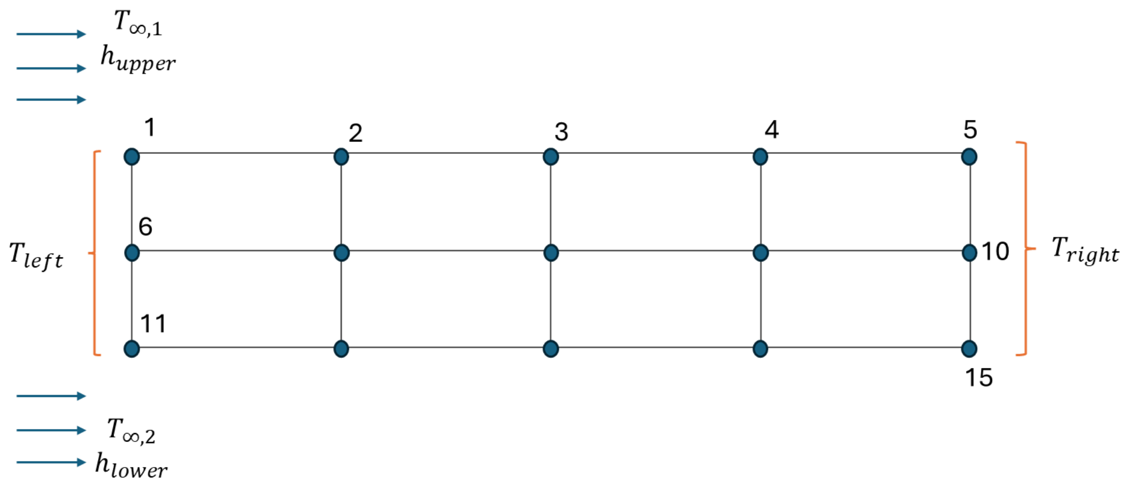
ME351 HEAT TRANSFER

ABILITY IMPROVEMENT APPLICATION INFORMATIVE NOTE

In this application, you will develop a code related to the main mechanisms of heat transfer, i.e., conduction, convection, and constant temperature medium. Utilizing finite difference methodology and numerical methods are required in the nodal analysis.

Please examine the current situation in detail and develop a new code for the given revision tasks.

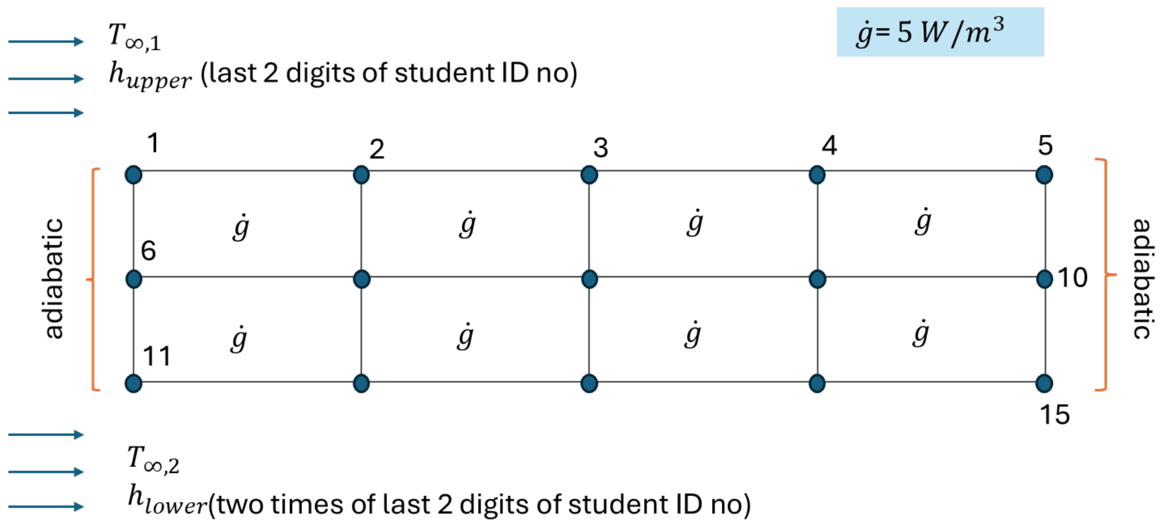
Current situation (Fig.1) within the sent MATLAB code:



In the current situation, the nodal network has 15 nodes experiencing integrated heat transfer modes. The given MATLAB code **(in the attachment)** numerically calculates the finite difference equations and displays the temperature distribution under given initial and boundary conditions.

....

What sort of developments you need to apply within the code (Fig.2):



During this ability improvement application of heat transfer physics, you are required to make revisions and/or develop a new code satisfying the following conditions:

- The left- and right-hand sides of the nodal network are now adiabatic,
- An additional volumetric heat generation of 5 W/m^3 needs to be integrated for each node (energy balance should be revised),
- h_{upper} and h_{lower} shall be updated according to your own student ID's last 2 digits.
- Assume $T_{\infty,1}$ and $T_{\infty,2}$ as 15 °C and 50 °C, respectively.
- Take the thermophysical properties of aluminum if needed.

Attachment: MATLAB code of the current situation (Fig.1).

```
% Steady-State Heat Diffusion in 2D
% No internal heat generation

clc
clear
close all

% Domain size
Lx = 4;     % Length in x-direction [m]
Ly = Lx/2;  % Length in y-direction [m]

% Grid parameters

Nx = Lx+1; % Number of nodes in x-direction
Ny = Ly+1; % Number of nodes in y-direction
dx = Lx / (Nx - 1); % Grid spacing in x
dy = dx; % Grid spacing in y

% Initialize temperature field
T = zeros(Ny, Nx);

% Boundary conditions
T_left = 20;    % Left boundary temperature [°C]
T_right = 20;   % Right boundary temperature [°C]

k = 20;         % Thermal conductivity [W/m·K]
```

```
% Convection parameters
h_upper = 1000;   % Heat transfer coefficient [W/m^2·K]
T_inf_1 = 20;
h_lower = 1000;
T_inf_2 = 20;


% Finite Difference Method (Iterative Solver)
tolerance = 1e-6; % Convergence criterion
max_iterations = 10000; % Maximum number of iterations
error = 1; % Initialize error
iter = 0; % Initialize iteration counter


while error > tolerance && iter < max_iterations

   T_old = T; % Save the old temperature field

   % Top Nodes
   T(Ny, 1) = ( (k/2)*(T_left + T_old(Ny, 2) + T_old(Ny-1, 1)) + h_upper*dx* T_inf_1 /2 ) / 
(h_upper*dx/2 + 3*k/2);
   T(Ny, Nx) = ( (k/2)*(T_old(Ny, Nx-1) + T_right + T_old(Ny-1, Nx)) + h_upper*dx* 
T_inf_1 /2 ) / (3*k/2 + h_upper*dx/2);

   % Bottom Nodes
   T(1, 1) = ( (k/2)*(T_left + T_old(2, 1) + T_old(1, 2)) + h_lower*dx* T_inf_2 /2 ) / 
(h_lower*dx/2 + 3*k/2);
   T(1, Nx) = ( (k/2)*(T_old(1, Nx-1) + T_right + T_old(2, Nx)) + h_lower*dx* T_inf_2 /2 ) 
/ (3*k/2 + h_lower*dx/2);

   for j = 2:Nx-1
      % Top middle nodes
      T(Ny,j) = ((k/2)*(T_old(Ny, j-1) + T_old(Ny, j+1) + T_old(Ny-1, j)) + 
h_upper*dx*T_inf_1) / (2*k + h_upper*dx);
      % Bottom middle nodes
      T(1,j) = ((k/2)*(T_old(1, j-1) + T_old(1, j+1) + T_old(2, j)) + h_lower*dx*T_inf_2) / 
(2*k + h_lower*dx);


end
```

```
    for i = 2:Ny-1
        % Left middle nodes
        T(i,1) = (T_old(i+1, 1) + T_old(i-1, 1) + 2*T_old(i, 1+1) + 2*T_left)/6;
        % Right middle nodes
        T(i,Nx) = (T_old(i+1, Nx) + T_old(i-1, Nx) + 2*T_old(i, Nx-1) + 2*T_right)/6;
    end



    for i = 2:Ny-1
        for j = 2:Nx-1

            T(i, j) = (T_old(i+1, j) + T_old(i-1, j) + T_old(i, j+1) + T_old(i, j-1))/4.0;

        end
    end

    % Compute maximum error
    error = max(max(abs(T - T_old))); % The difference between the iterations
    iter = iter + 1;
end

% Display results
disp(['Converged in ', num2str(iter), ' iterations with error ', num2str(error)]);

% Plot the temperature distribution
[X, Y] = meshgrid(0:dx:Lx, 0:dy:Ly);
contourf(X, Y, T, 20, 'LineColor', 'none');
colorbar;
xlabel('x [m]');
ylabel('y [m]');
title('Steady-State Temperature Distribution');
axis equal
```

…………………………………………………………………………………………………

Duration: 20th of Dec, 2024 until 5:00 pm.

Mandatory Sections in the Report: - A cover page including "ability improvement application", your student number, name-surname etc.

- **Improved code rows and temperature distribution results**
- **A comparison for the current case and the revised scenario.**