**2D MOVING HEAT SOURCE**

**Matlab code :-**

**%steadystatemovingheatsource**

**% Parameters**

**L = 300; % length of the square domain**

**t\_final = 500; % simulation time**

**n = 300; % number of grid points in each direction**

**dx = L/n; % grid spacing**

**dt = 0.05; % time step size**

**k = 5; % thermal conductivity**

**T\_init = 40; % initial temperature**

**T\_source = 100; % temperature of the heat source**

**source\_x = 0; % x-coordinate of the heat source**

**source\_y = L/2; % y-coordinate of the heat source**

**source\_radius = 0.05\*L; % radius of the heat source**

**source\_speed = 3; % speed of the heat source**

**% Initial conditions**

**T = T\_init\*ones(n,n);**

**T\_new = T;**

**% Main loop**

**t = 0;**

**while t < t\_final**

**% Update temperature at interior points**

**for i = 2:n-1**

**for j = 2:n-1**

**T\_new(i,j) = T(i,j) + k\*dt/(dx^2)\*(T(i+1,j) + T(i-1,j) + T(i,j+1) + T(i,j-1) - 4\*T(i,j));**

**end**

**end**

**% Apply Neumann boundary conditions at edges**

**T\_new(1,:) = T\_new(2,:);**

**T\_new(n,:) = T\_new(n-1,:);**

**T\_new(:,1) = T\_new(:,2);**

**T\_new(:,n) = T\_new(:,n-1);**

**% Update temperature at heat source**

**for i = 1:n**

**for j = 1:n**

**dist = sqrt((i\*dx - source\_x)^2 + (j\*dx - source\_y)^2);**

**if dist <= source\_radius**

**T\_new(i,j) = T\_source;**

**end**

**end**

**end**

**% Update heat source position**

**source\_x = source\_x + source\_speed\*dt;**

**% Update time and temperature**

**t = t + dt;**

**T = T\_new;**

**% Plot temperature**

**imagesc(T);**

**colorbar;**

**axis equal tight;**

**title(sprintf('Time: %0.2f', t));**

**drawnow;**

**end**