Heat Transfer

Newton's Law of Cooling:-

If the temperature of the body is T and T> Ts (Surrounding Temperature) then it continuously loses heat energy and its temperature goes on decreasing with time.

"The rate of loss of heat by a body and hence the rate of decrease of its temperature is directly proportional to the difference of temperatures of the body and its surrounding."

The body of mass m and specific heat c then, the heat required to change the temperature T is,

$$dQ = mc\ dT$$

Therefore the rate of loss of heat:-

$$\frac{dQ}{dt} = -mc\frac{dT}{dt}$$

$$\frac{dT}{dt}\alpha\ (T - Ts) \qquad (\therefore \text{By the Newton's law of cooling})$$

$$\frac{dT}{dt} = -k'(T - Ts)$$

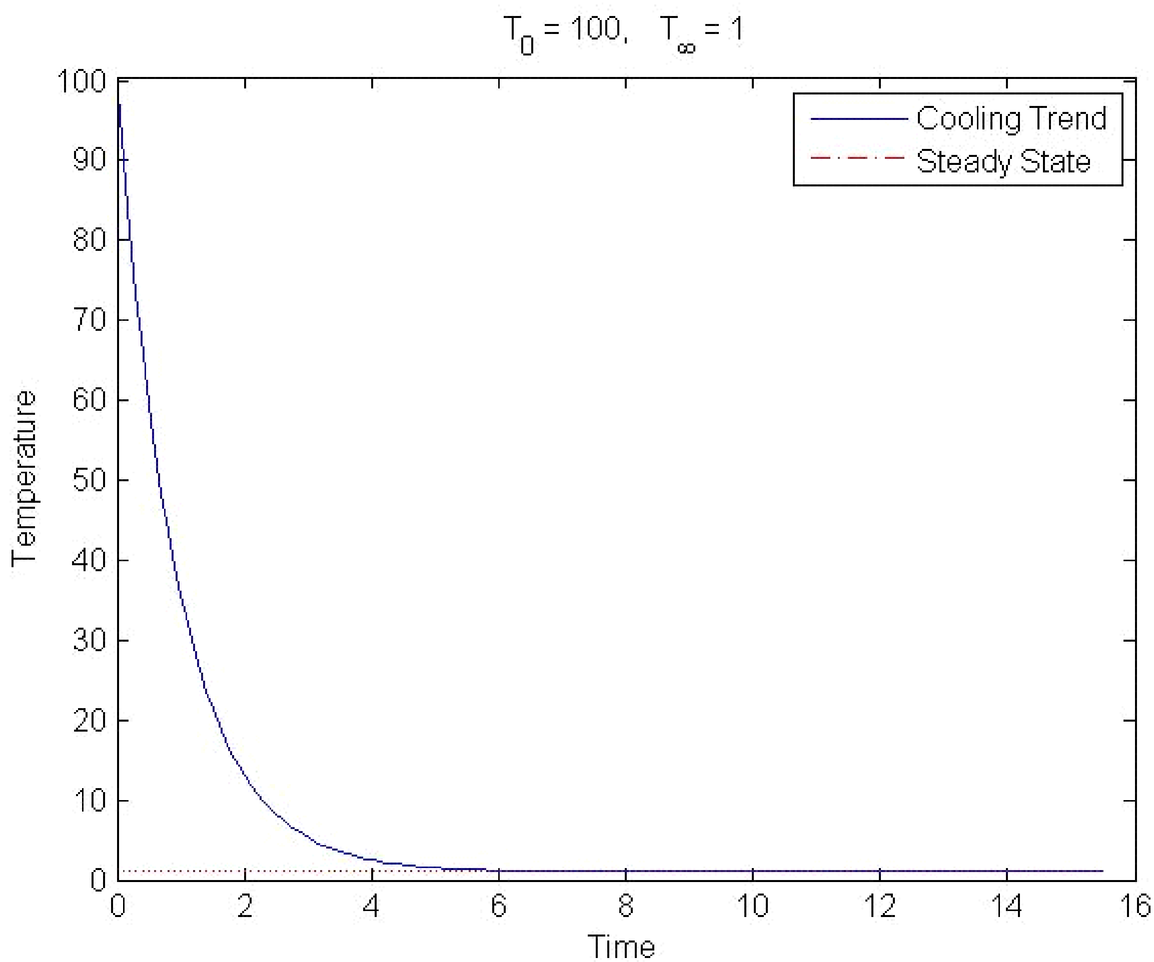
$$\frac{T_{i+1} - T_i}{dt} = k(T_i - T_\infty), \quad T(0) = T_0$$

MATLAB Code:-

```
%Newton Cooling Law
clear; close all; clc;
h = 1;
T(1) = 100; %T(0)
error = 1;
TOL = 1e-6;
k = 0;
dt = 1/10;
```

```
while error > TOL,
   k = k+1;
   T(k+1) = h*(1-T(k))*dt+T(k);
    error = abs(T(k+1)-T(k));
end
t = linspace(0,dt*(k+1),k+1);
plot(t,T),hold on, plot(t,1,'r-.')
xlabel('Time'),ylabel('Temperature'),
title(['T_0 = ',num2str(T(1)), ',    T_\infty = 1']),
legend('Cooling Trend','Steady State')
ylim([0 T(1)+0.5]);
```

Output:-



Stefan-Boltzmann's law:-

"The amount of radiant energy emitted by a surface per unit area per unit time is directly propositional to forth power of its absolute temperature."

W = $\sigma e \, T^4$

Emissivity (e):-

Emissivity is the measure of an object's ability to emit infrared energy. Emitted energy indicates the temperature of the object. Emissivity can have a value from 0 (shiny mirror) to **1.0** (blackbody). Most organic, painted, or oxidized surfaces have emissivity values close to **0.95**.

W → Total Emissive Power

T → Absolute Temperature

σ → Stefan-Boltzmann Constant (= 5.6 x 10^{-8} W$m^{-2}K^{-4}$)

If the temperature of the body is T and T> Ts (Surrounding Temperature) then it continuously loses heat energy and its temperature goes on decreasing with time.

$$\frac{dQ}{dt} = -\sigma e A(T^4 - Ts^4)$$

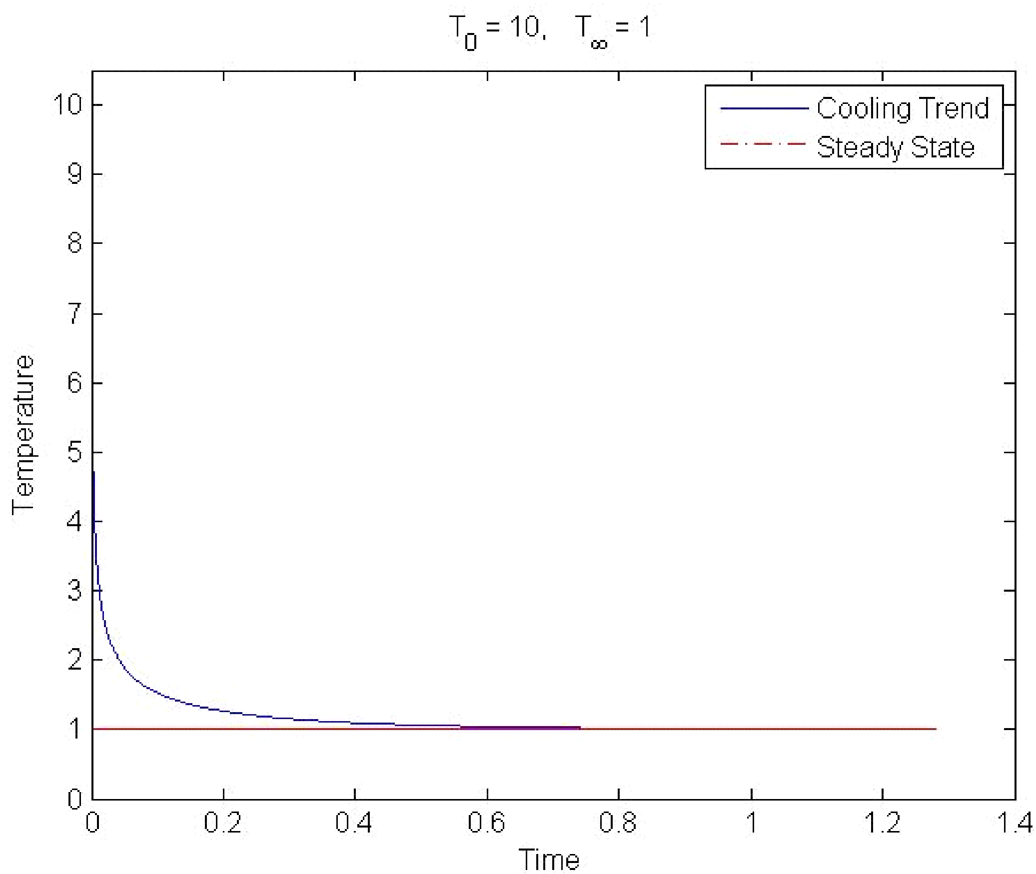
$$\frac{T_{i+1} - T_i}{dt} = \sigma\epsilon(T_i^4 - T_\infty^4), \quad T(0) = T_0$$

MATLAB Code:-

```
%Boltzman Cooling Law
clear; close all; clc;
h = 1;
T(1) = 10; %T(0)
error = 1;
TOL = 1e-6;
```

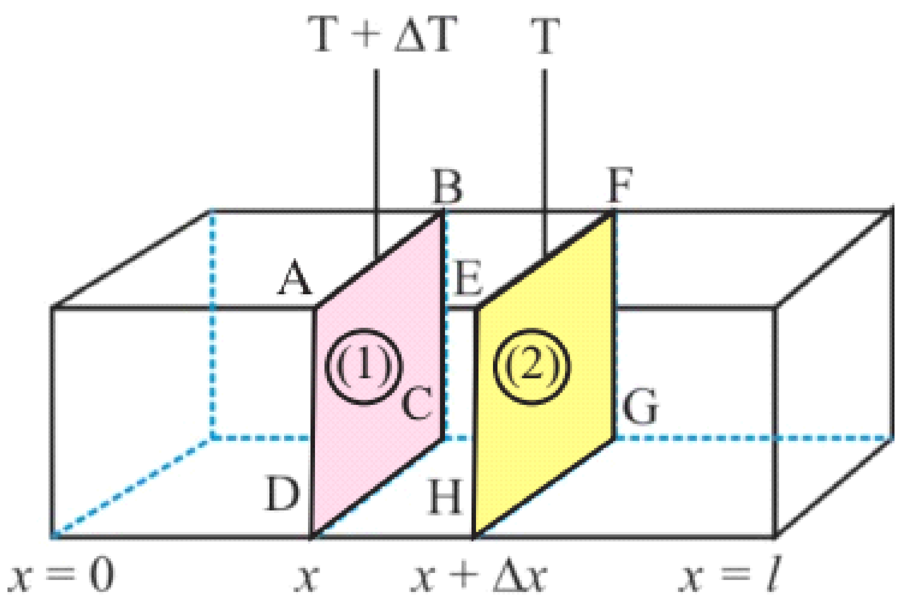
```
k = 0;
dt = 1/10000;
while error > TOL,
   k = k+1;
   T(k+1)  = h*(1-(T(k))^4)*dt+T(k);
   error = abs(T(k+1)-T(k));
end
t = linspace(0,dt*(k+1),k+1);

plot(t,T),hold on, plot(t,1,'r-.')
xlabel('Time'),ylabel('Temperature'),
title(['T_0 = ',num2str(T(1)), ',     T_\infty = 1']),
legend('Cooling Trend','Steady State')
ylim([0 T[1]]);
```

Output:-



Heat (Thermal) Conduction:-

The flow of energy between the adjacent parts of the body due to temperature difference between them is called thermal conduction or heat Conduction.



Assume heat transfer takes place only perpendicular to the phases x=0 & x=*l* and other phases are isolated such that energy cannot cross into or out of this block - our system of interest - as heat.

Applying the First Law of Thermodynamics on this block,

$$\delta Q - \delta W = dE$$

Q → Heat Transfer through the system

W→ Work done by the surrounding on the system

E → Internal Energy of the System

$$q - w = \frac{dE}{dt}$$

We have assume that there is no interaction between System & Surrounding such that w=0 in above formula.

Heat transfer interaction take place across the surface area A of the block, one at location x and the other one at (x + dx) is,

$$\Delta Q = \Delta E$$

And also we can say that the total Energy of the block depends only on its internal energy.

$$dE = \rho(A\, dx)u$$

$$u = c\, dT$$

Where u→ Specific (per unit mass) internal Energy of the block

c → Specific Heat (per unit mass).

For the rate of change of Total Energy,

$$\frac{dE}{dt} = \rho(A\, dx)c\, \frac{dT}{dt}$$

Fourier law of Heat conduction:-

Fourier Law of heat conduction proposes how heat transfers in a solid body.

According to Joseph Fourier Heat transfer is directly propositional to Temperature gradient$\left(\frac{dT}{dx}\right)$.

$$\Delta Q\ \alpha - \frac{dT}{dx}$$

$$\frac{dQ}{dt} = -kA\frac{dT}{dx}$$

Where k → Thermal Conductivity (Property of the Material)
A → Cross-Section Area of the block

$$\frac{dE}{dt} = -kA\frac{dT}{dx}$$

$$\frac{dE}{dt} = \int kA\ \frac{d^2T}{dx^2}dx$$

$$\rho(A\ dx)c\ \frac{dT}{dt} = kA\ \frac{d^2T}{dx^2}dx$$

$$\frac{1}{\alpha}\frac{dT}{dt} = \frac{d^2T}{dx^2}$$

Where $\alpha \rightarrow \frac{k}{\rho c}$ (Constant For particular Material)

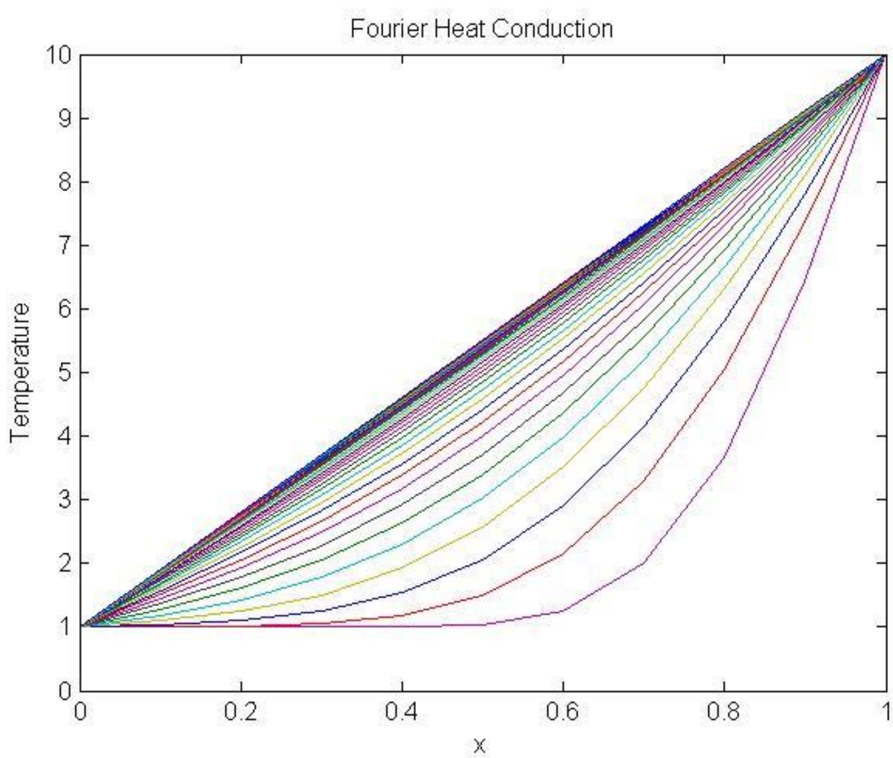
$$\frac{T_{i,n+1} - T_{i,n}}{dt} = \frac{T_{i+1,n} - 2T_{i,n} + T_{i-1,n}}{dx^2}, \quad T(x,0) = 1, T(0,t) = 1, T(1,t) = 10$$

MATLAB Code:-

```
%Fourier Heat conduction
clear; close all; clc;
```

```
alpha = 1;
n = 11;
T = ones(n,1); Told = T;
T(1) = 1; %Left boundary
T(n) = 10; %Right boundary
x = linspace(0,1,n);
dx = x(2)-x(1);
dt = dx^2/3; %cfl condition

error = 1;
TOL = 1e-6;
k = 0;
while error > TOL,
   Told = T;
   k = k+1;
   for i = 2:n-1
    T(i) = ((dt*(Told(i+1)-2*Told(i)+Told(i-1)))*alpha/dx^2) + Told(i);
   end
   error = max(abs(T-Told));
   if mod(k,5)==0, out(k,:) = T; end
end

plot(x,out)
xlabel('x'),ylabel('Temperature'),
title(['Fourier Heat Conduction']),
%legend('Cooling Trend','Steady State')
```

Output:-



Heat Conduction in 2-Dimension:-

$$\frac{1}{\alpha}\frac{dT}{dt} = \frac{d^2T}{dx^2} + \frac{d^2T}{dy^2}$$

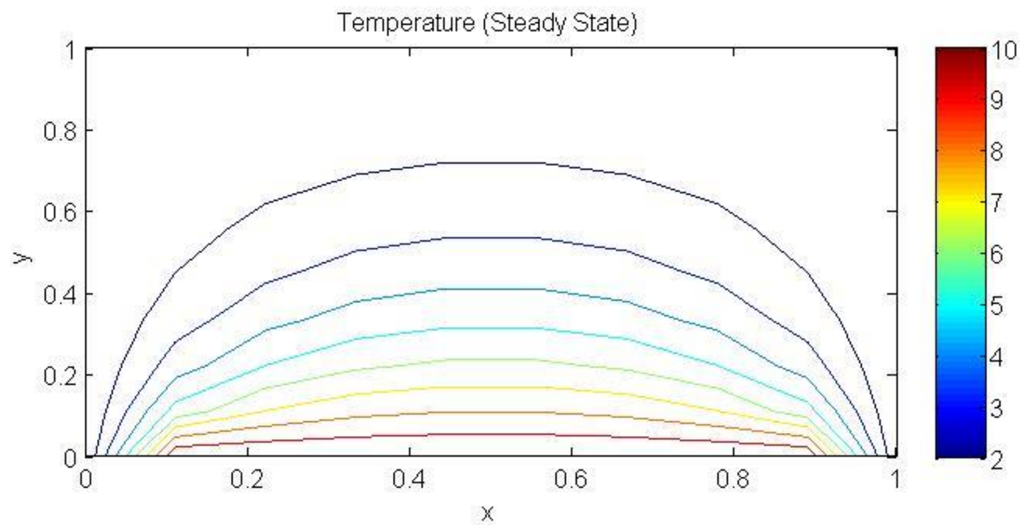
Where $\alpha \rightarrow \frac{k}{\rho c}$ (Constant For particular Material)

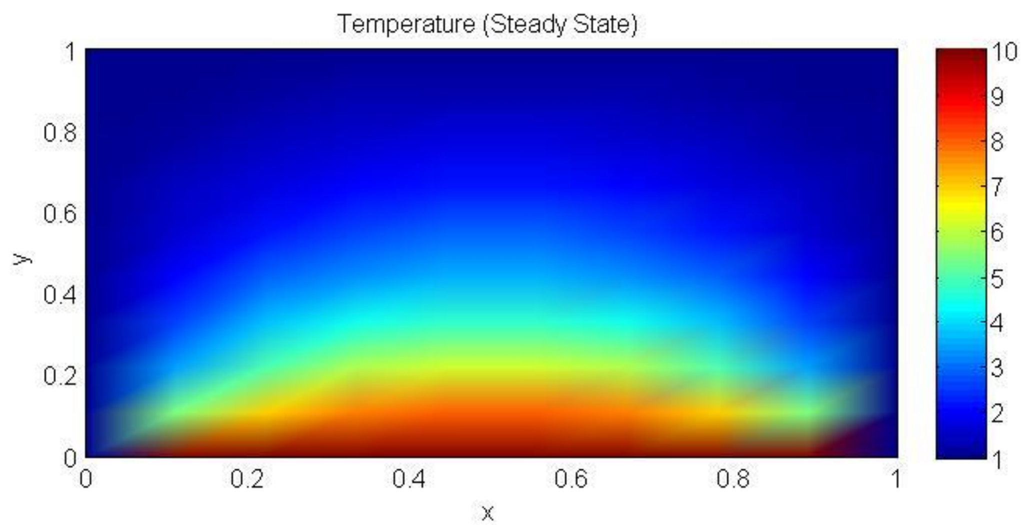
$$\frac{T_{i,n+1} - T_{i,n}}{dt} = k\left(\frac{T_{i+1,n} - 2T_{i,n} + T_{i-1,n}}{dx^2} + \frac{T_{i+1,n} - 2T_{i,n} + T_{i-1,n}}{dy^2}\right)$$

MATLAB Code:-

```
%2D Heat Equation.
clear; close all; clc
n = 10;  %grid has n - 2 interior points per dimension (overlapping)
x = linspace(0,1,n); dx = x(2)-x(1); y = x; dy = dx;
TOL = 1e-6;
T = zeros(n);
T(1,1:n) = 10; %TOP
T(n,1:n) = 1;  %BOTTOM
T(1:n,1) = 1;  %LEFT
T(1:n,n) = 1;  %RIGHT
dt = dx^2/4;
error = 1; k = 0;
    while error > TOL
        k = k+1;
          Told = T;
          for i = 2:n-1
              for j = 2:n-1
              T(i,j) = dt*((Told(i+1,j)-2*Told(i,j)+Told(i-1,j))/dx^2 ...
                      + (Told(i,j+1)-2*Told(i,j)+Told(i,j-1))/dy^2) ...
                      + Told(i,j);
              end
          end
          error = max(max(abs(Told-T)));
    end
subplot(2,1,1),contour(x,y,T),
title('Temperature (Steady State)'),xlabel('x'),ylabel('y'),colorbar
subplot(2,1,2),pcolor(x,y,T),shading interp,
title('Temperature (Steady State)'),xlabel('x'),ylabel('y'),colorbar
```

Output:-

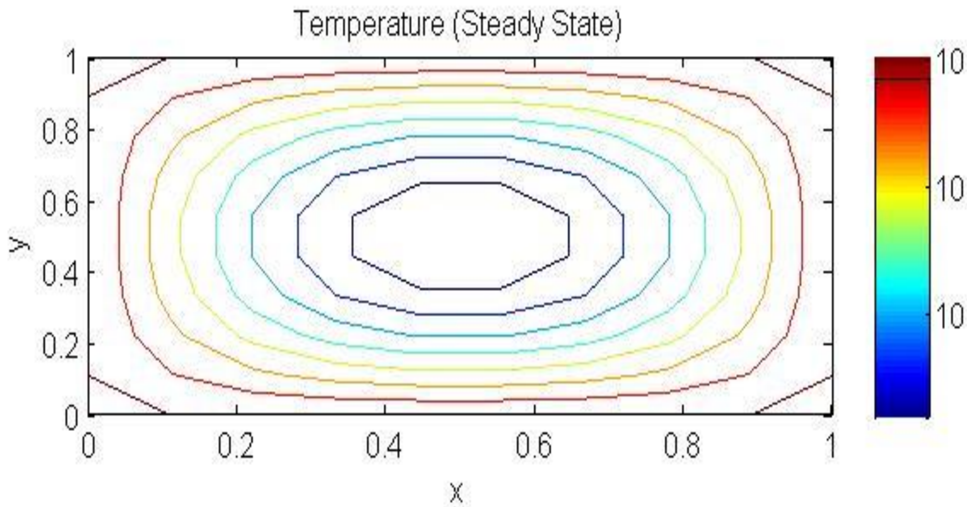


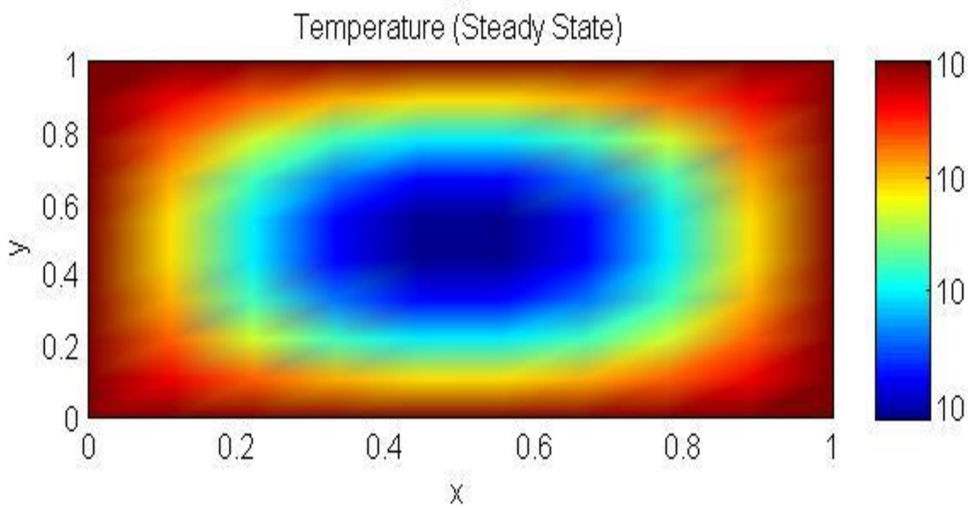


If we change the Initial Condition (As below):-

```
T(1,1:n) = 10; %TOP
T(n,1:n) = 10;  %BOTTOM
T(1:n,1) = 10;  %LEFT
T(1:n,n) = 10;  %RIGHT
```

Output:-


Temperature (Steady State)
y
x
0
0.2
0.4
0.6
0.8
1
10
10
10
10


Temperature (Steady State)
y
x
0
0.2
0.4
0.6
0.8
1
10
10
10
10
10