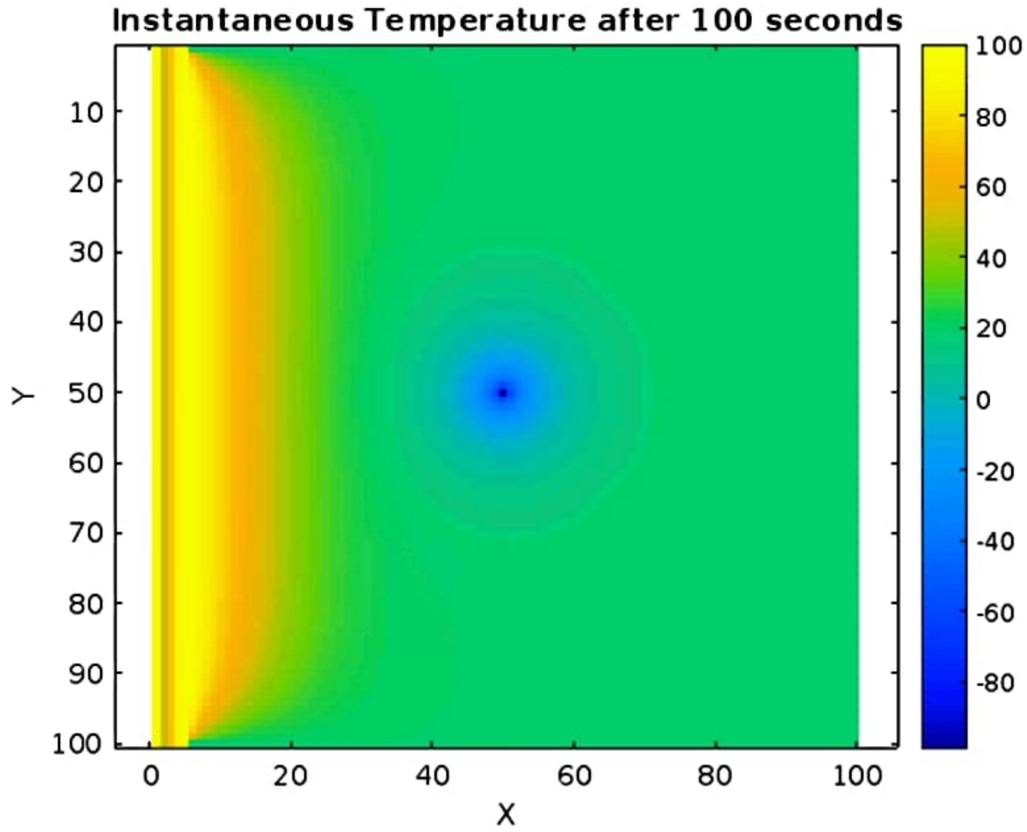
```
% Define parameters
L = 100; % Length of the pentagonal plate (arbitrary units)
N = 100; % Number of grid points in each dimension
dx = L / (N-1); % Grid spacing
dt = 0.01; % Time step
t_final = 100; % Final time
k = 1; % Thermal conductivity
alpha = k * dt / dx^2; % Diffusion coefficient

% Initialize temperature matrix
T = zeros(N, N); % Temperature matrix
T(:, :) = 20; % Initial temperature (arbitrary units)

% Define boundary conditions
boundary_temp = [100, 50, 70, 90, 100]; % Boundary temperatures
center_temp = -100; % Temperature at center

% Main loop for time evolution
for t = 0:dt:t_final
    % Update boundary conditions at every time step
    T(:, 1) = boundary_temp(1); % Wall 1
    T(:, 2) = boundary_temp(2); % Wall 2
    T(:, 3) = boundary_temp(3); % Wall 3
    T(:, 4) = boundary_temp(4); % Wall 4
    T(:, 5) = boundary_temp(5); % Wall 5
    T(floor(N/2), floor(N/2)) = center_temp; % Temperature at center

    % Compute new temperature using explicit finite difference method
    T_new = T;
    for i = 2:N-1
        for j = 2:N-1
            T_new(i, j) = T(i, j) + alpha * (T(i+1, j) + T(i-1, j) + T(i,
j+1) + T(i, j-1) - 4 * T(i, j));
        end
    end

    % Update temperature matrix
    T = T_new;
end

% Display instantaneous temperature after 100 seconds
imagesc(T);
colorbar;
title('Instantaneous Temperature after 100 seconds');
xlabel('X');
ylabel('Y');
axis equal;
```



*Published with MATLAB® R2023b*