```
MATLAB Code for solving the PDE for 3D Diffusion in a room with a heater and
window
%How temperature of system evolves with time

%Parameters
rho=1.11;  % Density of air in Bangalore at 25 deg C
cp=1000; % Specific Heat of air is 1000J/kg.K

%Domain and step; Room Dimensions;
Lx=10;
Ly=10;
Lz=10;

% Size of the array
Nx=21; Nt=1000; % Nt number of iterations
Ny=21;
Nz=21;

dx=Lx/(Nx-1);
dy=Ly/(Ny-1);
dz=Lz/(Nz-1);
dt=10; % Duration of each iteration is 10 sec fixed by Courant Rule

% Defining Field Variables
Tn=zeros(Nx,Ny,Nz); % Temperature
x=linspace(0,Lx,Nx); % Nx elements going from 0 to Lx
y=linspace(0,Ly,Ny); % Ny elements going from 0 to Ly
z=linspace(0,Lz,Nz); % Nz elements going from 0 to Lz

[X,Y,Z]= meshgrid(x,y,z); % For plotting

%Thermal Conductivity (Non uniform at the edges)
% Room is insulated, hence defining low thermal conductivity AT THE 6
enclosing surfaces
% Heat can go past the wall only at a very small rate

% Thermal conductivity as a function of space
K=ones(Nx,Ny,Nz) + 0.03;
K([1 end],:,:) = 0.001;
K(:,[1 end],:)= 0.001;
K(:,:,[1 end])= 0.001;

% Dirichlet Initial Conditions
Tn(:,:,:)=25;  % Initial Temperature, 3D matrix Tn
t=0;

for n=1:Nt
    Tc=Tn; % Saving the temperature to use in the central difference
technique
    t=t+dt; % New time


    % Finding the new temperature: Numerical Implementation of the 3D
    % Diffusion Equation
    for i=2:Nx-1
```

```
        for j=2:Ny-1
            for k=2:Nz-1

                % Using finite difference: Central Difference Technique
                Tn(i,j,k)= Tc(i,j,k) + ...
                dt * (K(i,j,k)/rho/cp) * ...
                ((Tc(i+1,j,k)- 2*Tc(i,j,k) + Tc(i-1,j,k))/dx/dx + ...
               % Second derivative wrt x-axis

                (Tc(i,j+1,k)- 2*Tc(i,j,k) + Tc(i,j-1,k))/dy/dy + ...
               % Second derivative wrt y-axis

                (Tc(i,j,k+1)- 2*Tc(i,j,k) + Tc(i,j,k-1))/dz/dz);
               % Second derivative wrt z-axis

            end
        end
    end

    Tbar= mean(Tn(:)); % Average Temperature, will be used for display

    % Heater source at the middle of room at a small height above the floor

    if(t<1800)  % Max. Time for which heater can continuously run (say 30
min)
       Tn(10,10,2)= Tn(10,10,2) + dt* 1300/rho/cp;  % 10,10,5 changed to 1300

    end

    % 3D Neumann Boundary Conditions

    Tn(1,:,:)= Tn(2,:,:);  % x-axis, left wall
    Tn(end,:,:)=Tn(end-1,:,:); % x-axis, right wall
    Tn(:,1,:)=Tn(:,2,:); % y-axis, front wall
    Tn(:,end,:)= Tn(:,end-1,:); % y-axis, back wall
    Tn(:,:,1)=Tn(:,:,2); % z-axis, floor
    Tn(:,:,end)= Tn(:,:,end-1); % z-axis, roof

    % Hence the room is insulated at all 6 sides

    % Window (SINK): conditions

    if(t>3600) % Window opening condition (Here, after 1 hr window opens,
assume)
        Tn(end,9:11,9:11)=20;
% Sink as a window at constant temp 20 deg C
    end
    %9:11 is the set of pixels which resembles the window of the room

    % Since the time scale is in hours, we can display at certain time
    % intervals like multiples of 10 minutes, an if condition is used

    if(mod(t,600)==0) % To plot only at time intervals of 10 min
        clf;
```

```
        subplot(2,1,1); % To show 2 plots simultaneously
        slice(X,Y,Z,Tn,5,5,2); % Displays change in temperature on a slice of
        %screen and 5,5,2 coordinates arbitrarily chosen ;

        colorbar; % To see how the temperature is varying
        axis([0 Lx 0 Ly 0 Lz]);
        title(sprintf('Average Temperature= %.2f Time= %f minutes', Tbar,
t/60))
        view(-75,15);

        % Gradient (To obtain and display the heat vector field)

        %Heat Flux (qx,qy,qz) in the 3 directions
        [Tx,Ty,Tz]= gradient(Tn);
        qx= -K.*Tx;  % .* is used to perform elementwise multiplication
        qy= -K.*Ty;
        qz= -K.*Tz;

        % qx,qy and qz form the vector field

        subplot(2,1,2); % 2nd plot
        % To see Temperature stream lines along 2 walls and the floor

        % The values 9.9 and 0.1 so as to view stream lines just near the
        % boundaries
        streamslice(X,Y,Z,qx,qy,qz,9.8,9.8,0.2);

        axis([0 Lx 0 Ly 0 Lz]);

        hold on;

        % To get 3x3x3 grid points from where stream lines will start
        [sx, sy, sz]= meshgrid([2 5 8], [2 5 8], [2 5 8]);

        % Using handle to set the colour of streamlines

        h= streamline(X,Y,Z,qx,qy,qz,sx,sy,sz);
        set(h,'color', 'red');
        hold off;

        view(-75,15);
        pause(0.01);
    end
end

% When window is open, heat energy flows towards the window
% When the heat source stops, we see all the heat energy in room starts
flowing towards the window

% In this code case, I have set heater on for 30min, then
% and window opens after 1 hr from start. Hence there is a period of 30 min
% between heater off and window open. This is when, heat in the room
% spreads and the heat flux is visually appealing to look at in the
% figure/movie generated
```