1. Assumptions:
   a. Two-dimensional heat transfer
   b. Steady state
   c. No internal heat generation

In cartesian coordinate system, the 2-D heat conduction equation is given by

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

The finite difference form of 2-D heat conduction equation can be written as

$$\frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{\Delta x^2} + \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta y^2} = 0$$

$$T_{i,j} = \frac{(T_{i+1,j} + T_{i-1,j}) + \left(\frac{\Delta x}{\Delta y}\right)^2 (T_{i,j+1} + T_{i,j-1})}{2[1 + \left(\frac{\Delta x}{\Delta y}\right)^2]}$$

```
%% MATLAB Program to generate the 2D temperature distribution at steady   state:

clear all;
close all;
clc;

%% Initial Conditions:
x=1;                % length on X-Axis
y=.5;               % height on Y-Axis
dx= x/3;            % grid size in x-direction
dy=y/3;             % grid size in y-direction
nx= 1+x/dx;         % nodes in x-direction
ny= 1+y/dy;         % nodes in y-direction

%for ease of calculation
A=(dy^2)/(2*(dx^2+dy^2));
B=(dx^2)/(2*(dx^2+dy^2));

%% Initializing the temperature matrix:
Ttop=100;
Tbottom=300;
Tright=200;
Tleft=50;

for i=1:nx
    for j=1:ny
    T(i,ny)=Tright;
    T(i,1)=Tleft;
    T(nx,j)=Tbottom;
    T(1,j)=Ttop;
    end
```

```
end
T(2:nx-1,2:ny-1)=(Ttop+Tright+Tbottom+Tleft)/4; %Taking initial guess of all notes within boundaries

% Defining temp at corner as mean of two sides
T(1,1)= 0.5*(Tleft+Ttop);
T(1,nx)= 0.5*(Tright+Ttop);
T(ny,1)= 0.5*(Tleft+Tbottom);
T(nx,ny)= 0.5*(Tright+Tbottom);

iter=0;
r= 10;

%% Gauss Siedel Iteration Scheme:
while r > 0.001 % error
    for i=2:nx-1
        for j=2:ny-1
        T_old(i,j)=T(i,j);
        T(i,j)=B*(T(i+1,j)+T(i-1,j))+ A*(T(i,j+1)+T(i,j-1));
        error(i,j)=(T_old(i,j)-T(i,j));
        end
    end

rsq=0;
for i=2:nx-1
        for j=2:ny-1
        rsq=(rsq+(error(i,j))^2);
        end
end
r=sqrt(rsq);
iter=iter+1;
end
%Results
 T1 =T(2,2)
 T2 =T(2,3)
 T3 =T(3,2)
 T4 =T(3,3)
 Tmid=(T1+T2+T3+T4)/4


After iteration, Nodes temperature:

                     75.0000    100.0000   100.0000   150.0000
                     50.0000    143.5163   164.9450   200.0000
                     50.0000    205.0549   226.4835   200.0000
                    175.0000    300.0000   300.0000   250.0000
```

Results:

T1	T2	T3	T4	Mid-Point
143.5163	164.9450	205.0549	226.4835	184.9999

(a) On Reducing the mesh size by factor of 2.

```
%% MATLAB Program to generate the 2D temperature distribution at steady   state:
```

```
clear all;
close all;
clc;

%% Initial Conditions:
x=1;             % length on X-Axis
y=.5;            % height on Y-Axis
dx= x/6;         % grid size in x-direction
dy=y/6;          % grid size in y-direction
nx= 1+x/dx;      % nodes in x-direction
ny= 1+y/dy;      % nodes in y-direction

%for ease of calculation
A=(dy^2)/(2*(dx^2+dy^2));
B=(dx^2)/(2*(dx^2+dy^2));

%% Initializing the temperature matrix:
Ttop=100;
Tbottom=300;
Tright=200;
Tleft=50;

for i=1:nx
    for j=1:ny
    T(i,ny)=Tright;
    T(i,1)=Tleft;
    T(nx,j)=Tbottom;
    T(1,j)=Ttop;
    end
end
T(2:nx-1,2:ny-1)=(Ttop+Tright+Tbottom+Tleft)/4; %Taking initial guess of all notes
within boundaries

% Defining temp at corner as mean of two sides
T(1,1)= 0.5*(Tleft+Ttop);
T(1,nx)= 0.5*(Tright+Ttop);
T(ny,1)= 0.5*(Tleft+Tbottom);
T(nx,ny)= 0.5*(Tright+Tbottom);

iter=0;
r= 10;

%% Gauss Siedel Iteration Scheme:
while r > 0.001 % error
    for i=2:nx-1
        for j=2:ny-1
        T_old(i,j)=T(i,j);
        T(i,j)=B*(T(i+1,j)+T(i-1,j))+ A*(T(i,j+1)+T(i,j-1));
        error(i,j)=(T_old(i,j)-T(i,j));
        end
    end

rsq=0;
for i=2:nx-1
        for j=2:ny-1
        rsq=(rsq+(error(i,j))^2);
        end
```

```
end
r=sqrt(rsq);
iter=iter+1;
end
%Results
 T1 =T(2,2)
 T2 =T(2,3)
 T3 =T(3,2)
 T4 =T(3,3)
 Tmid=(T1+T2+T3+T4)/4

 disp(T)
```

```
After iterations, nodes temperature:
   75.0000  100.0000  100.0000  100.0000  100.0000  100.0000  150.0000
   50.0000  103.1339  121.6220  129.0778  133.2106  142.5557  200.0000
   50.0000  114.9294  146.0024  158.9868  165.1185  173.0866  200.0000
   50.0000  135.1894  174.9054  190.6092  196.5674  198.8813  200.0000
   50.0000  166.8178  209.8117  224.6682  228.9277  224.9749  200.0000
   50.0000  216.9024  251.7526  261.3766  263.3412  256.3241  200.0000
  175.0000  300.0000  300.0000  300.0000  300.0000  300.0000  250.0000
```

Results:

Mesh	T1	T2	T3	T4	Mid-Point
Fine	146.0024	165.1185	209.8117	228.9277	190.6092
Coarse	143.5163	164.9450	205.0549	226.4835	184.9999

```
%Matlab Coding for Analytical method

clear all;
clc;

L=1;
W=0.5;

T1=0;
f_t=0;
f_s=0;

%Node Mid-Point

for      x=L/2;
         y=W/2;

         for n=1:10;

              f_t = f_t+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*x/L))*(sinh(n*(pi)*y/L))/sinh(n*(pi)*W/L);
              f_s = f_s+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*y/W))*(sinh(n*(pi)*x/W))/sinh(n*(pi)*L/W);
              ft=(2/pi)*f_t;
              fs=(2/pi)*f_s;
```

```
             for       T2=50;
            T50 = fs*(T2-T1)+T1;

            for       T2=100;
            T100 = ft*(T2-T1)+T1;
            end

             for       T2=200;
            T200 = fs*(T2-T1)+T1;
             end

            for       T2=300;
            T300 = ft*(T2-T1)+T1;
            end


            end


        end

end

 Tcentre = T50+ T100+ T200 +T300              %superposition


% Node 1
f_t1=0;
f_s1=0;

for x=L/3;
    y=W/3;
    for n=1:10;

            f_t1 = f_t1+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*x/L))*(sinh(n*(pi)*y/L))/sinh(n*(pi)*W/L);
            f_s1 = f_s1+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*y/W))*(sinh(n*(pi)*x/W))/sinh(n*(pi)*L/W);
            ft1=(2/pi)*f_t1;
            fs1=(2/pi)*f_s1;

              for       T2=50;
            TN150 = fs1*(T2-T1)+T1;
            end

            for       T2=100;
            TN1100 = ft1*(T2-T1)+T1;
            end

            for       T2=200;
            TN1200 = fs1*(T2-T1)+T1 ;
            end
```

```
            for      T2=300;
            TN1300 = ft1*(T2-T1)+T1;
            end

    end

end
 TNode1 = TN150+ TN1100+ TN1200 +TN1300            %superposition


 %Node 2

f_t2=0;
f_s2=0;

for x=2*L/3;
    y=W/3;

    for n=1:10;

            f_t2 = f_t2+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*x/L))*(sinh(n*(pi)*y/L))/sinh(n*(pi)*W/L);
            ft2=(2/pi)*f_t2;

            f_s2 = f_s2+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*y/W))*(sinh(n*(pi)*x/W))/sinh(n*(pi)*L/W);
            fs2=(2/pi)*f_s2;

            for      T2=50;
            TN250 = fs2*(T2-T1)+T1;
            end

            for      T2=100;
            TN2100 = ft2*(T2-T1)+T1;
            end

            for      T2=200;
            TN2200 = fs2*(T2-T1)+T1 ;
            end

            for      T2=300;
            TN2300 = ft2*(T2-T1)+T1;
            end
    end

end
 TNode2 = TN250+ TN2100+ TN2200 +TN2300        %superposition


 %Node 3

f_t3=0;
f_s3=0;
```

```
for x=L/3;
    y=2*W/3;

    for n=1:10;

            f_t3 = f_t3+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*x/L))*(sinh(n*(pi)*y/L))/sinh(n*(pi)*W/L);
            ft3=(2/pi)*f_t3;

            f_s3 = f_s3+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*y/W))*(sinh(n*(pi)*x/W))/sinh(n*(pi)*L/W);
            fs3=(2/pi)*f_s3;

            for     T2=50;
            TN350 = fs3*(T2-T1)+T1;
            end

            for     T2=100;
            TN3100 = ft3*(T2-T1)+T1;
            end

            for     T2=200;
            TN3200 = fs3*(T2-T1)+T1 ;
            end

            for     T2=300;
            TN3300 = ft3*(T2-T1)+T1;
            end
    end

end
 TNode3 = TN350+ TN3100+ TN3200 +TN3300              %superposition

  %Node 4
f_t4=0;
f_s4=0;

for x=2*L/3;
    y=2*W/3;

    for n=1:10;

            f_t4 = f_t4+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*x/L))*(sinh(n*(pi)*y/L))/sinh(n*(pi)*W/L);
            ft4=(2/pi)*f_t4;

            f_s4 = f_s4+((((-
1)^(n+1))+1)/n)*(sin(n*(pi)*y/W))*(sinh(n*(pi)*x/W))/sinh(n*(pi)*L/W);
            fs4=(2/pi)*f_s4;

            for     T2=50;
            TN450 = fs4*(T2-T1)+T1;
            end
```

```
            for       T2=100;
            TN4100 = ft4*(T2-T1)+T1;
            end

            for       T2=200;
            TN4200 = fs4*(T2-T1)+T1 ;
            end

            for       T2=300;
            TN4300 = ft4*(T2-T1)+T1;
            end
    end

end
 TNode4 = TN450+ TN4100+ TN4200 +TN4300                  %superposition
```

Results:

Mesh	T1	T2	T3	T4	Mid-Point
Fine	146.0024	165.1185	209.8117	228.9277	190.6092
Coarse	143.5163	164.9450	205.0549	226.4835	184.9999

Analytical Solution (using superposition)					
Analytical	108.6920	138.5124	238.7467	268.5671	191.7742

(c)

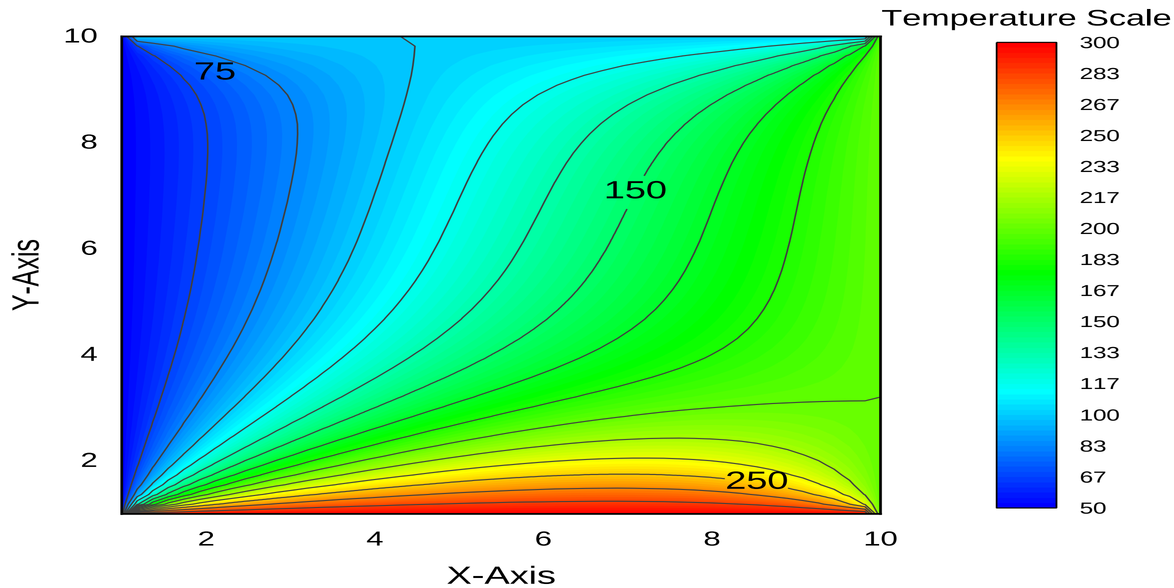


Figure 1. Isotherms for 75, 150 and 250℃ (Result from fine mesh grid)

2. Assumptions:

a. Two-dimensional heat transfer
b. No internal heat generation

In certain coordinate system, the 2-D heat conduction equation is given by

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t}$$

The finite difference form of 2-D heat conduction equation can be written as

$$\frac{T^n_{i+1,j} - 2T^n_{i,j} + T^n_{i-1,j}}{(\Delta x)^2} + \frac{T^n_{i,j+1} - 2T^n_{i,j} + T^n_{i,j-1}}{(\Delta y)^2} = \frac{1}{\alpha}\frac{T^{n+1}_{i,j} - T^n_{i,j}}{\Delta t}$$

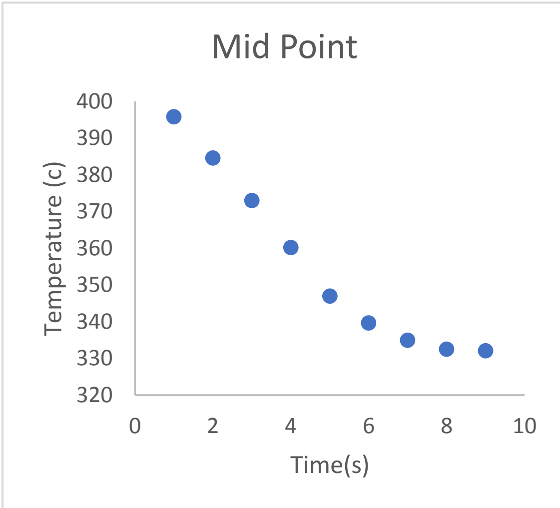
```
%Matlab coding for two-dimensional unsteady state conduction
%----------------------------------------------------------
clear all
close all
clc

X = 1;                          % length on X-axis (m)
Y = 0.5;                        % length on Y-axis (m)
nx =4;                          % number of points along x
ny = nx;                        % number of along x & y are same
dt = 1e-2;                      % time step size
x = linspace(0,X,nx);           % x nodes
y = linspace(0,Y,ny);           % y nodes
dx = x(2)-x(1);                 % gird size along x
dy = y(2)-y(1);                 % grid size along y
alpha = 1.4;                    % thermal diffusivity
[xx yy] = meshgrid(x,y);

%Initialization
T = 400*ones(nx, ny);

% Boundary conditions
T(1,1:end) = 300;               % bottom boundary condition
T(end,1:end) = 100;             % top boundary condition
T(1:end,1) = 50;                % left boundary condition
T(1:end,end) = 200;             % right boundary condition

% creating a copy of T
Told = T;
T_prev_dt = T;
tol = 1e-4;                    % tolerance limit
error = 9e9;                   % error value set
GS_iter=0;

% for ease of calculation
```

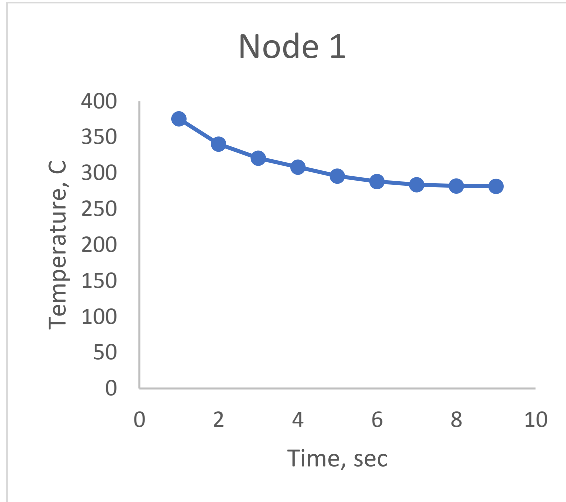
```
k1 = alpha*dt/dx^2;
k2 = alpha*dt/dy^2;
term1 = (1+(2*k1)+(2*k2));

% Gauss Seidel method
tic
while(error>tol)
       for i = 2:(nx-1)
            for j = 2:(ny-1)
                h = T(i-1,j)+Told(i+1,j);
                v = T(i,j-1)+Told(i,j+1);
                T(i,j)= ((T_prev_dt(i,j) +h*k1 +v*k2)/term1);
            end
        end
    error=max(max(abs(Told-T)));
    GS_iter=GS_iter+1;
% updating old values
    Told = T
% creating a contour plot
[C,d]=contourf(xx,yy,T); % to fill contour plot
clabel(C,d);
xlabel('X axis');
ylabel('Y axis');
pause(1)
end
T_prev_dt = T
GS_iteration_time = toc
```

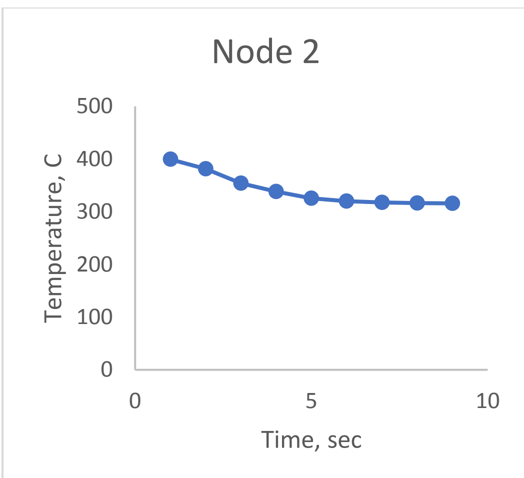
Results:

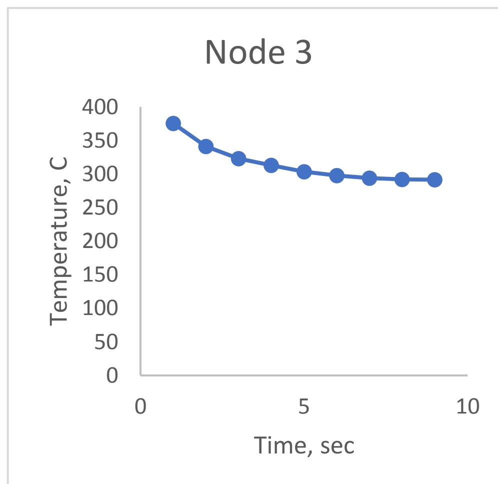
	T_1	T_2	T_3	T_4	$T_{mid\text{-}point}$
Unsteady state	278.4852	307.1432	291.8752	320.5332	299.5092
Steady state (Coarse Mesh)	143.5163	164.9450	205.0549	226.4835	184.9999

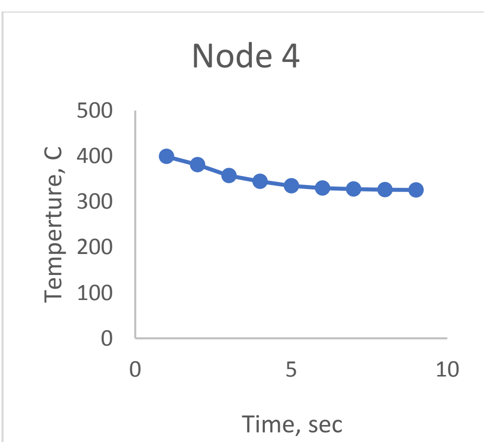
Temperature Evolution



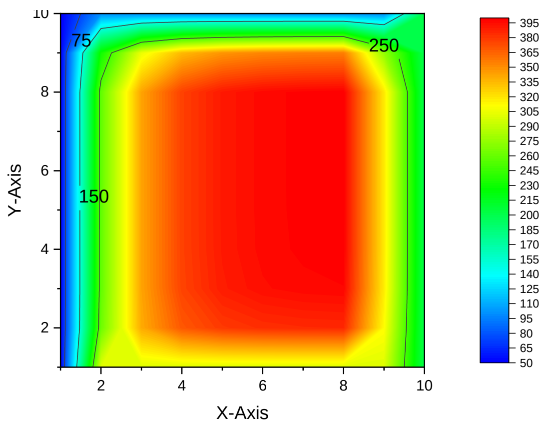




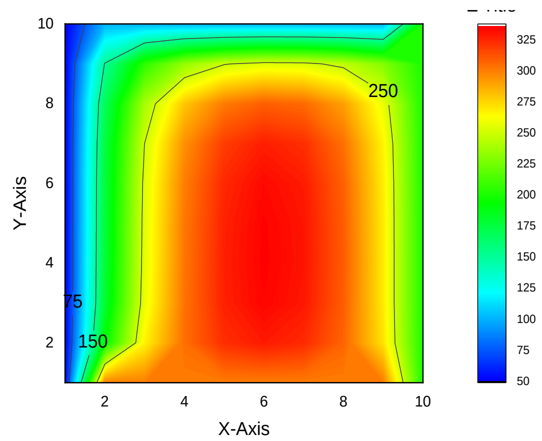




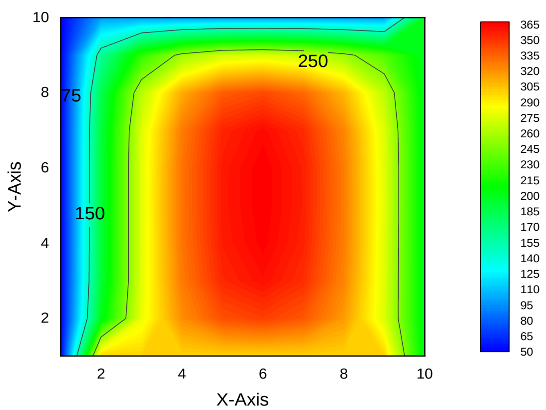
Isotherms:



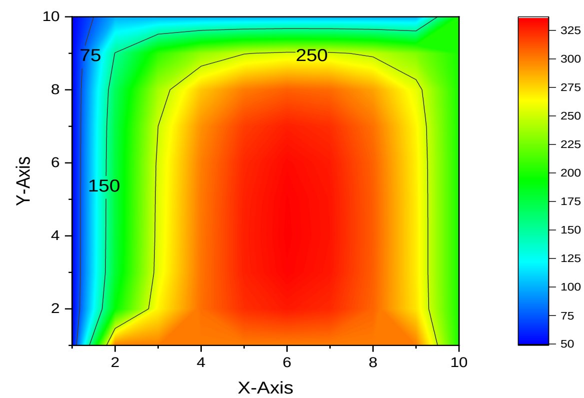
t = 1 sec



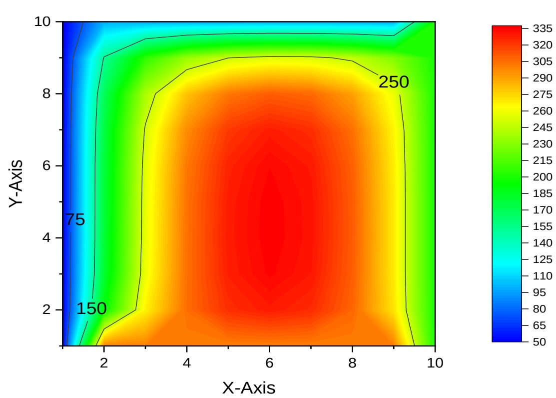
t =30 sec



t = 5 sec



t = 46 sec



t =15 sec

Figure 2. Isotherms at 75, 150 and 250°C (result from fine mesh grid)