ME685
Applied Numerical Methods (ANM)
Term Project 2024 - 25 (2nd Semester)

Project Title – "Numerical Simulation of Transient Heat Conduction Using the Finite Difference Method"

Guided by	*Prof. Abhishek Sarkar*
Group Members	*5*
Abhishek Rajput	*241050603*
Shubham Sharnagat	*241050081*
Kamal Verma	*241050033*
Mridul Kumar	*242050602*
Mugilan T	*241050044*

Table of Contents

1. Introduction 3
2. Methodology 3
2.1 Governing Equation 4
2.2 Discretization 5
2.3 Boundary and Initial Conditions 6
2.4 Laser Heat Source (Gaussian Profile) 6
2.5 MATLAB Implementation 7
3. Results and Discussion 9
3.1 Temperature Evolution 9
3.2 Visualization Outputs 9
Time 12
Peak Temperature 12
End Temperature 12
3.3 Peak Temperature Behaviour 12
3.4 Centreline Temperature Profile 12
4. Conclusion 13
5. References 14
6. Appendix 15

1. Introduction

This project is carried out as a part of the Applied Numerical Methods (ANM) course to demonstrate the use of numerical techniques for solving real-world engineering problems involving heat transfer. A key focus is on transient heat conduction, which is critical in various thermal processes such as welding, machining, and laser-based manufacturing.

In many such processes, understanding the heat conduction behaviour due to a stationary or moving heat source is essential, as the resulting temperature distribution affects thermal gradients, cooling rates, and potential material defects. To explore this, we simulate transient heat conduction in a 2D metallic domain subjected to a Gaussian heat source, using methods learned in the ANM course.

In this updated project model, material properties for a Titanium alloy (Ti-6Al-4V) are used to make the simulation more realistic. The model is solved numerically using the Finite Difference Method (FDM) with an explicit time-stepping scheme. MATLAB is used for implementation, visualization, and video generation. This project serves as a practical demonstration of numerical modelling techniques and stability criteria covered in ANM and provides valuable insights into thermal behaviour under localized heat input.

The spatial and temporal resolution of the simulation are carefully chosen to satisfy stability requirements, specifically the Fourier number criterion. The Gaussian heat source introduces a localized, time-varying thermal input, challenging the numerical scheme to accurately capture steep thermal gradients. The study highlights how changes in source intensity, exposure duration, and material properties affect the heat penetration and cooling profiles. Such simulations help develop a deeper understanding of transient thermal responses, which are crucial in optimizing industrial heat treatment and manufacturing processes.

2. Methodology

This project numerically solves the two-dimensional transient heat conduction equation using the Finite Difference Method (FDM). A stationary Gaussian laser heat source is applied at the surface, and convective heat loss is incorporated through boundary conditions. The

domain is discretized using a structured Cartesian grid, and an explicit time-stepping scheme is employed for temporal advancement.

Material properties of Ti-6Al-4V are used to define thermal conductivity, density, and specific heat. The simulation is implemented in MATLAB with a modular code structure, separating input definition, numerical computation, visualization, and video generation. Stability is ensured by satisfying the Fourier number criterion, and appropriate time step and grid size are selected to balance accuracy and computational cost.

The temperature distribution is updated iteratively over time, capturing the transient thermal response of the material to the localized heat input. Results are visualized through contour plots and time-lapse animations to analyse heat diffusion, thermal gradients, and the effects of heat source parameters. All numerical techniques and formulations are based on topics covered in the Applied Numerical Methods (ANM) course

2.1 Governing Equation

The mathematical model is governed by the transient heat conduction equation with both internal heating and convective cooling:

$$\rho\, c_p \frac{\partial T}{\partial t} = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) + Q(x,y) - h\,(T - T_{am\beta})$$

Where:

- **T(x, y, t):** Temperature field **[K]**
- **ρ:** Density of titanium alloy **(4430 kg/m³)**
- **c_p:** Specific heat capacity **(560 J/kg·K)**
- **k:** Thermal conductivity **(6.7 W/m·K)**
- **α:** Thermal diffusivity **(k / ρ c_p)**
- **Q(x, y):** Gaussian heat source **[W/m³]**
- **P:** Laser power **(1500 W)**
- **r_o:** Laser beam radius **(50 µm = 0.00005 m)**
- **x_o, y_o:** Center coordinates of laser **(0.0015 m, 0.0015 m)**
- **$T_{am\beta}$:** Ambient temperature **(300 K)**
- **h:** Convective heat transfer coefficient **(50 W/m²·K)**
- **Δx, Δy:** Grid spacing in x and y directions **[m]**
- **Δt:** Time step **[s]**

2.2 Discretization

T To numerically solve the transient heat conduction equation, the domain is discretized using a structured Cartesian grid with uniform spacing Δx=Δy. The finite difference method (FDM) is employed to approximate the partial derivatives, and an explicit forward Euler scheme is used for time integration.

Spatial Discretization

The second-order spatial derivatives in the x and y directions are discretized using central difference schemes:

$$\left.\frac{\partial^2 T}{\partial x^2}\right|_{i,j} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2}$$

$$\left.\frac{\partial^2 T}{\partial y^2}\right|_{i,j} \approx \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2}$$

Since Δx = Δy, the Laplacian simplifies to the sum of these two terms.

Temporal Discretization

The transient term is discretized using a forward difference in time:

$$\left.\frac{\partial T}{\partial t}\right|_{i,j} \approx \frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t}$$

Combining the discretized terms, the temperature update equation becomes:

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \Delta t \left[\alpha\left(\frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{(\Delta x)^2} + \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{(\Delta y)^2}\right) + \frac{Q_{i,j}}{\rho c_p} - \frac{h}{\rho c_p}\left(T_{i,j}^{n} - T_{\text{ambient}}\right)\right]$$

Here:

- $\alpha = \frac{k}{\rho c_p}$ is the thermal diffusivity.

- $Q_{i,j}$ is the heat input at point (i, j) from the Gaussian Source.
- The convective term is included as a volumetric sink, scaled appropriately.

2.3 Boundary and Initial Conditions

- **Initial Condition:**

$$\boldsymbol{T(x, y, 0) = 300\ K}$$

- **Boundary Conditions:**

  All four edges are treated as insulated, modelled using Neumann (zero-gradient) conditions:

$$\frac{\boldsymbol{\partial T}}{\boldsymbol{\partial n}} = \mathbf{0}$$

These are enforced by copying the adjacent interior node values to the boundaries:

$$T_{1,j} = T_{2,j}, \quad T_{N_x,j} = T_{N_x-1,j}$$

$$T_{i,1} = T_{i,2}, \quad T_{i,N_y} = T_{i,N_y-1}$$

This ensures no heat flow across the boundaries.

This is implemented by copying adjacent interior node values to the boundary points.

2.4 Laser Heat Source (Gaussian Profile)

The laser input is modelled using a 2D Gaussian distribution centred at the domain's midpoint (x_0, y_0):

$$\boldsymbol{Q(x, y) = (2P / \pi {r_0}^2) * \exp 2 * ((x - x_0)^2 + (y - y_0)^2) / {r_0}^2]}$$

Where:

- P = 1500 W is the laser power
- r_0 = 0.00005 m (50 µm) is the laser beam radius
- x_0 = y_0 = 0.0015 m is the centre of the domain

The Gaussian profile is used in this simulation because it accurately represents the intensity distribution of a laser, where the maximum heat is concentrated at the centre and diminishes symmetrically towards the edges. This is typical of most real-world laser beams, making the Gaussian function an ideal choice for modelling the heat source. By using this profile, the simulation can more realistically capture the temperature distribution and heat conduction behaviour in the material, which is crucial for understanding the thermal effects in processes like welding or additive manufacturing. Moreover, the Gaussian profile ensures smooth, physically meaningful results and enhances the stability and accuracy of the numerical solution.

2.5 MATLAB Implementation

The simulation is implemented in MATLAB using modular code structure. The following features are included:

- Calculation of the laser heat input using the Gaussian formula
- Temperature update with conduction, laser heating, and convective cooling
- Insulated boundary conditions using mirrored interior values
- Real-time visualization and MP4 video recording
- Final contour plots and cross-sectional temperature graphs

This setup provides a clear and stable numerical solution, and allows visualization of how heat diffuses through the metal under realistic conditions — directly applying techniques learned in the ANM course.

Pseudo Code :

Main()

-Initialize material properties: ρ, cp, k, T_ambient, h

-Compute thermal diffusivity: $\alpha = \frac{k}{\rho c_p}$

-Define spatial domain: Nx, Ny, L, dx, dy

-Create meshgrid (X, Y) using linspace and meshgrid

-Set simulation time: t_end, compute dt using stability condition, calculate Nt

-Define laser parameters: P, r0, x0, y0

-Initialize temperature matrix T to T_ambient

-Set snapshot times and initialize video writer

-Save initial snapshot

For n = 1 to Nt:

- Compute Q using gaussianHeatSource(X, Y, x0, y0, P, r0)
- Update temperature field using computeTemp()
- If current time step matches snapshot time, store T
- Every few steps, plot and record video frame

End loop

Close video

-Plot and save temperature snapshots

- Plot final contour
- Plot temperature along centre line x = L/2 for all snapshots

End Main()

Function gaussianHeatSource(X, Y, x0, y0, P, r0)

- Return Q using Gaussian distribution:

$$Q(x,y) = \frac{2P}{\pi r_0^2} \exp\left(-\frac{2((x-x_0)^2 + (y-y_0)^2)}{r_0^2}\right)$$

Function computeTemp(T, Q, α, dt, dx, ρ, cp, T_ambient, h)

- For each interior node (i, j):
- Compute second-order central differences $\nabla^2 T$
- Update T using explicit time stepping:

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \Delta t \left[\alpha \left(\frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{\Delta x^2} + \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{\Delta y^2} \right) + \frac{Q_{i,j}}{\rho c_p} - \frac{h}{\rho c_p} \left(T_{i,j}^{n} - T_{\text{ambient}} \right) \right]$$

- Apply Neumann (zero-flux) BCs by copying adjacent interior values

- Return T_new

3. Results and Discussion

The simulation was carried out on a 2D domain (3 mm × 3 mm) representing a Titanium alloy plate subjected to a stationary Gaussian laser beam. The solution to the discretized heat conduction equation was computed over time, and the temperature field was visualized to observe the spatial and temporal behaviour of the heat flow.

3.1 Temperature Evolution

- The initial temperature of the entire domain was set to T = 300 K.
- A Gaussian heat source centred in the domain rapidly raised the temperature in the surrounding area.
- Heat diffusion occurred symmetrically outward from the laser spot, following the conduction-dominated behaviour of the material.
- The influence of convective cooling helped maintain realistic peak temperatures over time.

3.2 Visualization Outputs

The following outputs were generated using **MATLAB:**

- A real-time temperature evolution video **(Temperature_Evolution.mp4)**
- Contour plots of temperature at the final time step
- Line plot showing temperature variation along the vertical centerline **(x = L/2)**

These visualizations are essential for understanding how the temperature gradients develop in Laser melting/LPBF processes and how they can be controlled through laser parameters and material properties.

Temperature evolution at different time steps:

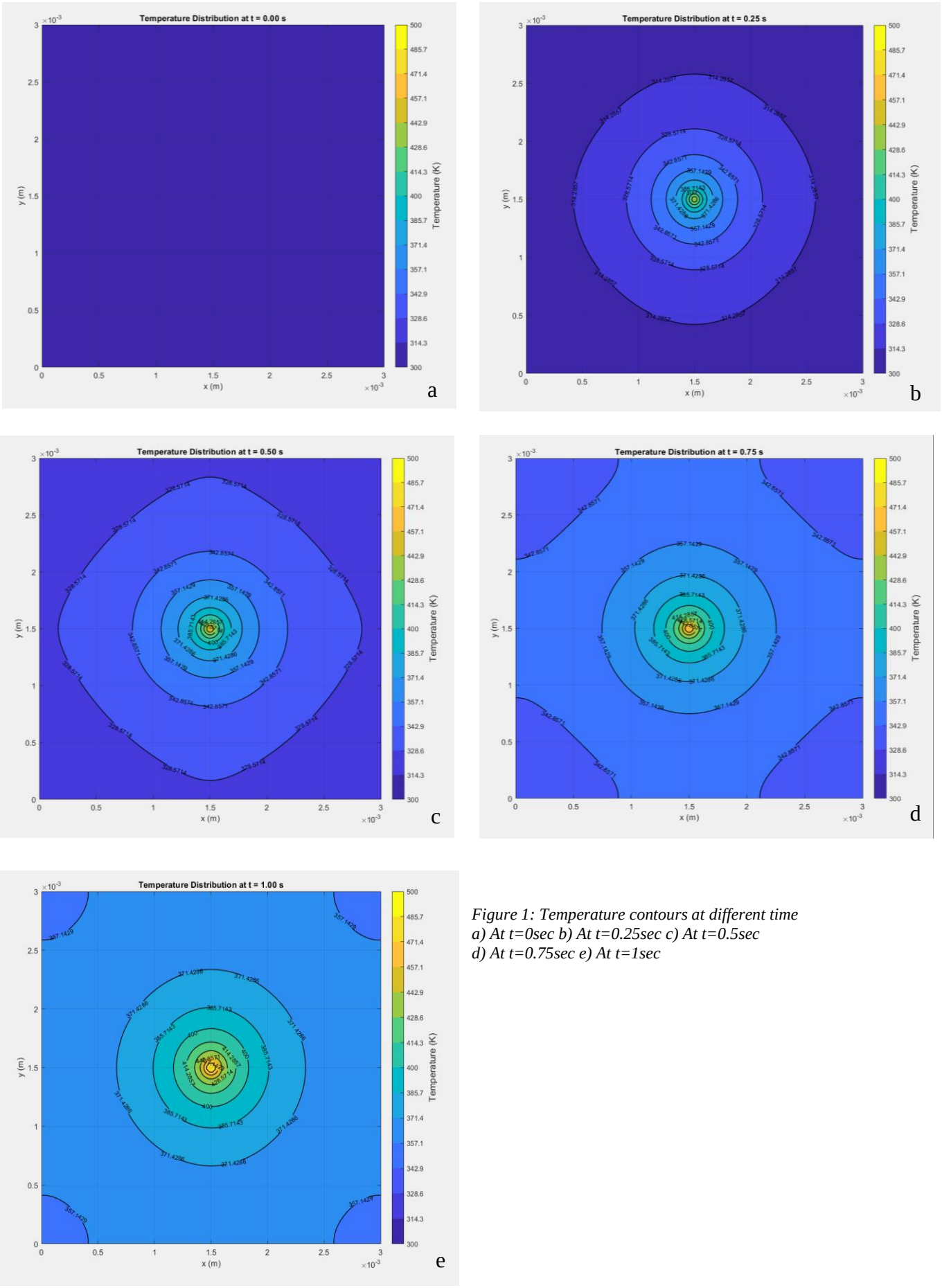

Figure 1: Temperature contours at different time
a) At t=0sec b) At t=0.25sec c) At t=0.5sec
d) At t=0.75sec e) At t=1sec

Temperature Plot at Final Time step:

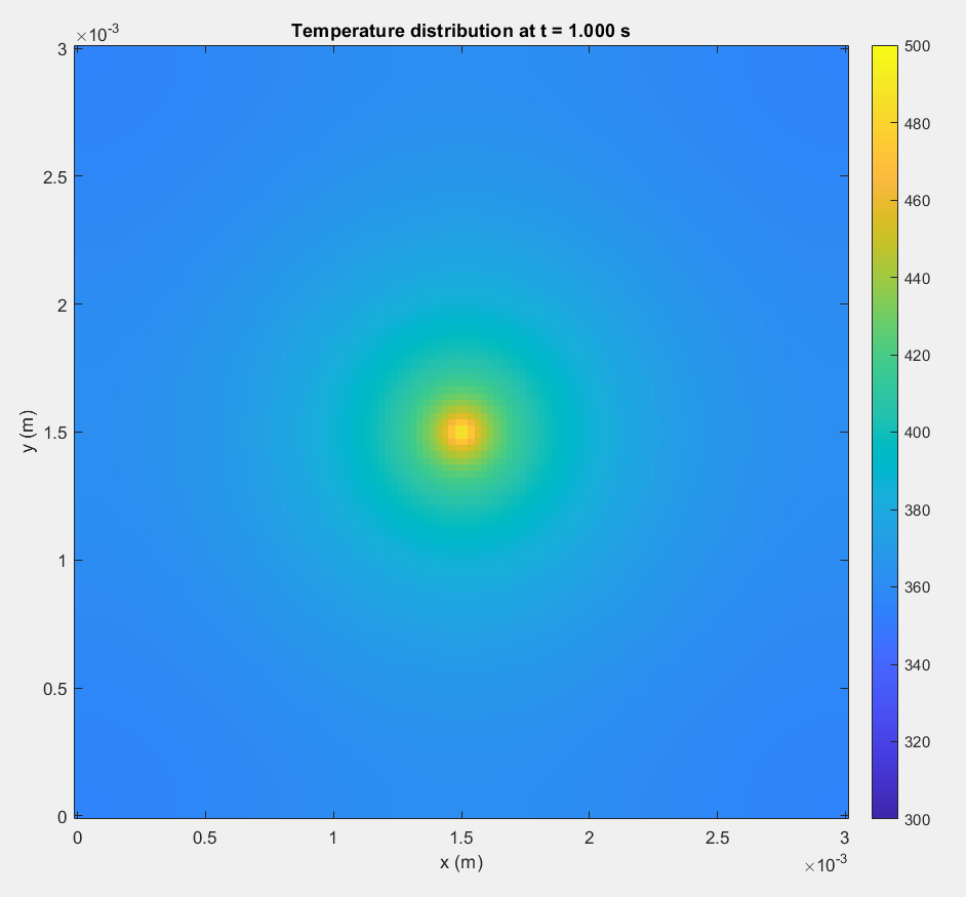

Figure 2: Final Temperature distribution at t=1sec

Line plot showing temperature variation along the vertical centreline (x = L/2) :

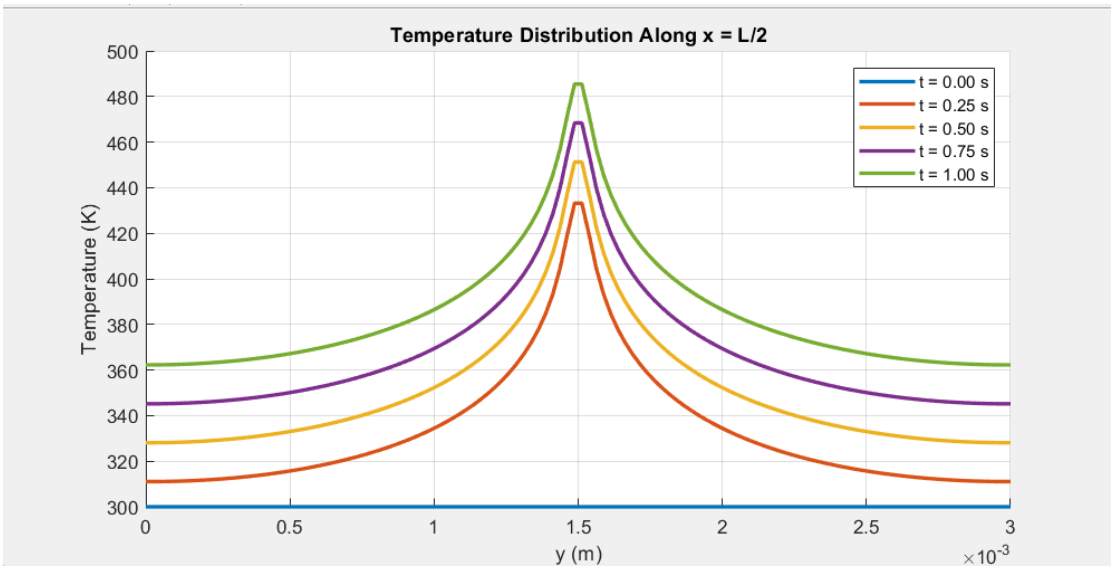

Figure 3: Centre line temperature with time

The graph illustrates the temperature distribution along the centreline x=L/2 for different time intervals. Initially, at t = 0, the temperature is uniform and equal to the ambient value of 300 K. As time progresses, the Gaussian heat source causes a rapid rise in temperature, especially at the centre of the domain (y = L/2), where the laser intensity is maximum. his results in a distinct peak temperature at the centre, which increases with time due to continuous heat input. Away from the centre, the temperature decreases gradually, indicating the diffusion of heat throughout the material. The growth and spread of the peak over time also reflect the heat conduction process and the broadening of the heat-affected zone.

Time	Peak Temperature	End Temperature
At t=0s	300.00	300.00
At t=0.25s	433.17	311.04
At t= 0.5s	451.27	328.1
At t= 0.75s	468.40	345.18
At t= 1s	485.48	362.26

Table1: Peak and end temperature at centreline

3.3 Peak Temperature Behaviour

The highest temperature was observed at the centre of the laser beam, where the Gaussian heat input was most intense. Over time, due to conduction and convective losses, the system moved toward a stabilized thermal field.

- The peak temperature $\mathbf{T_{max}}$ was found to depend on:
  - **Laser power (P)**
  - **Beam radius (r_0)**
  - **Material properties (ρ, c_p, k)**
  - **Convective loss coefficient (h)**

3.4 Centreline Temperature Profile

To examine the temperature gradient across the domain, a plot was generated at the centreline **(x = L/2):**

- The temperature distribution across the y direction was symmetric and peaked at the centre **(y = L/2).**
- This plot is valuable for estimating melt pool width and thermal gradient, both of which are important for predicting material behaviour in Laser melting/LPBF**.**

4. Conclusion

- In this project, we successfully applied numerical methods from the Advanced Numerical Methods (ANM) course to simulate transient heat conduction in a 2D metallic domain subjected to a stationary Gaussian laser heat source. The simulation was implemented in MATLAB using the Finite Difference Method (FDM) with explicit time integration, incorporating convective boundary conditions to capture realistic thermal behaviour.
- Titanium alloy (Ti-6Al-4V) was chosen for its industrial relevance, and the Gaussian heat flux profile was used to accurately represent the intensity distribution of laser-based heat sources. This profile naturally results in a sharp peak temperature at the centre of the domain, as observed in the temperature vs. position plot, where heat first accumulates at the centre and then gradually diffuses outward due to conduction and is dissipated through convection. The peak temperature evolution with time clearly demonstrates the transient nature of the process, and the plots highlight how the initial sharp temperature rise smooths out over time as heat spreads across the domain.
- The numerical results were visualized through contour plots, line plots, and an animation, providing insight into how material properties, heat source characteristics, and thermal losses influence the thermal field. While this study is particularly relevant to laser-based manufacturing methods such as Laser Powder Bed Fusion (LPBF), the methodology is equally applicable to broader thermal analysis in areas such as welding, laser surface treatment, and thermal management in electronic components.
- This work strengthened core concepts from ANM, including spatial and temporal discretization, stability analysis, and boundary condition implementation. It also forms a foundational step toward

more advanced thermal simulations involving phase change, fluid flow, and dynamic heat sources — allowing for comprehensive modelling of practical heat transfer scenarios in modern engineering applications.

5. References

1. S. Kalpakjian and S. R. Schmid, Manufacturing Engineering and Technology, 7th Edition, Pearson Education, 2014.
2. J. N. Reddy, An Introduction to the Finite Element Method, 3rd Edition, McGraw-Hill, 2006.
3. F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, Fundamentals of Heat and Mass Transfer, 7th Edition, Wiley, 2011.
4. MATLAB Documentation – https://www.mathworks.com/help
5. Lecture Notes – ME 621A: Applied Numerical Methods, Department of Mechanical Engineering, IIT Kanpur.
6. D. Gu et al., "Laser Additive Manufacturing of Metallic Components: Materials, Processes and Mechanisms," International Materials Reviews, 2012.

6. Appendix

Code used for this project is attached below:

```
clear; clc; close all;

%% Material and Physical Properties %% Titanium alloy Grade 5 (Ti-6Al-4V)
rho = 4430;              % Density (kg/m^3)
cp  = 560;               % Specific heat (J/kg.K)
k   = 6.7;               % Thermal conductivity (W/m.K)
alpha = k / (rho*cp);    % Thermal diffusivity (m^2/s)
T_ambient = 300;         % Ambient temperature (K)

%% Convective Heat Loss Parameter
h = 50;                  % Convective heat transfer coefficient (W/m^2.K)

%% Domain Setup
Nx = 120;                % Number of nodes in x-direction
Ny = 120;                % Number of nodes in y-direction
L = 0.003;               % Domain length (meters)
dx = L / Nx;             % Grid spacing in x
dy = dx;                 % Uniform grid spacing in y

x = linspace(0, L, Nx);  % x-coordinate vector
y = linspace(0, L, Ny);  % y-coordinate vector
[X, Y] = meshgrid(x, y); % 2D grid for spatial domain

%% Time Setup
t_end = 1;                               % Simulation end time in seconds
dt = 0.25 * dx^2 / alpha;                % Time step (CFL condition for
stability)
Nt = floor(t_end / dt);                  % Total number of time steps

%% Snapshot Times Setup
% Snapshots for 0, 0.25, 0.5, 0.75, and 1.0 seconds
snapshot_times = [0, 0.25, 0.5, 0.75, 1.0];
snapshot_indices = round(snapshot_times / dt);
T_snapshots = cell(length(snapshot_times), 1);
snapshot_counter = 1;

%% Laser (Gaussian heat source) Parameters
P = 1500;              % Laser power (W)
r0 = 0.00005;          % Laser beam radius (m) [50 um]
x0 = L / 2;            % Laser center x-coordinate
y0 = L / 2;            % Laser center y-coordinate

%% Initial Condition
T = T_ambient * ones(Nx, Ny);  % Initialize ambient temperature field

T_snapshots{snapshot_counter} = T;
snapshot_counter = snapshot_counter + 1;

%% MP4 Video Setup (Optional)
v = VideoWriter('Temperature_Evolution.mp4', 'MPEG-4'); % Create video file
v.FrameRate = 30;
```

```
v.Quality = 100;
open(v);  % Open video file for writing

fig = figure('Position', [200, 0, 1000, 800]); % Figure for simulation
video frames

%% Main Time-Stepping Loop
for n = 1:Nt
    Q = gaussianHeatSource(X, Y, x0, y0, P, r0);  % Compute laser heat
source
    T = computeTemp(T, Q, alpha, dt, dx, rho, cp, T_ambient, h);  % Update
temperature field

    if snapshot_counter <= length(snapshot_indices) && n ==
snapshot_indices(snapshot_counter)
        T_snapshots{snapshot_counter} = T;
        snapshot_counter = snapshot_counter + 1;
    end

    % Render and record a video frame every 30 time steps
    if mod(n, 30) == 0
        imagesc(x, y, T');
        set(gca, 'YDir', 'normal');
        colorbar;
        title(['Temperature distribution at t = ', num2str(n*dt, '%.3f'), '
s']);
        xlabel('x (m)');
        ylabel('y (m)');
        caxis([300 500]);
        axis square;
        drawnow;
        frame = getframe(fig);
        writeVideo(v, frame);
    end
end

if snapshot_counter <= length(snapshot_times)
    T_snapshots{snapshot_counter} = T;
end

close(v);
disp('Simulation complete. Video and snapshots are created.');

%% Mid-line (x = L/2) Temperature Line Plot
mid_x_index = round(Nx/2);  %  x = L/2

figure('Position', [250, 100, 800, 400]);
hold on;

for i = 1:length(snapshot_times)
    T_cross = T_snapshots{i}(mid_x_index, :);  % Temperature along the mid-
line
    plot(y, T_cross, 'LineWidth', 2, 'DisplayName', sprintf('t = %.2f s',
snapshot_times(i)));
end

title('Temperature Distribution Along x = L/2');
xlabel('y (m)');
ylabel('Temperature (K)');
```

```
grid on;
legend('show', 'Location', 'best');
axis([min(y) max(y) 300 500]);
hold off;
saveas(gcf, 'Combined_TempLine_xL2.png');

%% Discrete Contour Plots
nLevels = 15;
levels = linspace(300, 500, nLevels);  % Define levels from 300 to 500 K

for i = 1:length(snapshot_times)
    figure('Position',[200, 0, 1000, 800]);

    [Cs_filled, hContourF] = contourf(x, y, T_snapshots{i}', levels,
'LineStyle','none');
    colormap(parula(nLevels-1));
    caxis([levels(1), levels(end)]);

    cb = colorbar;
    cb.Ticks = levels;
    cb.TickLabels = round(levels, 1);
    cb.Label.String = 'Temperature (K)';
    cb.Label.FontSize = 12;

    hold on;
    [C, hContourL] = contour(x, y, T_snapshots{i}', levels,
'LineColor','k', 'LineWidth',1);
    clabel(C, hContourL, 'FontSize', 8, 'Color', 'k');

    title(sprintf('Temperature Distribution at t = %.2f s',
snapshot_times(i)));
    xlabel('x (m)');
    ylabel('y (m)');
    axis square;
    grid on;
    hold off;

    saveas(gcf, sprintf('Discrete_Contour_Temperature_t_%.2f_s.png',
snapshot_times(i)));
end

%% Functions Used

% Gaussian heat source function
function Q = gaussianHeatSource(X, Y, x0, y0, P, r0)
    % Returns a 2D Gaussian heat source centered at (x0, y0)
    Q = (2*P/(pi*r0^2)) .* exp(-2*((X - x0).^2 + (Y - y0).^2) / r0^2);
end

% Temperature field update function
function T_new = computeTemp(T, Q, alpha, dt, dx, rho, cp, T_ambient, h)
    [Nx, Ny] = size(T);
    T_new = T;
    for i = 2:Nx-1
        for j = 2:Ny-1
            d2Tdx2 = (T(i+1,j) - 2*T(i,j) + T(i-1,j)) / dx^2;
            d2Tdy2 = (T(i,j+1) - 2*T(i,j) + T(i,j-1)) / dx^2;
            T_new(i,j) = T(i,j) + dt * ( ...
```

```
                alpha*(d2Tdx2 + d2Tdy2) + ...            % Thermal
diffusion
                Q(i,j)/(rho*cp) - ...                    % Laser heating
                h*(T(i,j) - T_ambient)/(rho*cp) );       % Convective heat
loss
        end
    end

    % Apply Neumann boundary conditions (zero flux)
    T_new(1,:)   = T_new(2,:);
    T_new(end,:) = T_new(end-1,:);
    T_new(:,1)   = T_new(:,2);
    T_new(:,end) = T_new(:,end-1);
end
```