^2) + ... % Conduction in +X
            k_fin * ( - T(nx, j-1, k) - 2*T(nx, j, k) + T(nx, j+1, k)) / (dy)^2 + ... % Conduction in Y
            k_fin * ((T(nx, j, k+1) - 2*T(nx, j, k)+T(nx,j,k-1)))/( dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(nx, j, k)) / (dx/2))); % Convection term
        end
    end
end
% Loop for nodes on the YZ plane for i=27:325, j=3, k=251
for i = u+1:nx-1
    for j = v;
        % Discretized equation for node T(1,j,1)
        T_new(i, j, nz) = T(i, j, nz) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * ((T(i+1, j, nz) - 2*T(i, j, nz) + T(i-1, j, nz)) / (dx)^2) + ... % Conduction in +X
            k_pv * ( - T(i, j, nz) + T(i, j-1, nz)) / (dy)^2 + ... % Conduction in Y
            k_pv * ((T(i, j, nz-1) - T(i, j, nz)))/( dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, nz)) * (2/dz + 2/dy))); % Convection term
        end
    end
% Loop for nodes on the YZ plane for i=27:325, j=3, k=1
for i = u+1:nx-1
    for j = v;
        % Discretized equation for node T(1,j,1)
        T_new(i, j, 1) = T(i, j, 1) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * ((T(i+1, j, 1) - 2*T(i, j, 1) + T(i-1, j, 1)) / (dx)^2) + ... % Conduction in +X
            k_pv * ( - T(i, j, 1) + T(i, j-1, 1)) / (dy)^2 + ... % Conduction in Y
            k_pv * ((T(i, j, 2) - T(i, j, 1)))/( dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, 1)) * (2/dz + 2/dy))); % Convection term
        end
    end
end

```

```

-----
% Loop for nodes on the YZ plane for i=27:325, j=3, k=1
for i = u+1:nx-1
    j = v;
    % Discretized equation for node T(1,j,1)
    T_new(i, j, 1) = T(i, j, 1) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * ((T(i+1, j, 1) - 2*T(i, j, 1) + T(i-1, j, 1))) / (dx)^2) + ... % Conduction in +X
        k_pv * (- (T(i, j, 1) + T(i, j-1, 1))) / (dy)^2 + ... % Conduction in Y
        k_pv * ((T(i, j, 2) - T(i, j, 1))) / (dz)^2 + ... % Conduction in Z
        (h * (T_inf_t - T(i, j, 1)) * (2/dz + 2/dy))); % Convection term
    end
% Loop for nodes on the YZ plane for i=27:325, j=2, k=1
for j = 2:v-1
    for i=u+1:nx-1
        % Discretized equation for node T(1,j,1)
        T_new(i, j, 1) = T(i, j, 1) + (dt / (rho_fin * cp_fin)) * ( ...
            k_fin * ((T(i+1, j, 1) - 2*T(i, j, 1) + T(i-1, j, 1))) / (dx)^2) + ... % Conduction in +X
            k_fin * (T(i, j-1, 1) - 2*T(i, j, 1) + T(i, j+1, 1))) / (dy)^2 + ... % Conduction in Y
            k_fin * ((T(i, j, 2) - T(i, j, 1))) / (dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, 1)) / (dz/2))); % Convection term
        end
    end
% Loop for nodes on the YZ plane for i=27:325, j=2, k=251
for j = 2:v-1
    for i=u+1:nx-1
        % Discretized equation for node T(1,j,1)
        T_new(i, j, 251) = T(i, j, 251) + (dt / (rho_fin * cp_fin)) * ( ...
            k_fin * ((T(i+1, j, 251) - 2*T(i, j, 251) + T(i-1, j, 251))) / (dx)^2) + ... % Conduction in +X
            k_fin * (T(i, j-1, 251) - 2*T(i, j, 251) + T(i, j+1, 251))) / (dy)^2 + ... % Conduction in Y
            k_fin * ((T(i, j, 250) - T(i, j, 251))) / (dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, 251)) / (dz/2))); % Convection term
        end
    end
% Loop for nodes on the YZ plane for i=27:325, j=1, k=251
for j = 1
    for i=u+1:nx-1
        % Discretized equation for node T(1,j,1)
        T_new(i, j, 251) = T(i, j, 251) + (dt / (rho_fin * cp_fin)) * ( ...
            k_fin * ((T(i+1, j, 251) - 2*T(i, j, 251) + T(i-1, j, 251))) / (dx)^2) + ... % Conduction in +X
            k_fin * (- (T(i, j, 251) + T(i, j+1, 251))) / (dy)^2 + ... % Conduction in Y
            k_fin * ((T(i, j, 250) - T(i, j, 251))) / (dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, 251)) / (dz/2))); % Convection term
        end
    end
% Loop for nodes on the YZ plane for i=27:325, j=1, k=1
for j = 1
    for i=u+1:nx-1
        % Discretized equation for node T(1,j,1)
        T_new(i, j, 1) = T(i, j, 1) + (dt / (rho_fin * cp_fin)) * ( ...
            k_fin * ((T(i+1, j, 1) - 2*T(i, j, 1) + T(i-1, j, 1))) / (dx)^2) + ... % Conduction in +X
            k_fin * (- (T(i, j, 1) + T(i, j+1, 1))) / (dy)^2 + ... % Conduction in Y
            k_fin * ((T(i, j, 2) - T(i, j, 1))) / (dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, 1)) / (dz/2))); % Convection term
        end
    end

% Loop for nodes on the edge for i=1, j=2:200, k=1
for j = 2:200
    k = 1; % Z = 0 plane (k = 0)
    % Discretized equation for node T(1,j,1)
    T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
        k_pv * ((T(1, j+1, k) - 2*T(1, j, k) + T(1, j-1, k))) / (dy)^2 + ... % Conduction in Y
        k_pv * 2*((T(1, j, k+1) - T(1, j, k)) / (dz)^2) + ... % Conduction in Z
        Qrad / (dx/2) + ... % Radiation term
        (h * (T_inf_t - T(1, j, k)) / (dz/2)) + (h*(T_inf_t - T(1, j, k))/(dx/2))); % Convection term
    end
% Loop for nodes on the edge for i=1, j=201, k=2:250
for k = 2:250
    j = 201;
    % Discretized equation for node T(1,j,1)
    T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
        k_pv * ((T(1, j+1, k) - 2*T(1, j, k) + T(1, j-1, k))) / (dy)^2 + ... % Conduction in Y
        k_pv * ((T(1, j, k+1) - 2*T(1, j, k) + T(1, j, k-1))) / (dz)^2 + ... % Conduction in Z
        Qrad / (dx/2) + ... % Radiation term
        (h * (T_inf_t - T(1, j, k)) / (dx/2))); % Convection term
    end
% Loop for nodes on the edge for i=1, j=2:200, k=251
for j = 2:200
    k = 251;
    % Discretized equation for node T(1,j,1)
    T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
        k_pv * ((T(1, j+1, k) - 2*T(1, j, k) + T(1, j-1, k))) / (dy)^2 + ... % Conduction in Y
        k_pv * 2*((T(1, j, k+1) - T(1, j, k)) / (dz)^2) + ... % Conduction in Z
        Qrad / (dx/2) + ... % Radiation term
        (h * (T_inf_t - T(1, j, k)) / (dz/2)) + (h*(T_inf_t - T(1, j, k))/(dx/2))); % Convection term
    end
% Loop for nodes on the edge for i=1, j=1, k=2:250
for k = 2:250
    j = 1;
    % Discretized equation for node T(1,j,1)
    T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
        k_pv * 2*((- (T(1, j, k) + T(1, j+1, k))) / (dy)^2) + ... % Conduction in Y
        k_pv * ((T(1, j, k+1) - 2*T(1, j, k) + T(1, j, k-1))) / (dz)^2 + ... % Conduction in Z
        Qrad / (dx/2) + ... % Radiation term
        (h * (T_inf_t - T(1, j, k)) / (dx/2))); % Convection term
    end
% Loop for nodes on the edge for i=26, j=201, k=2:250
for k = 2:250
    j = 201;
    % Discretized equation for node T(1,j,1)
    T_new(26, j, k) = T(26, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * 2*((T(25, j, k) - T(26, j, k)) / (dx)^2) + ... % Conduction in +X
        k_pv * 2*((- (T(26, j, k) + T(26, j-1, k))) / (dy)^2) + ... % Conduction in Y
        k_pv * ((T(26, j, k+1) - 2*T(26, j, k) + T(26, j, k-1))) / (dz)^2 + ... % Conduction in Z
        (h * (T_inf_t - T(26, j, k)) / (dx/2))); % Convection term
    end

```

```

% Loop for nodes on the edge for i=2:25, j=201, k=1
for i = 2:25
    j = 201;
    % Discretized equation for node T(i,j,1)
    T_new(i, j, 1) = T(i, j, 1) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * ((T(i+1, j, 1) - 2*T(i, j, 1) + T(i-1, j, 1)) / (dx)^2) + ... % Conduction in +X
        k_pv * 2 * (-T(i, j, 1) + T(i, j-1, 1)) / (dy)^2 + ... % Conduction in Y
        k_pv * 2 * ((T(i, j, 1) - T(i, j, 2))) / (dz)^2 + ... % Conduction in Z
        (h * (T_inf_t - T(i, j, 1)) / (dz/2))); % Convection term
end

% Loop for nodes on the edge for i=2:25, j=201, k=251
for i = 2:25
    j = 201;
    % Discretized equation for node T(i,j,251)
    T_new(i, j, 251) = T(i, j, 251) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * ((T(i+1, j, 251) - 2*T(i, j, 251) + T(i-1, j, 251)) / (dx)^2) + ... % Conduction in +X
        k_pv * 2 * (-T(i, j, 251) + T(i, j-1, 251)) / (dy)^2 + ... % Conduction in Y
        k_pv * 2 * ((T(i, j, 251) - T(i, j, 250))) / (dz)^2 + ... % Conduction in Z
        (h * (T_inf_t - T(i, j, 251)) / (dz/2))); % Convection term
end

% Loop for nodes on the edge for i=26, j=4:200, k=1
for j = 4:200
    k = 1; % Z = 0 plane (k = 0)
    % Discretized equation for node T(i,j,1)
    T_new(26, j, k) = T(26, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * 2 * ((T(25, j, k) - T(26, j, k)) / (dx)^2) + ... % Conduction in +X
        k_pv * ((T(26, j+1, k) - 2*T(26, j, k) + T(26, j-1, k)) / (dy)^2) + ... % Conduction in Y
        k_pv * 2 * ((T(26, j, k+1) - T(26, j, k)) / (dz)^2) + ... % Conduction in Z
        (h * (T_inf_t - T(26, j, k)) / (dz/2)) + (h * (T_inf_t - T(26, j, k)) / (dx/2))); % Convection term
end

% Loop for nodes on the edge for i=26, j=4:200, k=251
for j = 4:200
    k = 251; % Z = 0 plane (k = 0)
    % Discretized equation for node T(i,j,1)
    T_new(26, j, k) = T(26, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * 2 * ((T(25, j, k) - T(26, j, k)) / (dx)^2) + ... % Conduction in +X
        k_pv * ((T(26, j+1, k) - 2*T(26, j, k) + T(26, j-1, k)) / (dy)^2) + ... % Conduction in Y
        k_pv * 2 * ((T(26, j, k+1) - T(26, j, k)) / (dz)^2) + ... % Conduction in Z
        (h * (T_inf_t - T(26, j, k)) / (dz/2)) + (h * (T_inf_t - T(26, j, k)) / (dx/2))); % Convection term
end

% Loop for nodes on the edge for j = 1 k=1
for i = 2:25
    k=1;
    T_new(i, 1, k) = T(i, 1, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * ((T(i+1, 1, k) - 2*T(i, 1, k) + T(i-1, 1, k)) / (dx)^2) + ... % Conduction in X
        k_pv * 2 * ((T(i, 2, k) - T(i, 1, k)) / (dy)^2) + ...
        k_pv * 2 * ((T(i, 1, k+1) - T(i, 1, k)) / (dz)^2) + ... % Conduction in Z
        h * (T_inf_t - T(i, 1, k)) / (dz/2)); % Convection at the front
end

% Loop for nodes on the edge for j = 1 k=251
for i = 2:25
    k=251;
    T_new(i, 1, k) = T(i, 1, k) + (dt / (rho_pv * cp_pv)) * ( ...
        k_pv * ((T(i+1, 1, k) - 2*T(i, 1, k) + T(i-1, 1, k)) / (dx)^2) + ... % Conduction in X
        k_pv * 2 * ((T(i, 2, k) - T(i, 1, k)) / (dy)^2) + ...
        k_pv * 2 * ((T(i, 1, k+1) - T(i, 1, k)) / (dz)^2) + ... % Conduction in Z
        h * (T_inf_t - T(i, 1, k)) / (dz/2)); % Convection at the front
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Loop over XZ plane for j = 1 k=2:250 i=2:325
for i = 1:325
    for k = 1:251 % Loop over Z direction (top to bottom)
        T_new(i, 1, k) = T(i, 2, k);
    end
end

% Loop over XZ plane for j=201 k=2:250 i=2:25
for i = 1:25
    for k = 1:251 % Loop over Z direction (top to bottom)
        T_new(i, 201, k) = T(i, 200, k);
    end
end

% Loop for nodes on the YZ plane for i=2:25, j=2:200, k=1
for j = 2:200
    for i=2:25
        % Discretized equation for node T(i,j,1)
        T_new(i, j, 1) = T(i, j, 1) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * ((T(i+1, j, 1) - 2*T(i, j, 1) + T(i-1, j, 1)) / (dx)^2) + ... % Conduction in +X
            k_pv * ((T(i, j-1, 1) - 2*T(i, j, 1) + T(i, j+1, 1)) / (dy)^2) + ... % Conduction in Y
            k_pv * 2 * ((T(i, j, 2) - T(i, j, 1))) / (dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, 1)) / (dz/2))); % Convection term
    end
end

% Loop for nodes on the YZ plane for i=2:25, j=2:200, k=251
for j = 2:200
    for i=2:25
        % Discretized equation for node T(i,j,1)
        T_new(i, j, 251) = T(i, j, 251) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * ((T(i+1, j, 251) - 2*T(i, j, 251) + T(i-1, j, 251)) / (dx)^2) + ... % Conduction in +X
            k_pv * ((T(i, j-1, 251) - 2*T(i, j, 251) + T(i, j+1, 251)) / (dy)^2) + ... % Conduction in Y
            k_pv * 2 * ((T(i, j, 250) - T(i, j, 251))) / (dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(i, j, 251)) / (dz/2))); % Convection term
    end
end
end
end

```

```

% Loop for nodes on the YZ plane for i=1, j=2:200, k=2:250
for j = 2:200
    for k=2:250
        % Discretized equation for node T(1,j,1)
        T_new(1, j, k) = T(1, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * 2*((T(2, j, k) - T(1, j, k)) / (dx)^2) + ... % Conduction in +X
            k_pv * ( T(1,j-1,k) - 2*T(1, j, k) + T(1, j+1, k)) / (dy)^2 + ... % Conduction in Y
            k_pv * ( T(1,j,k+1) - 2*T(1, j, k) + T(1, j, k-1)) / (dz)^2 + ... % Conduction in Z
            Qrad / (dx/2) + ... % Radiation term
            (h * (T_inf_t - T(1, j, k))) / (dx/2)); % Convection term
        end
    end

% Loop for nodes on the YZ plane for i=26, j=4:200, k=2:250
for j = 4:200
    for k=2:250
        % Discretized equation for node T(1,j,1)
        T_new(26, j, k) = T(26, j, k) + (dt / (rho_pv * cp_pv)) * ( ...
            k_pv * 2*((T(25, j, k) - T(26, j, k)) / (dx)^2) + ... % Conduction in +X
            k_pv * ( T(26,j-1,k) - 2*T(26, j, k) + T(26, j+1, k)) / (dy)^2 + ... % Conduction in Y
            k_pv * ( T(26,j,k+1) - 2*T(26, j, k) + T(26, j, k-1)) / (dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(26, j, k))) / (dx/2)); % Convection term
        end
    end

% Loop for nodes on the XZ plane for i=27:325, j=3, k=2:250
for k = 2:250
    for i=27:325
        % Discretized equation for node T(1,j,1)
        T_new(i, 3, k) = T(i, 3, k) + (dt / (rho_fin * cp_fin)) * ( ...
            k_fin * ((T(i+1, 3, k) - 2*T(i, 3, k) + T(i-1, 3, k)) / (dx)^2) + ... % Conduction in +X
            k_fin * ( T(i,2,k) - T(i, 3, k)) / (dy)^2 + ... % Conduction in Y
            k_fin * ((T(i, 3, k) - T(i, 3, k)) / (dz)^2 + ... % Conduction in Z
            (h * (T_inf_t - T(i, 3, k)) / (dy/2)); % Convection term
        end
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 2:(nx-1)
    for j = 2:(ny-1)
        for k = 2:(nz-1)
            if i <= u
                rho = rho_pv;
                cp = cp_pv;
                k_val = k_pv;
            else
                rho = rho_fin;
                cp = cp_fin;
                k_val = k_fin;
            end
            T_new(i, j, k) = T(i, j, k) + (dt / (rho * cp)) * ...
                (k_val * ((T(i+1, j, k) - 2*T(i, j, k) + T(i-1, j, k)) / dx^2 + ...
                    (T(i, j+1, k) - 2*T(i, j, k) + T(i, j-1, k)) / dy^2 + ...
                    (T(i, j, k+1) - 2*T(i, j, k) + T(i, j, k-1)) / dz^2));
        end
    end

fprintf('Time: %.2f s, T(26, 3, 10) = %.2f K\n', time(t_idx), T_new(26, 3, 10));
fprintf('Time: %.2f s, T(26, 4, 10) = %.2f K\n', time(t_idx), T_new(26, 4, 10));
fprintf('Time: %.2f s, T(30, 4, 10) = %.2f K\n', time(t_idx), T_new(30, 4, 10));

% Update T for next iteration
T = T_new;
if time(t_idx) == 10
    fprintf('At t = 10s, the Temperature at (17,8,9) is %g\n', T(17,8,9));
    break; %
end

% Check for nodes exceeding 310 K
[exceeding_indices_x, exceeding_indices_y, exceeding_indices_z] = ind2sub(size(T), find(T > 310));
if ~isempty(exceeding_indices_x)
    for idx = 1:length(exceeding_indices_x)
        fprintf('Temperature exceeds 310 K at position (x=%d, y=%d, z=%d) at time %.2f s\n', ...
            exceeding_indices_x(idx), exceeding_indices_y(idx), exceeding_indices_z(idx), time(t_idx));
    end
end

% Check for nodes with temperature below 268 K
[low_temp_indices_x, low_temp_indices_y, low_temp_indices_z] = ind2sub(size(T), find(T < 268));
if ~isempty(low_temp_indices_x)
    for idx = 1:length(low_temp_indices_x)
        fprintf('Temperature below 268 K at position (x=%d, y=%d, z=%d) at time %.2f s\n', ...
            low_temp_indices_x(idx), low_temp_indices_y(idx), low_temp_indices_z(idx), time(t_idx));
    end
end
end

```