GRENOBLE ALPES UNIVERSITY

SOLUTION OF HEAT EQUATION WITH FINITE DIFFERENCES

1D PROBLEM

<u>COURSE:</u> Numerical Modeling Workshop

MASTER GEOPHYSICS

STUDENT:

TRABZI Mohammed Amine

SUPERVISOR:

ROMAIN Brossier

2018/2019

1. The heat equation

The heat equation is a parabolic partial differential equation that describes the distribution of heat (or variation in temperature) in a given region over time. [1]

Mathematically, the heat equation is described by following equation: $\frac{\partial u}{\partial t} = \alpha * \nabla^2 u$, where α is a positive constant, and Δ or ∇^2 denotes the Laplace operator. In the physical problem of temperature variation, $u(x,y,z,t)$ is the temperature and α is the thermal diffusivity. For the mathematical treatment it is sufficient to consider the case $\alpha = 1$. [1]

2. Presentation of the problem

In this work, we are about to study the heat diffusion through a rod having dimension and boundary conditions as following:

- We consider a rod having length L=10 meters
- Initial conditions : u(x,0)=100 deg° for 0<x<L
- We are going to study the heat diffusion using two boundary conditions :
  - Dirichlet boundary condition :

    u (0, t)=0, u (L, t)=0

  - Neumann boundary condition :

    $$\frac{\partial u(x,t)}{\partial x}\Big|_{x=0} = 0$$

The following figure represents the rod:

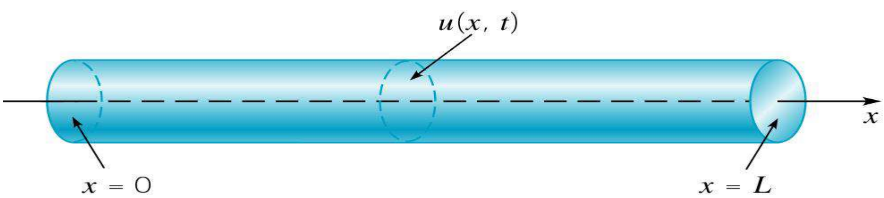

Figure.1. the model of the rod used to study the heat diffusion u(x, t), L=10.

3. Approximation of the heat equation using finite differences method

3.1. The 1D analytical heat diffusion equation is:

$$\frac{\partial u(x,t)}{\partial t} = \alpha * \frac{\partial^2 u(x,t)}{\partial x^2}$$

3.2. Discretization of space and time domain: space step **dx** and time step **dt**.

3.3. Taylor series approximation of the heat equation:

3.3.1. We begin by approximating the left part of the equation:

$$\mathrm{u}(\mathrm{x},\mathrm{t}+\Delta t)=\mathrm{u}(\mathrm{x},\mathrm{t})+\Delta t*\frac{\partial u(x,t)}{\partial t}+\frac{\Delta \mathrm{t}^2}{2!}*\frac{\partial^2 \mathrm{u}(\mathrm{x},\mathrm{t})}{\partial \mathrm{t}^2}+\cdots+\frac{\Delta t^n}{n!}*\frac{\partial^{(n)}u(x,t)}{\partial t^{(n)}}$$

$$\mathrm{u}(\mathrm{x},\mathrm{t}+\Delta t)-\mathrm{u}(\mathrm{x},\mathrm{t})=+\Delta t*\frac{\partial u(x,t)}{\partial t}+O(\Delta t)^2$$

$$\frac{\mathrm{u}(\mathrm{x},\mathrm{t}+\Delta t)-\mathrm{u}(\mathrm{x},\mathrm{t})}{\Delta t}=\frac{\partial u(x,t)}{\partial t}+O(\Delta t)$$

Thus:

$$\frac{\partial u(x,t)}{\partial t}=\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}+O(\Delta t)$$

To make it simpler we can write:

x = i . Δx

t = n . Δt

So: $u(x,t)$= u(i, n) = u_i^n

$u(x,t+\Delta t)$= u(i . Δx, [n+1]. Δt) = u_i^{n+1}

Then, we can write:

$$\boxed{\frac{\partial u(x,t)}{\partial t}=\frac{u_i^{n+1}-u_i^n}{\Delta t}+O(\Delta t)}$$

3.3.2. We approximate the right part of the equation:

$$\mathrm{u}(\mathrm{x}+\Delta x,\mathrm{t})=\mathrm{u}(\mathrm{x},\mathrm{t})+\Delta x*\frac{\partial u(x,t)}{\partial x}+\frac{\Delta x^2}{2!}*\frac{\partial^2 \mathrm{u}(\mathrm{x},\mathrm{t})}{\partial \mathrm{x}^2}+\cdots+\frac{\Delta x^n}{n!}*\frac{\partial^{(n)}u(x,t)}{\partial x^{(n)}}$$

$$\mathrm{u}(\mathrm{x}-\Delta x,\mathrm{t})=\mathrm{u}(\mathrm{x},\mathrm{t})-\Delta x*\frac{\partial u(x,t)}{\partial x}+\frac{\Delta x^2}{2!}*\frac{\partial^2 \mathrm{u}(\mathrm{x},\mathrm{t})}{\partial x^2}-\frac{\Delta x^3}{3!}*\frac{\partial^3 u(x,t)}{\partial x^3}+\cdots+\frac{(-\Delta x)^n}{n!}*\frac{\partial^{(n)}u(x,t)}{\partial x^{(n)}}$$

$$\mathrm{u}(\mathrm{x}+\Delta x,\mathrm{t})+\ \mathrm{u}(\mathrm{x}-\Delta x,\mathrm{t}) = 2*\mathrm{u(x,t)}+\Delta x^2 * \frac{\partial^2 \mathrm{u(x,t)}}{\partial \mathrm{x}^2} + O(\Delta x^4)$$

$$\frac{\partial^2 \mathrm{u(x,t)}}{\partial \mathrm{x}^2} = \frac{\mathrm{u}(\mathrm{x}+\Delta x,\mathrm{t}) - 2*\mathrm{u(x,t)} + \mathrm{u}(\mathrm{x}-\Delta x,\mathrm{t})}{\partial \mathrm{x}^2} + +O(\Delta x^2)$$

Then, we can write:

$$\frac{\partial^2 \mathrm{u(x,t)}}{\partial \mathrm{x}^2} = \frac{u_i^{n+1} - 2*u_i^n + u_{i-1}^n}{\Delta x^2} + \ O(\Delta x^2)$$

The complete approximation of the heat equation is:

$$\boxed{\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha * \frac{u_{i+1}^n - 2 * u_i^n + \ u_{i-1}^n}{\Delta x^2}}$$

After simplification w find the explicit Euler's scheme formula:

$$\boxed{\mathrm{u}_\mathrm{i}^{\mathrm{n}+1} = u_i^n + \frac{\alpha * \Delta t}{\Delta x^2} * (u_{i+1}^n - 2 * u_i^n + \ u_{i-1}^n)}$$

4. Analysis of consistency and convergence of the discrete solution towards the analytical solution

4.1. Consistency:

We found that:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\partial u(x,t)}{\partial t} + O(\Delta t)^1$$

$$\frac{u_i^{n+1} - 2*u_i^n + u_{i-1}^n}{\Delta x^2} = \frac{\partial^2 u(x,t)}{\partial x^2} + \; O(\Delta x^2)$$

<u>So:</u>

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \alpha * \frac{u_i^{n+1} - 2 * u_i^n + u_{i-1}^n}{\Delta x^2} = \cancel{\frac{\partial u(x,t)}{\partial t}} - \alpha * \cancel{\frac{\partial^2 u(x,t)}{\partial x^2}} + \; O(\Delta t + \Delta x^2)$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \alpha * \frac{u_i^{n+1} - 2*u_i^n + u_{i-1}^n}{\Delta x^2} = O(\Delta t + \Delta x^2) \xrightarrow[\Delta t, \Delta x \to 0]{} 0$$

This result means that the explicit Euler scheme is **<u>consistent.</u>**

4.2. Convergence:

<u>The Lax equivalence theorem:</u> In numerical analysis, the equivalence theorem is the fundamental theorem in the analysis of finite difference methods for the numerical solution of partial differential equations. It states that for a consistent finite difference method for a well-posed linear initial value problem, the method is convergent if and only if it is stable.

In the following, we are going to use **<u>Von Neumann the analysis</u>** to proof stability:

We consider T_i^n the exact solution of the heat equation

And we have the computed solution: u_i^n

Then we can write:

$$u_i^n = T_i^n + E_i^n$$

E_i^n: is the error

<u>One can write:</u>

$$\frac{E_i^{n+1} - E_i^n}{\Delta t} = \alpha * \frac{E_{i+1}^n - 2*E_i^n + E_{i-1}^n}{\Delta x^2} \quad \text{.....................} \; (1)$$

The **<u>Von Neumann analysis</u>** consists of considering the error in the following form:

$$E_i^n = e^{c*n*\Delta t} \; . \; e^{j*\epsilon*i.\Delta x}$$

$$E_i^{n+1} = e^{c*(n+1)*\Delta t} \; . \; e^{j*\epsilon*i.\Delta x}$$

We define G the Amplification factor as:

$$G = \frac{E_i^{n+1}}{E_i^n} = e^{c*\Delta t}$$

If "c" is positive, we can have the error increasing!

By substituting in equation (1) and dividing it by $\boldsymbol{e^{a*n*\Delta t}}$ **.** $\boldsymbol{e^{j*\epsilon*i.\Delta x}}$, we find the following equation:

$$\frac{\boldsymbol{e^{(c*\Delta t)}-1}}{\boldsymbol{\Delta t}} = \boldsymbol{\alpha} * \frac{\boldsymbol{e^{j*\epsilon*\Delta x}-2+e^{-j*\epsilon*\Delta x}}}{\boldsymbol{\Delta x^2}}$$

$$G = 1 + \frac{\alpha*\Delta t}{\Delta x^2} * (e^{j*\epsilon*\Delta x} + e^{-j*\epsilon*\Delta x} - 2)$$

Let $A = \frac{\boldsymbol{\alpha*\Delta t}}{\boldsymbol{\Delta x^2}}$

And $e^{j*\epsilon*\Delta x} + e^{-j*\epsilon*\Delta x} = 2*\cos(\epsilon * \Delta x) = 2*\cos(\phi)$

$$G = 1 + A * (2*cos(\phi) - 2)$$

$$G = 1 + 2 * A * (cos(\phi) - 1)$$

$$G = 1 - 2 * A * (1 - cos(\phi))$$

$$\sin^2\left(\frac{\phi}{2}\right) = \frac{1 - cos(\phi)}{2} \quad => \quad 1 - cos(\phi) = 2 * sin^2\left(\frac{\phi}{2}\right)$$

$$G = 1 - 4 * A * sin^2\left(\frac{\phi}{2}\right)$$

The condition to have stability is: $|G| \leq 1$

We have two cases to analyze:

If $1 - 4 * A * sin^2\left(\frac{\phi}{2}\right) > 0$, in this case we have the stability

If $1 - 4 * A * sin^2\left(\frac{\phi}{2}\right) < 0$, to get stability we must respect the following condition:

$0 > 1 - 4 * A * sin^2\left(\frac{\phi}{2}\right) > -1$ => $A * sin^2\left(\frac{\phi}{2}\right) < \frac{1}{2}$

=> $A < \frac{1}{2}$

As soon as we verify the consistency and the stability of our scheme, we can ensure that our calculated solution converges towards the exact solution.

5. Presentation of the results

5.1. Parameters of the rod:

Initial conditions: for time equals to zero, u(x, 0) =100 deg° for 0<x<L

L = 10 meters. In our example we use α=2.

Number of time space samples Nx = 21, Space step dx = $\frac{\mathrm{L}}{(\mathrm{Nx}-1)} = \frac{10}{20} = 0.5$ meter

We have now to calculate the time step depending on the stability condition:

$$A = \frac{\alpha * \Delta t}{\Delta x^2} < 0.5$$

$\Delta t < 0.5 * \frac{\Delta x^2}{\alpha}$ => $\Delta t < 0.0625$; we chose Δt =0.05 second

5.2. Initial and boundary conditions:

At time =0 the temperature of the rod in everywhere 100°deg except in the both ends of the rods. Next, we are going to use two types of boundary conditions. The finite differences method requires predefining boundary conditions in order to find the solution of the heat diffusion equation problem.

5.3. Now we represent the solution of the heat diffusion equation along the rod:

a) **Using Dirichlet boundary conditions:** u(0,t)=0 , u(L,t)=0

In figure2 we represent the initial temperature condition in the rod:

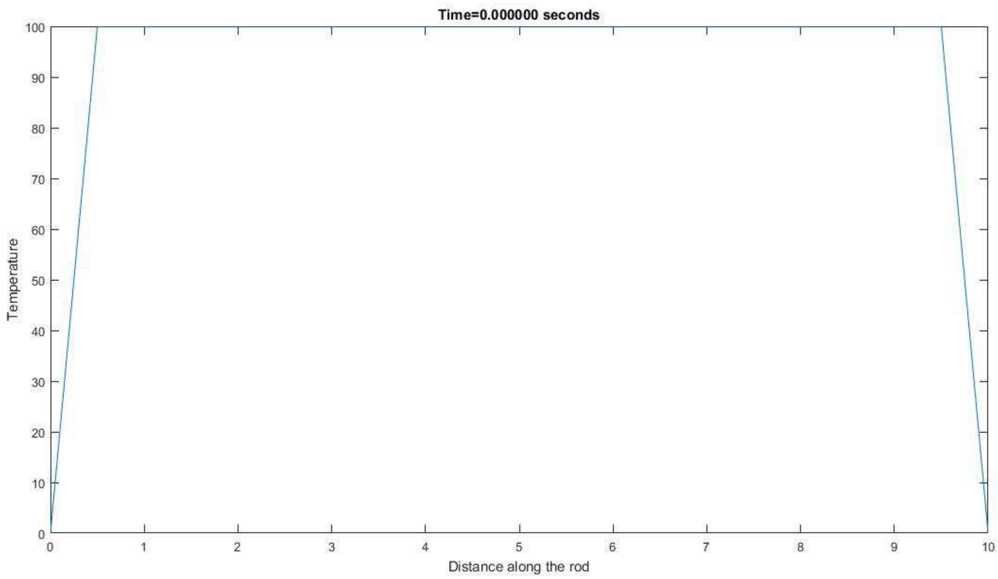

Figure.2. Graphical presenting the temperature in time =0 sec

The sharp drop of temperature from 100°deg to 0°deg is not realistic. To overcome this issue we will apply function that smoothes the rod initial temperature graphical.

To do so, we create, using Matlab, a function that we call sigma:

```
Nx=21;
Lx=10;
dx=Lx/(Nx-1) % step length
sigma= ones(1,Nx);
width=5;
  for  kk=1:width
       sigma(kk) = exp(-1*((width-kk)/width)^2);
       sigma(Nx-kk+1)=sigma(kk);
  end
```

The sigma function in represented in figure 3.

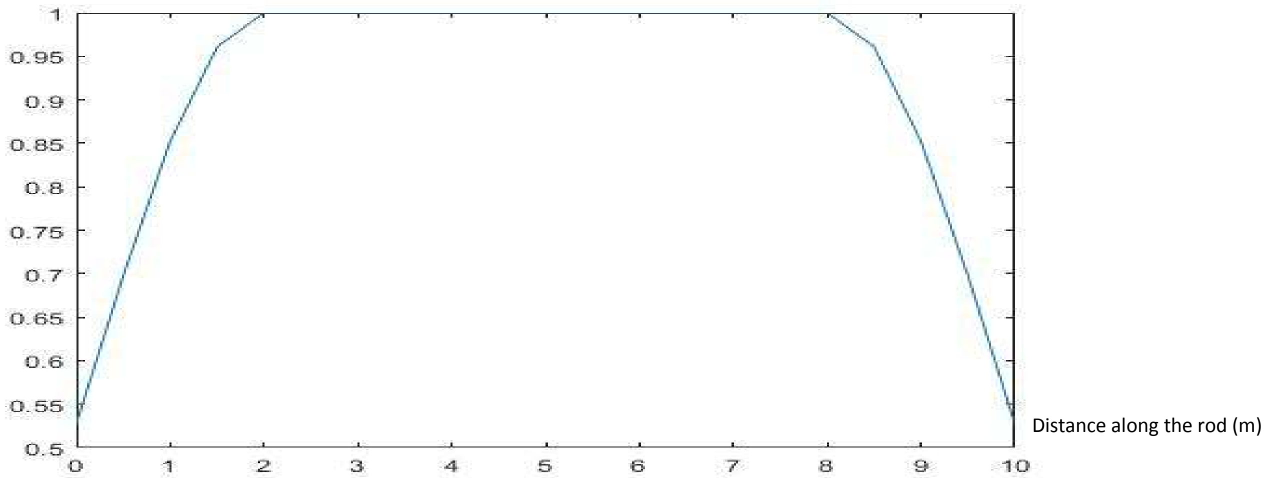

Figure.3. the sigma function used to smooth the initial temperature graphical of rod.

The result after applying the function sigma is represented in figure 4.

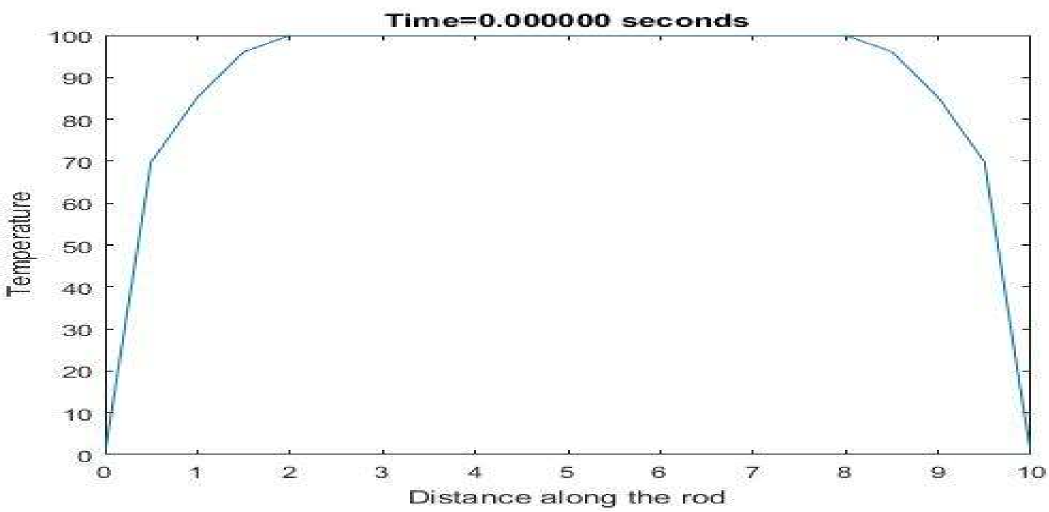

Figure.4. the result after applying sigma function to the first initial condition graphical. This figure represents the temperature of the rod for different positions at time =0 sec, at every point of the rod.

The calculated solution of heat equation u(x, t) is represented in Figure5.

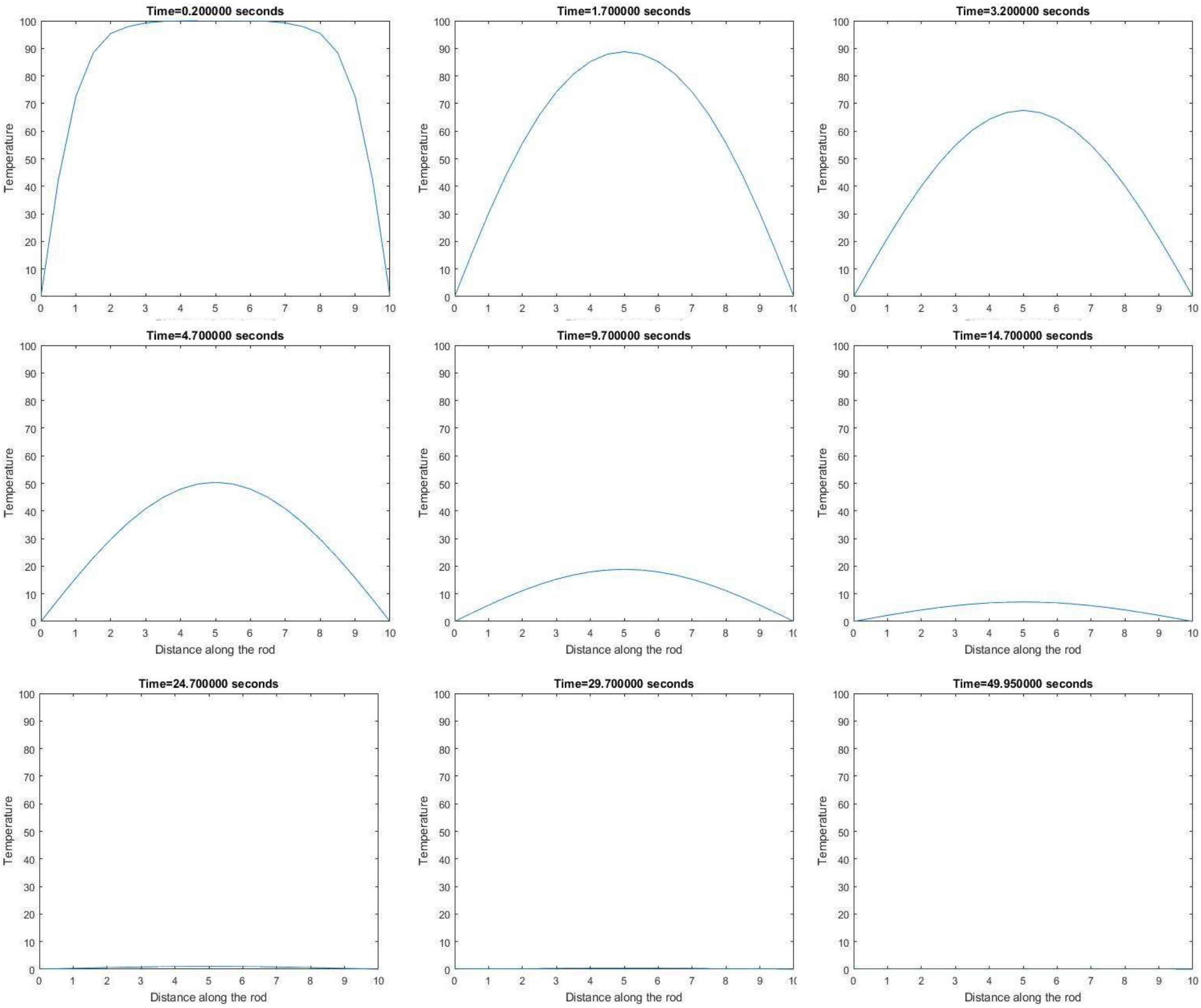

Figure.5.

The heat diffusion through the rod from t=0 to t=50 second

Initial temperature is represented in **figure.4**.

Dirichlet boundary conditions applied are: u(0,t)=0 deg , u(L,t)=0 deg.

b) Using Neumann boundary conditions:

$\frac{\partial u(x,t)}{\partial x}|_{x=0} = 0$ => $\frac{u(1,t)-u(0,t)}{\partial x}$ =0 => u (1,t) = u(2, t);

u(end, t)=0;

The sigma function applied is represented in figure6.

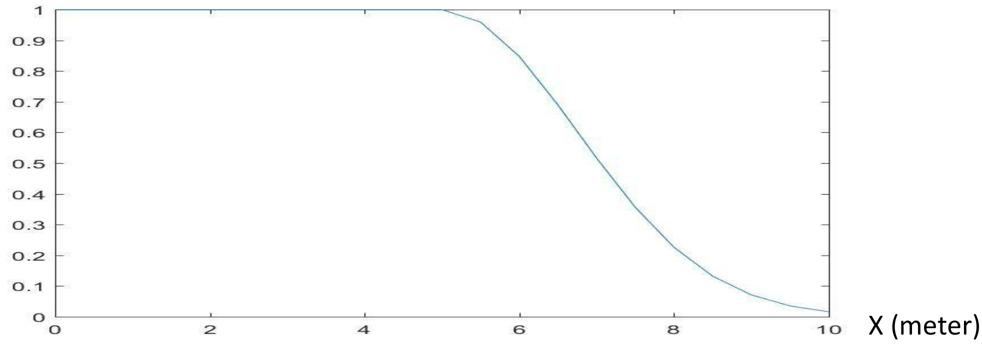

Figure.6. the sigma function used

The temperature at time=0 sec is represented in figure7.

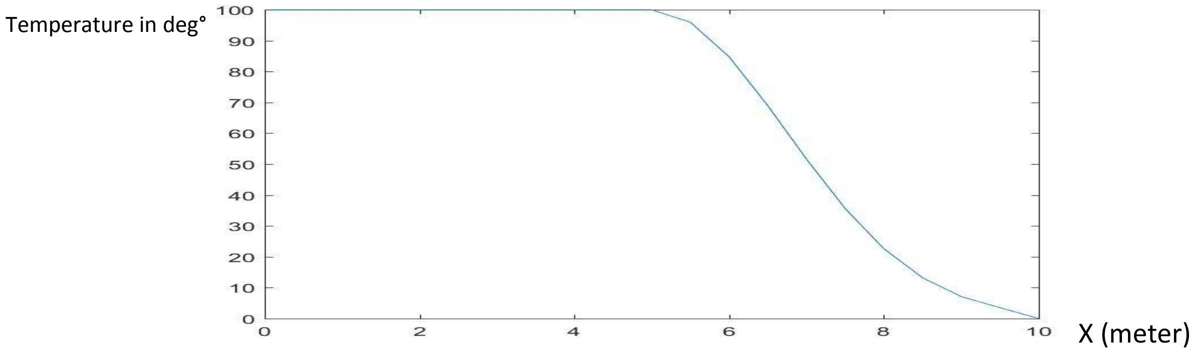

Figure.7. the temperature at time=0, at every point x of the rod of the rod.

The calculated solution u(x,t) of heat equation is represented in Figure8.

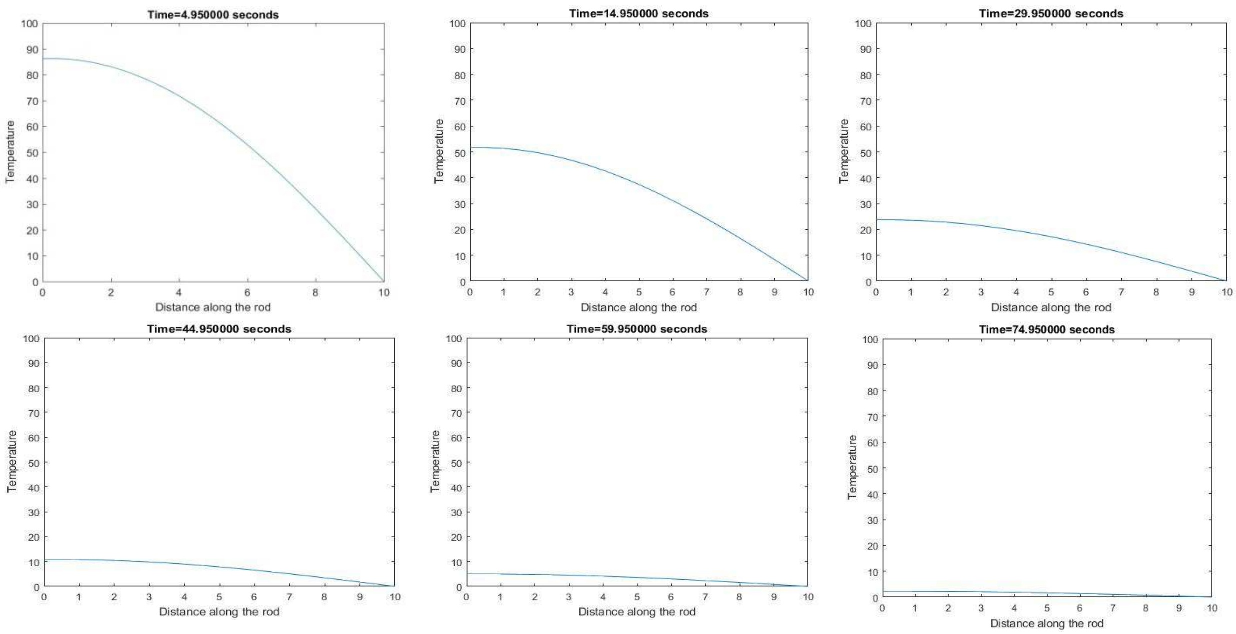

Figure.8. the solution u(x,t) of heat equation. Initial temperature is represented in **figure.7**

Neumann boundary conditions applied are: u (1, t) = u(2, t).

5.4. Analyzing what happens if we do not respect the stability condition:

The condition for stability is $\Delta t < 0.0625$; we chose $\Delta t = 0.07$ second.

In the following figure we represent the results.

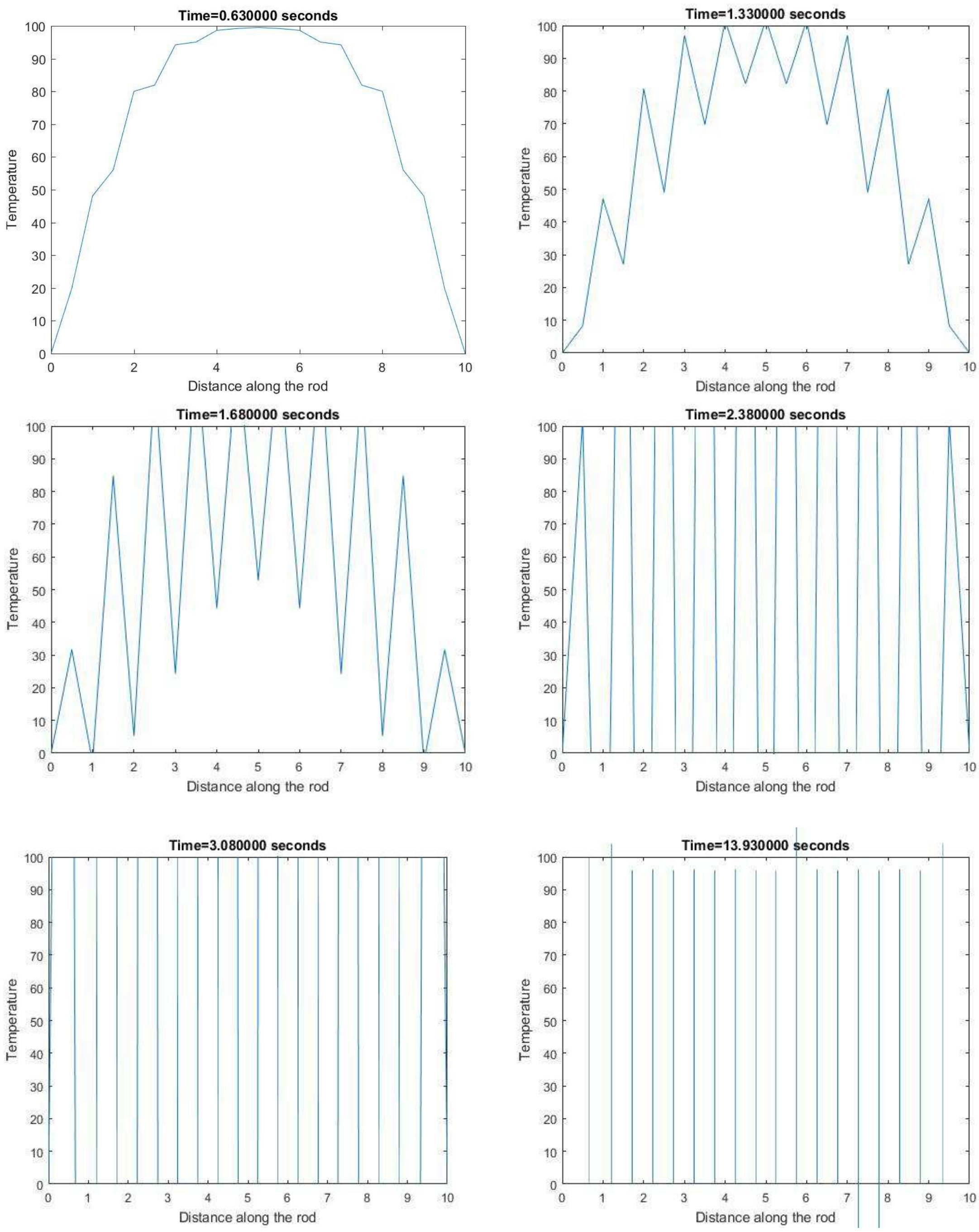

Figure.9. the solution u(x,t) calculated for a time step not satisfying the stability condition. We can see clearly that the solution is **oscillating** rapidly and does not converge to the exact solution at all.

Comment: This example shows that the stability condition is very important to get the calculated solution converged to the exact solution. Even though the scheme is consistent, the solution found is diverging from the right one because the stability is not verified.

5.5. Analyzing what happens if we make α equals to 1:

We maintain the same space and time steps, and then we analyze the effect of the thermal diffusivity α from 2 to 1 to 0.5. Using the same initial condition as before, with Dirichlet boundary conditions. The results are represented in figure 10. We first, analyze the stability condition: A = $\frac{\alpha * \Delta t}{\Delta x^2}$ < 0.5 , $\Delta \boldsymbol{x} = 0.5\, m$, $\Delta \boldsymbol{t} < 0.5 * \frac{\Delta x^2}{\alpha}$, for α=1 $\Delta \boldsymbol{t} < 0.125$, for α=0.5 $\Delta \boldsymbol{t} < 0.25$. So we can use dx=0.5 and dt=0.05, for the thermal diffusivity α changing from 2 to 1 to 0.5. The results are represented in figure 10.

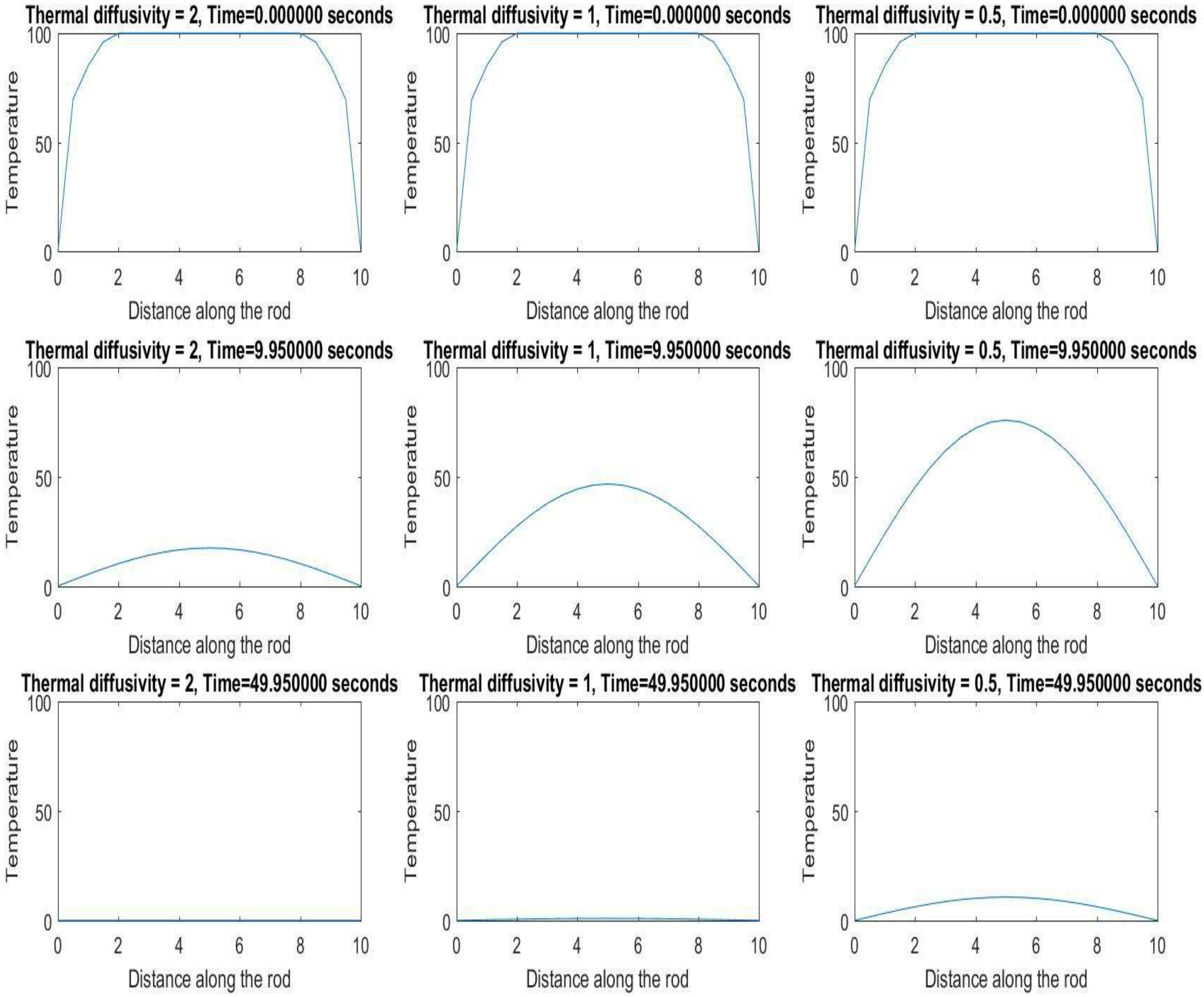

Figure.10. this figure shows the effect of changing thermal diffusivity α, from 2 to 1 to 0.5, on the heat diffusion though rod. We notice that diffusion of the heat is faster when we have α bigger

5.6. Analyzing the diffusion of heat energy applied for 5 second on the rod (L=10 m) at position x=8 meters:

- We remain the same rod used before.
- The initial condition is: u(x, t=0) = 0°deg.
- The thermal diffusivity α=2.
- The boundary conditions: we use the Dirichlet boundary conditions for this example.

Figure 11 represents the results found.

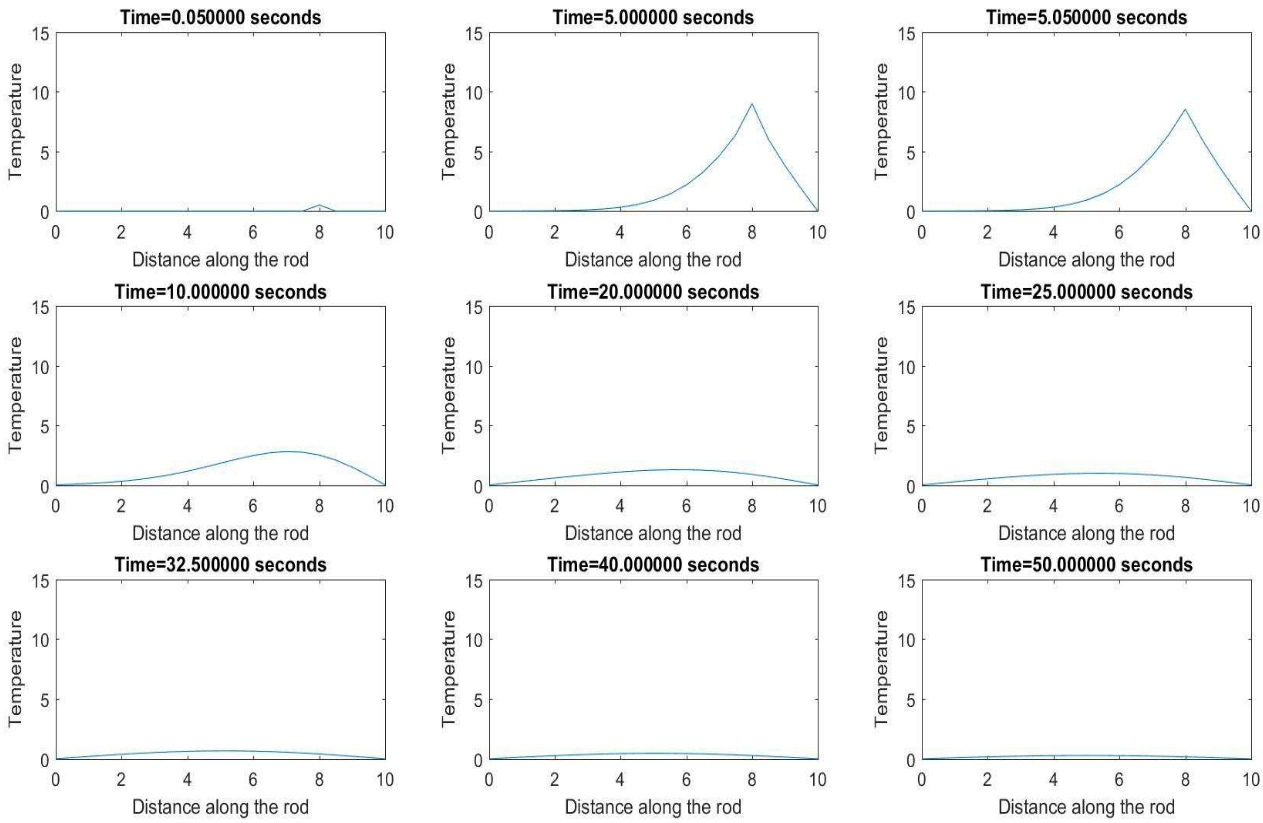

Figure.11. the diffusion of heat energy applied for 5 second on the rod (L=10 m) at position x=8 meters. The code used is presented in the appendix CODE2.

6. Difficulties and learned skills during the project:

During this project, we learned how to use the finite differences method to make a precise approximation of the heat equation in order to get a numerical solution which is the evolution in time and space of the heat through the rod. We have experienced the effect of the stability condition on the results. Also we have seen how the value of thermal diffusivity affects the speed of heat diffusion.

The encountered difficulties that we managed to overcome during this project were:

- How to write down the Matlab code that gives properly the simulation of the heat equation using finite differences method and different boundary conditions.
- How to choose properly space and time steps in order to get the stability and convergence of the solution.

We can do the same simulation of heat equation in 2D dimensions, which is in this case a metal plate (see in the appendix figure 12).

References

I. Wikipedia [1]

II. MIT open source courses website: https://ocw.mit.edu/index.htm

III. Evans, L.C. (1998), Partial Differential Equations, American Mathematical Society, ISBN 0-8218-0772-2

IV. The 1-D Heat Equation 18.303 Linear Partial Differential Equations Matthew J. Hancock

V. Dr. Benjamin Seibold; Numerical Methods for Partial Differential Equations; MIT Course Number 18.336

VI. MIT OpenCourseWare http://ocw.mit.edu 18.336 Numerical Methods for Partial Differential Equations Spring 2009

Appendix:

The main Matlab used for this project

```
% 1---- resolution of heat diffusion equation for 1D rod
%  parameters :
alpha= 2; % Thermal conductivity
%*********************************************************
% Discretization of the rod
Lx=10;
Nx=21; Nt=1001;
% Space step length
dx=Lx/(Nx-1);
% Time step
 dt=0.05 %seconds
% Field variables
T=zeros(Nt,Nx);      % Temperature
x=linspace(0,Lx,Nx); % Distance
% Defining the initial conditions
T(1,:)=100; % Initial Temperature of the rod
T(1,end)=0; % BC at t=0
T(1,1)=0;   % BC at t=0
% loop
for n=1:Nt-1
    % loop for the new temperature for deferent x positions
    for i=2:Nx-1
        T(n+1,i)=T(n,i)+dt*alpha*((T(n,i+1)-2*T(n,i)+T(n,i-1))/dx/dx);
    end
    %BC:
    T(n,1)=0; T(n,end)=0;      % Dirichlet BC
    T(n+1,1)=0; T(n+1,end)=0; % Dirichlet BC
    %T(n+1,1)=T(n+1,2); T(n+1,end)=0; % Neumann BC
end
%plot%
figure(1)
for n=1:Nt
plot(x,T(n,:)), set(gca,'ylim', [0 100]);
xlabel('Distance along the rod'); ylabel('Temperature');
title(sprintf('Thermal diffusivity = 2, Time=%f seconds',(n-1)*dt));
pause(0.01);
end
```

CODE1. the main code used

```matlab
% 2---- Having the two end of the rod with temperatur= 0°deg,  After a long
tim our rod is being have a zero deg°c
% We add now a heat source at x=8 meters and see how will the
% temperature evolves
% resolution of heat diffusion equation for 1D rod
%  parameters :
alpha= 2; % Thermal conductivity
% Descritisation of the rod
Lx=10;
Nx=21; Nt=1001;
dx=Lx/(Nx-1); % step length
% Time step
dt=0.05; % dt=0.05 seconds
% Field variables
Tn=zeros(1,Nx);  % Temperature
x=linspace(0,Lx,Nx); % Distance
% Defining the initiale conditions
T(1,:)=0;
t=0 ;
% loop
for n=1:Nt-1
     t=t+dt;
    % loop for the new temperature for different x positions
    for i=2:Nx-1
        T(n+1,i)=T(n,i)+dt*alpha*((T(n,i+1)-2*T(n,i)+T(n,i-1))/dx/dx);
    end
    % source position
    sx=round(8*Nx/Lx);
    s=10 ; % source energy amplitude
    if t<5
    T(n,sx)=T(n,sx)+dt*s;
    T(n+1,sx)=T(n+1,sx)+dt*s;
    end % at t=0
    %BC:
    T(n,1)=0; T(n,end)=0;     % Dirichlet BC
    T(n+1,1)=0; T(n+1,end)=0; % Dirichlet BC
    % T(n,1)=Tn(n,2); T(n,end)=0; % Neuman BC
end
%plot
figure(1)
 for i=1:Nt
  plot(x,T(i,:)), set(gca,'ylim', [0 15]);
  xlabel('Distance along the rod'); ylabel('Temperature');
  title(sprintf('Time=%f seconds',i*dt));
  pause(0.01);
 end
```

CODE2. The main code Matlab used to simulate 1D heat diffusion through 0°deg rod, using a heat source placed on x=8m for 5 seconds

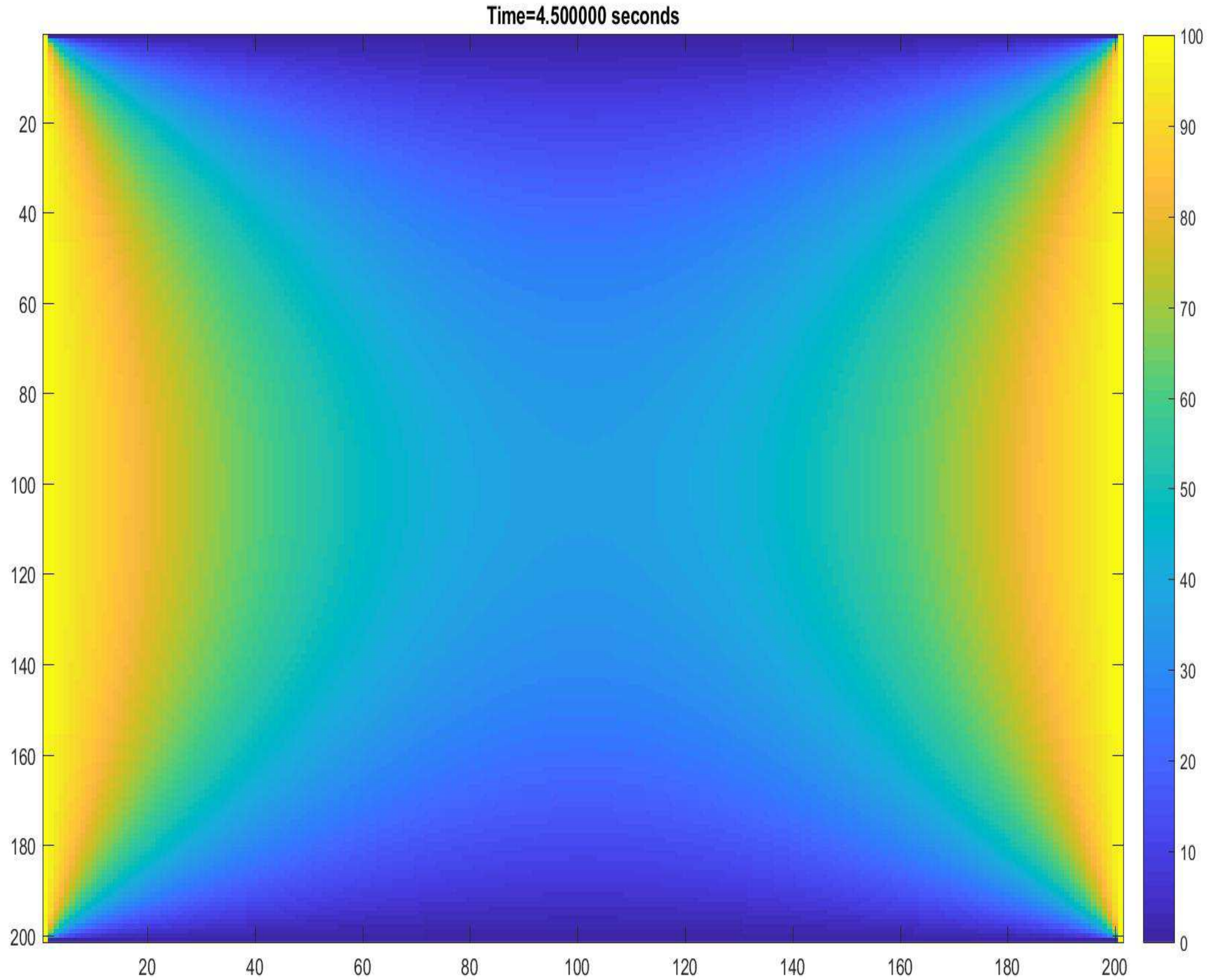

Figure.12. Resolution of heat diffusion equation for 2D metal plate.

Initial conditions: Temperature=0°deg at t=0.

Boundary conditions: Dirichlet BC

T(x=0, y, t)=100°deg; T (x=100, y, t)=100°deg;

T(x, y=0, t)=0°deg; T (x, y=200, t)=0°deg;