Bharath

INSTITUTE OF HIGHER EDUCATION AND RESEARCH

Declared as **Deemed-to-be-University** u/s 3 of the UGC Act, 1956

B.Tech – Aerospace Engineering

U20ASSJ10 – Numerical Heat Transfer

Lab Manual

Vision of the Institute

"Bharath Institute of Higher Education & Research (BIHER) envisions and constantly strives to provide an excellent academic and research ambience for students and members of the faculties to inherit professional competence along with human dignity and transformation of community to keep pace with the global challenges so as to achieve holistic development."

Mission of the Institute

- To develop as a Premier University for Teaching, Learning, Research, and Innovation on par with leading global universities.
- To impart education and training to students for creating a better society with ethics and morals.
- To foster an interdisciplinary approach in education, research and innovation by supporting lifelong professional development, enriching knowledge banks through scientific research, promoting best practices and innovation, industry driven and institute-oriented cooperation, globalization and international initiatives.
- To develop as a multi-dimensional institution contributing immensely to the cause of societal advancement through spread of literacy, an ambience that provides the best of international exposures, provide health care, enrich rural development and most importantly impart value based education.
- To establish benchmark standards in professional practice in the fields of innovative and emerging areas in engineering, management, medicine, dentistry, nursing, physiotherapy and allied sciences.
- To imbibe human dignity and values through personality development and social service activities.

B.Tech- Aerospace Engineering

Vision of the Department

Department of Aeronautical Engineering will endeavor to accomplish worldwide recognition with a focal point of Excellence in the field of Aeronautics by providing quality Education through world class facilities, enabling graduates turning out to be Professional Experts with specific knowledge in Aeronautical & Aerospace engineering.

Mission of the Department

- To be the state of art Teaching and Learning center with excellent infrastructure and empowered Faculties in Aeronautical & Aerospace Engineering.
- To foster a culture of innovation among students in the field of Aeronautics and Aerospace with updated professional skills to enhance research potential for sponsored research and innovative projects.
- To Nurture young individuals to be knowledgeable, skillful, and ethical professionals in their pursuit of Aeronautical & Aerospace Engineering.

B.Tech- Aerospace Engineering

Program Educational Objectives Statements (PEO)

PEO 1: Demonstrate a solid grasp of fundamental concepts in Mathematics, Science, and Engineering, essential for effectively addressing engineering challenges within the Aerospace industry.

PEO 2: Involve in process of designing, simulating, fabricating, testing, and evaluating in the field of Aerospace.

PEO 3: Obtain advanced skills to actively engage in research and development endeavors within emerging domains, while also pursuing further education opportunities.

PEO 4: Demonstrate efficient performance both as independent contributors and as valuable team members in diverse multidisciplinary projects.

PEO 5: Embrace lifelong learning and career advancement while adapting to the evolving social demands and needs.

B.Tech- Aerospace Engineering

Programme Outcomes (PO's)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and Engg. Specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, research literature, and analyze engineering problems to arrive at substantiated conclusions using first principles of mathematics, natural, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components, processes to meet the specifications with consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

B.Tech- Aerospace Engineering

Program Specific Outcomes (PSO) - R2020

PSO1: Design and analyze aerospace components/systems for aerospace industries.

PSO2: Acquire the concepts of spacecraft attitude dynamics for the prediction of spacecraft motion.

Course Outcomes (COs)

CO1	**Describe** the concepts of Numerical heat transfer and its applications .(**Understand**)
CO2	**Apply** Partial Differential Equations to solve the heat transfer problems (**Apply**)
CO3	**Describe** the methods for solving heat transfer problems in conduction(**Understand**)
CO4	**Describe** the methods for solving convective heat transfer problems(**Understand**)
CO5	**Determine** the shape factor of a material using numerical heat transfer analysis. (**Analyze**)
CO6	**Carry out** the numericals on boundary value problems and validate using matlab code **(Imitation)**
CO7	**Acquire** data for the steady and unsteady state heat transfer problems using a computer program**(Manipulation)**
CO8	**Perform** basic post-Processing techniques and represent the results in the form of graphs and tables**(Precision).**

Mapping/Alignment of COs with POs & PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	H	L		L										
CO2	M	L		L	L					M				
CO3	M	L	L	L	L					L				
CO4	H	L		L	L					L				
CO5	M	L		L	L					L				
CO6		L	L	L						L		H	H	
CO7				L								H	H	
CO8	L	L		M								H	H	

LIST OF EXPERIMENTS:

S.No	Name of Experiment	Course Outcome
1.	Solve the boundary value problem (differential equation) in matlab	CO6,CO7,CO8
2.	Solve the heat conduction equation(Parabolic PDE) using matlab PDE solver	CO6,CO7,CO8
3.	Steady-state 2D heat transfer problem using matlab program.	CO6,CO7,CO8
4.	Mathematical calculation of temperature difference and plotting of temperature variation along the fin using matlab program.	CO6,CO7,CO8
5.	Developing a numerical code for 1D, 2D heat transfer problems.	CO6,CO7,CO8
6.	Solving of Radiation Equation by using MATLAB	CO6,CO7,CO8

EXP NO : 1 DATE :	**TO SOLVE THE BOUNDARY VALUE PROBLEM (DIFFERENTIAL EQUATION) IN MATLAB.**

AIM:

To solve the boundary value problem(differential equation)in matlab.

TOOLS REQUIRED:

Matlab R2021.

DESCRIPTION:

Solving a boundary value problem (BVP) involving differential equations in MATLAB typically involves using the bvp4c function. bvp4c is a built-in MATLAB function specifically designed for solving two-point boundary value problems.

PROCEDURE:

1. Define your differential equation: Write down the differential equation that represents your physical problem.
2. Transform the problem into a system of first-order ODEs: MATLAB's bvp4c function requires the problem to be in the form of a system of first-order ordinary differential equations (ODEs). Convert your second-order ODE or higher-order ODE into a system of first-order ODEs.
3. Set up the boundary conditions: Specify the boundary conditions for your problem.
4. Write a function representing the system of ODEs: Create a MATLAB function that returns the derivatives of the unknown functions with respect to the independent variable.
5. Use bvp4c to solve the BVP: Call the bvp4c function, providing it with the ODE function, initial guess for the solution, and boundary conditions.

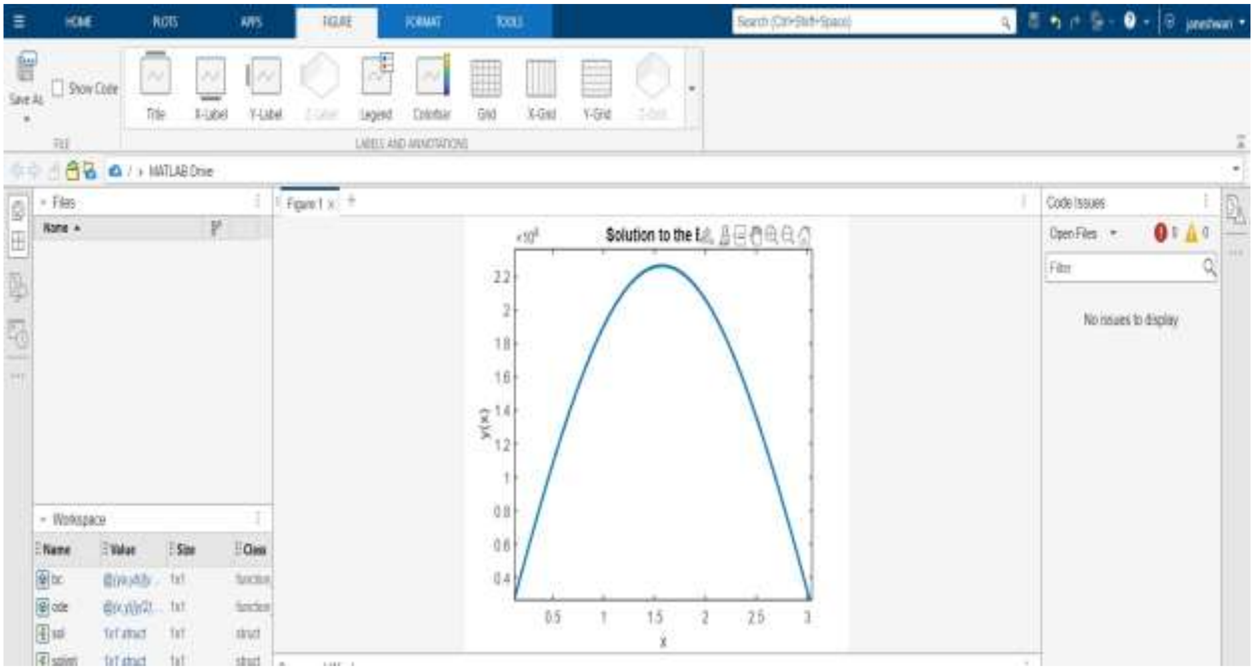
CODE:

Here's a simple example to illustrate the steps:

Suppose you want to solve the BVP for the differential equation y'' +y=0, y(0) = 0 and y(π) = 1.

```
% Define the differential equation
   ode = @(x, y) [y(2); -y(1)];

   % Define the boundary conditions
   bc = @(ya, yb) [ya(1); yb(1) - 1];

   % Initial guess for the solution
   solinit = bvpinit(linspace(0, pi, 100), [0, 0]);

   % Solve the BVP using bvp4c
   sol = bvp4c(ode, bc, solinit);

   % Plot the solution
   x = linspace(0, pi, 100);
   y = deval(sol, x);

   figure;
   plot(x, y(1, :), 'LineWidth', 2);
   xlabel('x');
   ylabel('y(x)');
   title('Solution to the BVP');
   grid on;
```

RESULTS:



Hence the boundary value problem (differential equation) in MATLAB is proved.

EXP NO : 2 DATE :	**MATHEMATICAL CALCULATION OF TEMPERATURE DIFFERENCE AND PLOTTING OF TEMPERATURE VARTION ALONG THE FIN USING MATLAB PROGRAM**

AIM:

mathematical calculation of temperature difference and plotting of temperature vartion along the fin using matlab program

TOOLS REQUIRED = Matlab R2021

DESCRIPTION:

To calculate the temperature difference along a fin and plot the temperature variation, you can use mathematical equations based on heat conduction along the fin. A common model is the one-dimensional heat conduction equation for a straight fin:

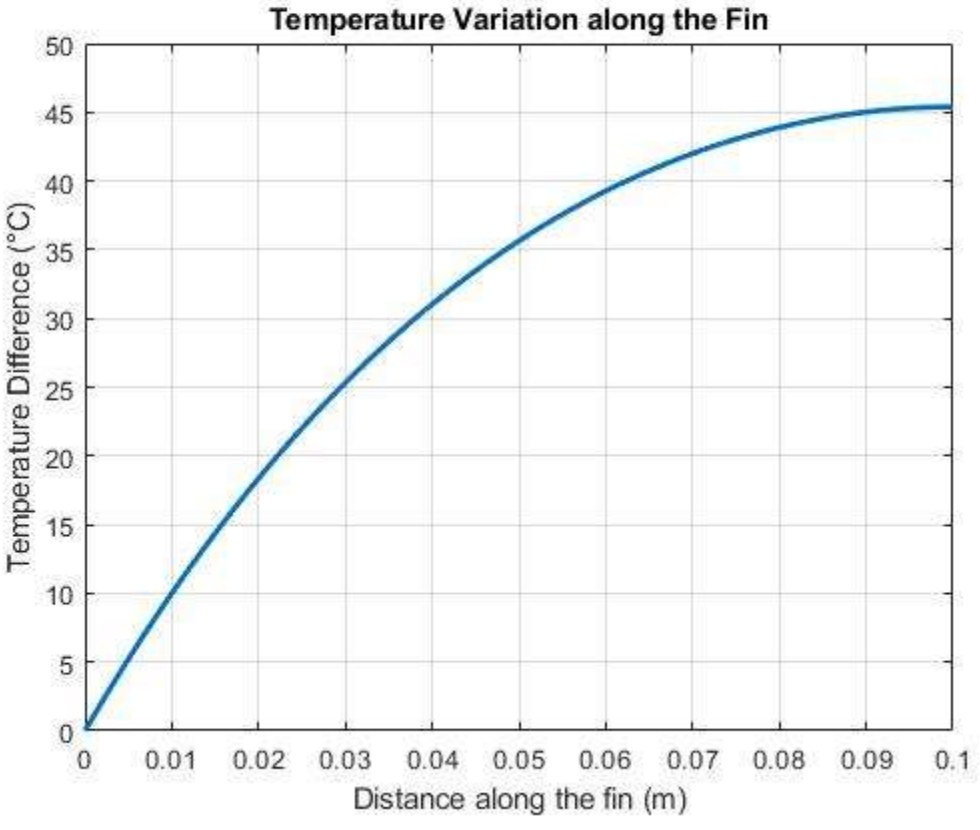
where *T* is the temperature, *x* is the distance along the fin, and *m* is a constant related to the fin geometry and material propertie

CODE:

```
function finTemperature()
    % Parameters
    L = 0.1;        % Length of the fin (m)
    k = 200;        % Thermal conductivity of the fin material (W/m-K)
    A = 0.0001;     % Cross-sectional area of the fin (m^2)
    h = 10;         % Heat transfer coefficient at the fin tip (W/m^2-K)
    T_base = 100;   % Temperature at the base of the fin (°C)
    T_air = 25;     % Ambient temperature (°C)
     P  = 0.5          % perimeter (m)
    % Calculate constant 'm'
    m = sqrt(h * P / (k * A));
    % Calculate temperature difference along the fin
```

```
    x = linspace(0, L, 100);
    T_difference = T_base - T_air - (T_base - T_air) * cosh(m * (L - x)) / cosh(m * L);
    % Plot temperature variation
    figure;
    plot(x, T_difference, 'LineWidth', 2);
    xlabel('Distance along the fin (m)');
    ylabel('Temperature Difference (°C)');
    title('Temperature Variation along the Fin');
    grid on;

end
```

RESULT:



EXP NO : 3 DATE :	TO SOLVE THE STEADY STATE 2D HEAT TRANSFER PROBLEM USING MATLAB PROGRAM.

AIM:

To solve the Steady state 2D heat transfer problem using MATLAB program.

TOOLS REQUIRED:

Matlab R2021.

DESCRIPTION:

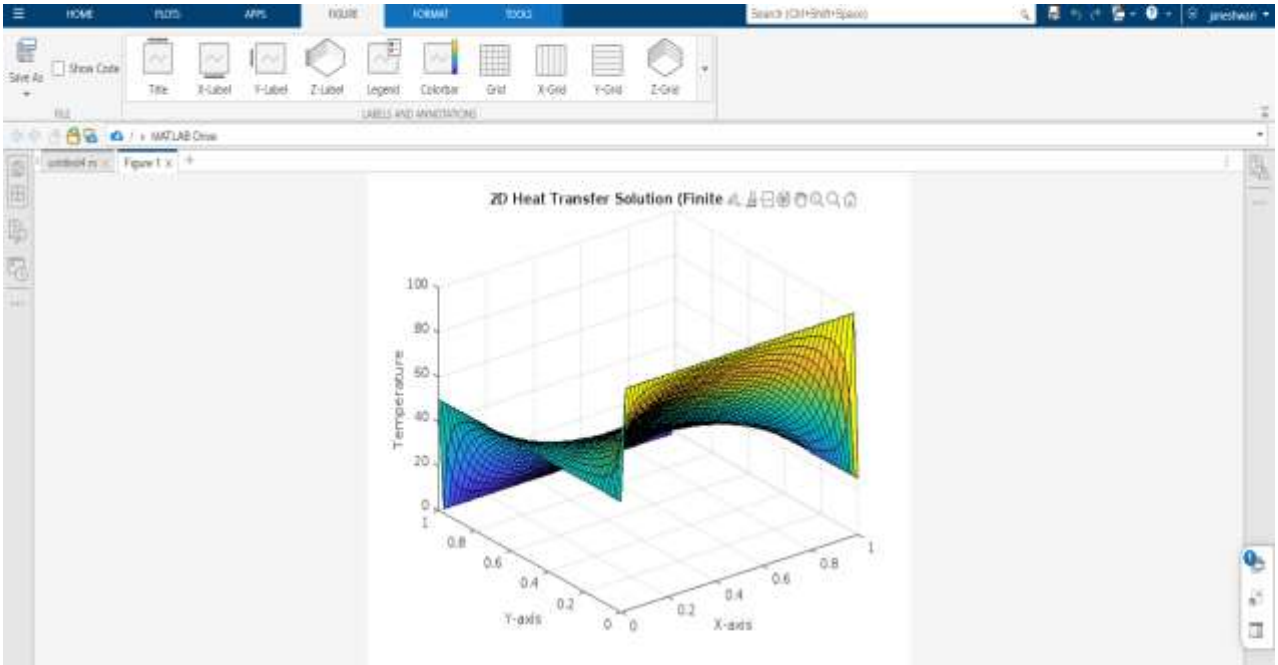
The 2-D heat conduction equation is a partial differential equation which governs the heat transfer through a medium by thermal conduction. The equation is defined as: ∂T/∂t=α[∂2/T∂/x2+∂2/T∂y2] ∂ T ∂ t = α [∂ 2/ T/ ∂ x 2 + ∂ 2 T ∂ y 2] For the steady state.

CODE:

```
% Parameters
Lx = 1; % Length of the domain in the x-direction
Ly = 1; % Length of the domain in the y-direction
Nx = 50; % Number of grid points in the x-direction
Ny = 50; % Number of grid points in the y-direction

dx = Lx / (Nx - 1);
dy = Ly / (Ny - 1);
% Thermal conductivity
k = 1;
% Initial temperature distribution
T = zeros(Nx, Ny);
% Boundary conditions
T(:, 1) = 100;    % Left boundary
T(:, end) = 0;    % Right boundary
T(1, :) = 50;     % Bottom boundary
T(end, :) = 25;   % Top boundary
```

```
% Iterative solution using finite difference
maxIter = 1000;
tolerance = 1e-5;

for iter = 1:maxIter
    T_old = T;

    for i = 2:Nx-1
    for j = 2:Ny-1
    T(i, j) = 0.25 * (T(i+1, j) + T(i-1, j) + T(i, j+1) + T(i, j-1));
    end
    end

    % Check for convergence
    if max(abs(T_old(:) - T(:))) < tolerance
    break;
    end
     end

% Plot the solution
x_values = linspace(0, Lx, Nx);
y_values = linspace(0, Ly, Ny);

[X, Y] = meshgrid(x_values, y_values);

figure;
surf(X, Y, T');
xlabel('X-axis');
ylabel('Y-axis');
zlabel('Temperature');
title('2D Heat Transfer Solution (Finite Difference)');
```

RESULT:



This code initializes a 2D grid, sets up the initial temperature distribution and boundary conditions, and then iteratively solves the steady-state heat equation using the finite difference method. The result is visualized using a 3D surface plot. You can adjust the parameters, boundary conditions, and grid size according to your specific problem. Keep in mind that this is a basic example, and for more complex problems, you may need to consider more advanced numerical methods or toolboxes like MATLAB's PDE Toolbox.

EXP NO: 4 DATE:	**SOLVING THE MATHEMATICAL CALCULATION OF TEMPERATURE DIFFERENCE AND PLOTTING OF TEMPERATURE VARIATION ALONG THE FIN USING MATLAB.**

AIM:

Solving the Mathematical calculation of temperature difference and plotting of temperature variation along the fin using MATLAB.

DISCRIPTION:

calculating the temperature difference along a fin and plotting the temperature variation using MATLAB. Let's consider a simple 1D steady-state heat conduction along a fin. The differential equation governing the temperature distribution along the fin is given by:

$$\frac{d^\wedge 2T}{dx^\wedge 2} + \frac{h.p}{k.A} \cdot (T - T\infty)$$

Where:

- T is the temperature along the fin,
- x is the distance along the fin,
- h is the convective heat transfer coefficient,
- P is the perimeter of the fin,
- k is the thermal conductivity of the material,
- A is the cross-sectional area of the fin,
- T∞ is the ambient temperature.

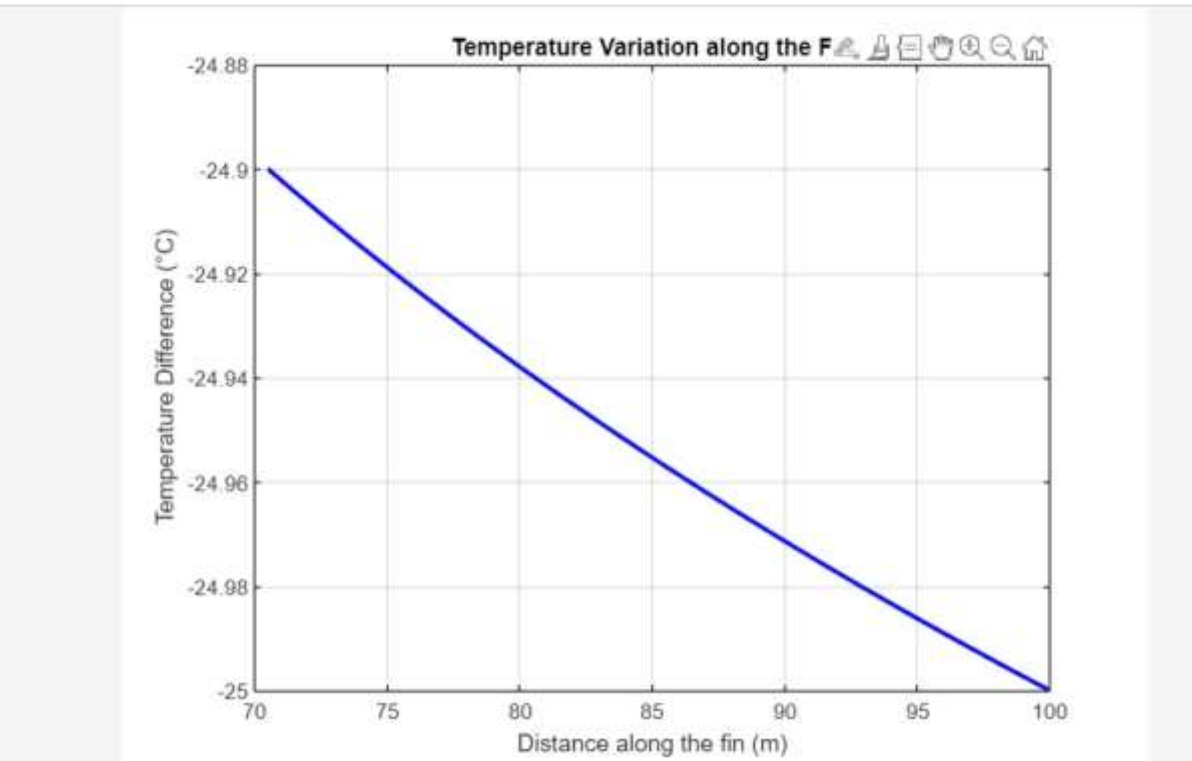
CODE:

Let's create a MATLAB script to solve this equation and plot the temperature variation along the fin:

```
% Parameters
h = 10;       % Convective heat transfer co efficient (W/m^2K)
P = 0.01;     % Perimeter of the fin (m)
k = 200;      % Thermal conductivity of the material (W/mK)
A = 1e-4;     % Cross-sectional area of the fin (m^2)
T infinity = 25;  % Ambient temperature (°C)
```

```
% Define the differential equation
ode = @(x, T) heat conduction ode(x, T, h, P, k, A, T infinity);

% Solve the differential equation
X span = Lin space(0, 0.1, 100);  % Define the distance along the fin
Initial condition = (100);       % Initial temperature at the base of the fin
[T, x] = ode45(ode, x span, initial condition);

% Calculate temperature difference
temperature_difference = T - T_infinity;

% Plot temperature variation along the fin
figure;
plot(x, temperature_difference, 'b-', 'LineWidth', 2);
xlabel('Distance along the fin (m)');
ylabel('Temperature Difference (°C)');
title('Temperature Variation along the Fin');
grid on;

% Define the ODE for heat conduction along the fin
function dTdx = heat_conduction_ode(~, T, h, P, k, A, T_infinity)
    dTdx = -h * P / (k * A) * (T - T_infinity);
end.
```

RESULT:



This script defines the differential equation for heat conduction along the fin and uses the ode45 solver to find the temperature distribution along the fin. The script then calculates the temperature difference with respect to the ambient temperature and plots the temperature variation along the fin.

EXP NO: 5 DATE:	DEVELOP NUMERICAL CODE FOR 1D, 2D, HEAT TRANSFER PROBLEM

AIM: develop numerical code for 1D, 2D, heat transfer problem

TOOLS REQUIRED = Matlab R2021

DESCRIPTION:

numerical codes for solving the 1D and 2D heat transfer problems using heat equation for simplicity:

$2\partial t/\partial u=\alpha\ \partial x^2/\partial^2 u$

CODE:

1. **Here's the MATLAB code for the 1D heat transfer problem:**

```
function solve1DHeatTransfer()
    % Parameters
    L = 1;             % Length of the rod
    T = 0.1;           % Total simulation time
    alpha = 0.01;      % Thermal diffusivity
    Nx = 100;          % Number of spatial grid points
    Nt = 200;          % Number of time steps

    % Discretization
    dx = L / (Nx - 1);
    dt = T / Nt;
    x = linspace(0, L, Nx);
    t = linspace(0, T, Nt);
    % Initial condition
    u0 = sin(pi * x);
    % Finite difference scheme
    u = zeros(Nx, Nt);
    u(:, 1) = u0;
    for n = 2:Nt
```

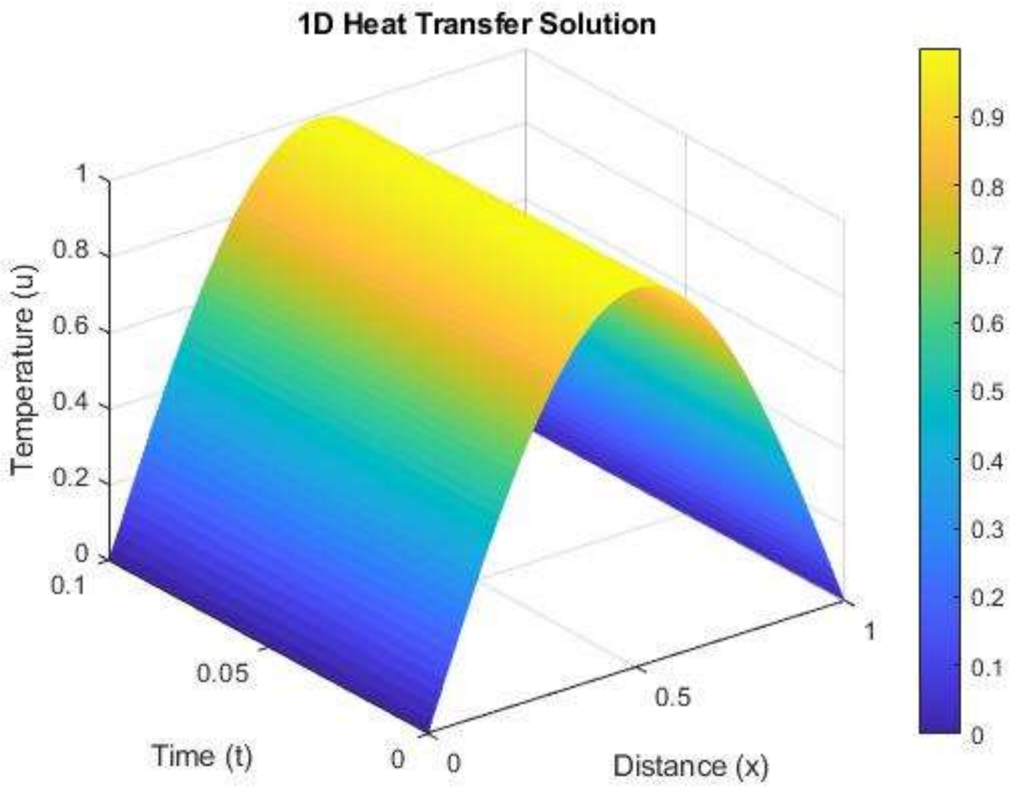
```
    for i = 2:Nx-1
   u(i, n) = u(i, n-1) + alpha * dt / dx^2 * (u(i+1, n-1) - 2*u(i, n-1) + u(i-1, n-1));
        end
    end

    % Plot the solution
    figure;
    surf(x, t, u', 'EdgeColor', 'none');
    xlabel('Distance (x)');
    ylabel('Time (t)');
    zlabel('Temperature (u)');
    title('1D Heat Transfer Solution');
    colorbar;
end
```

2. For the 2D heat transfer problem, we'll use a similar approach:

```
function solve2DHeatTransfer()
    % Parameters
    Lx = 1;             % Length of the plate in x-direction
    Ly = 1;             % Length of the plate in y-direction
    T = 0.1;            % Total simulation time
    alpha = 0.01;       % Thermal diffusivity
    Nx = 50;            % Number of spatial grid points in x-direction
    Ny = 50;            % Number of spatial grid points in y-direction
    Nt = 200;           % Number of time steps
    % Discretization
    dx = Lx / (Nx - 1);
    dy = Ly / (Ny - 1);
    dt = T / Nt;
    x = linspace(0, Lx, Nx);
    y = linspace(0, Ly, Ny);
    t = linspace(0, T, Nt);

    % Initial condition
```

```
    [X, Y] = meshgrid(x, y);
    u0 = sin(pi * X) .* sin(pi * Y);

    % Finite difference scheme
    u = zeros(Nx, Ny, Nt);
    u(:, :, 1) = u0;

   for n = 2:Nt
   for i = 2:Nx-1
   for j = 2:Ny-1
   u(i, j, n) = u(i, j, n-1) + alpha * dt / dx^2 * (u(i+1, j, n-1) - 2*u(i, j, n-1) +   u(i-1, j, n-1)) + ...
  alpha * dt / dy^2 * (u(i, j+1, n-1) - 2*u(i, j, n-1) + u(i, j-1, n-1));
            end
        end
    end

    % Plot the solution
    figure;
    for n = 1:Nt
        surf(x, y, u(:, :, n)', 'EdgeColor', 'none');
        xlabel('Distance (x)');
        ylabel('Distance (y)');
        zlabel('Temperature (u)');
        title(['2D Heat Transfer Solution at Time = ', num2str(t(n))]);
        colorbar;
        pause(0.1);  % Pause to visualize each time step
    end

end
```

RESULT:



1D Heat Transfer Solution
Temperature (u)
1
0.8
0.6
0.4
0.2
0
0.1
0.05
0
Time (t)
0
0.5
1
Distance (x)
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

EXP NO: 6 **DATE:**	**SOLVING OF RADIATION EQUATION BY USING MATLAB**

Aim:

To determine the Radiation heat transfer Equation by using MATLAB

Software Used: MATLAB

Procedure:

1. Open MATLAB on your computer. It may vary take a moment to initialize.
2. Click on the new script to start coding in the MATLAB toolbar.
3. In the script, write your MATLAB code for the respective problem.
4. Apply the Richardson equation in the coding window, press enter to run the code.
5. Enter all the input values that to be given.
6. The output will be displayed in the command window.

Program:

```
                %calculate the parameters
sigma= input ('enter the value of sigma:');
A=input ('enter the value of surface area:');
T= input ('enter the value of the temperature:');
%calculate the radiation
q= sigma *A *(T^4)
%Display the result
    fprintf('radiation (q)=%2f kW',q);
enter the value of sigma: 5.669*10^-8
enter the value of surface area:75
enter the value of the temperature:20
q = 0.6803
fx radiation (q)=0.680280 kW
```

Results:

The results for the radiation heat transfer has been determined by using MATLAB.