Heat Equation

Heat equation governs the temperature distribution in an object. According to the second law of thermodynamics, if two identical bodies are brought into thermal contact and one is hotter than the other, then heat must flow from hotter body to the colder one at a rate proportional to the temperature difference of the two bodies. Therefore, in a metal rod with non-uniform temperature, heat (thermal energy) is transferred from regions of higher temperature to regions of lower temperature. Consider a uniform rod of length L with non-uniform temperature lying on the x-axis from x = 0 to x = L. Assume that the lateral surface of the rod is perfectly insulated, and heat can enter or leave the rod through either of the rod ends and thereby creating a 1D temperature distribution.

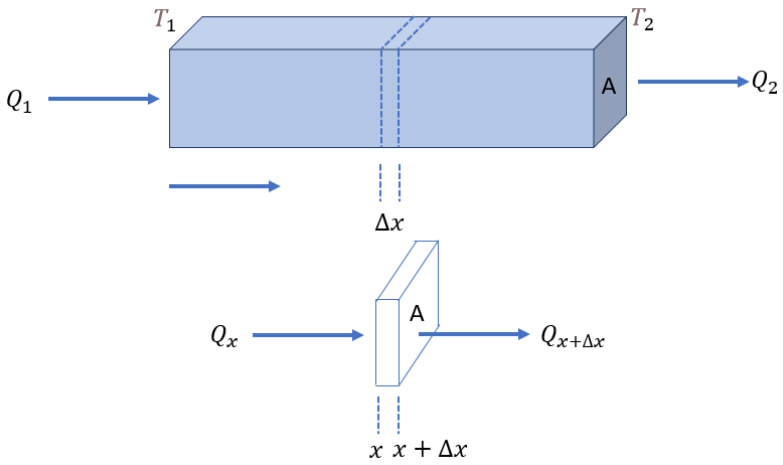


Fig. 1: A rectangular metallic rod with insulated lateral surface and nonuniform heat distribution along length.

Consider an arbitrary thin slice of the rod of width Δx, between x and x+ Δx. The slice is so thin that the temperature throughout the slice is T (x, t). The time heat energy needs to transit through the tiny slice is Δt. The Heat (or thermal) energy of a body with uniform properties is defined as:

$$Q(x,t) = c \times m \times T = c(x) \times \rho(x) A \Delta x \times T(x,t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

Where, c(x) is the specific heat of the material, defined as the amount of heat energy that it takes to raise one unit of mass of the material by one unit of temperature [$c(x) > 0$]. The specific heat may not be uniform throughout the bar and in practice the specific heat depends upon the temperature. However, this will generally only be an issue for large temperature differences. T(x, t) is body temperature at any point x and any time t, m is the body mass. ρ(x) is the mass density which is the mass per unit volume of the material. The mass density may not be uniform throughout the rod.

Let $S(x,t)$ be the heat energy generated per unit volume at location x, and time t. Then, the total energy generated inside the thin slice is given by:

$$\Delta Q_g = A \times \Delta x \times S(x,t) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Now let, Φ (x, t) be the heat flux that is the amount of thermal energy that flows to the right per unit surface area per unit time. The "flows to the right" bit simply tell us that if ϕ (x, t) > 0 for some x and t then the heat is flowing to the right at that point and time. Likewise, if ϕ (x, t) < 0 then the heat will be flowing to the left at that point and time.

According to the law of conservation of energy, the time rate of change of the heat stored at a point on the rod is equal to the net flow of heat into that point.

Change of heat energy of the segment in time Δt	+	Total heat energy generated inside the segment	=	Heat in from left boundary	-	Heat out from right boundary

$$c(x)\,\rho(x)\,A\,\Delta x[T(x,t+\Delta t)-T(x,t)] + A\,\Delta x\,S(x,t)\,\Delta t = A\,\Delta t\,\varphi(x,t) - A\,\Delta t\,\varphi(x+\Delta x,t) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Dividing both sides by $A\,\Delta x\,\Delta t$, equation (3) becomes:

$$c(x)\,\rho(x)[\tfrac{T(x,t+\Delta t)-T(x,t)}{\Delta t}] + S(x,t) = \left[\tfrac{\varphi(x,t)-\varphi(x+\Delta x,t)}{\Delta x}\right] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

The above equation contains two unknown functions T and φ, both of which are function of both time and space. According to Fourier's law of heat transfer, rate of heat transfer is proportional to negative temperature gradient.

$$\varphi(x,t) = -k(x)\frac{\boldsymbol{\partial T}}{\boldsymbol{\partial x}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Where $k(x)$ is the thermal conductivity of the material being studied and measures the ability of the material to conduct heat energy. Thermal conductivity can vary with the location of the rod as well as the temperature. But for small change in total temperature (less than 10 degree), the thermal conductivity can be treated as temperature independent. Now applying Fourier law and then rearranging equation (4) we have,

$$c(x)\rho(x)\,\frac{T(x,t+\Delta t)-T(x,t)}{\Delta t} = k(x)\left[\frac{\left(\frac{\partial T}{\partial x}\right)_{x+\Delta x} - \left(\frac{\partial T}{\partial x}\right)_{x}}{\Delta x}\right] + S(x,t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Now taking the limit Δt, Δx → 0 equation (6) becomes:

$$c(x)\rho(x)\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}[k(x)\frac{\partial T}{\partial x}] + S(x,t) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Now assume that the material in the rod is uniform in nature and thus the thermal properties (specific heat and thermal conductivity) and mass density all are constants.

$$c(x) = c;\ \rho(x) = \rho; and\ k(x) = k$$

The heat equation then takes the form:

$$c\rho \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} + S(x,t) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots . (8)$$

The above equation can further be simplified by defining the thermophysical term: thermal diffusivity to be

$$\alpha = \frac{k}{c\rho}$$

The heat equation then takes the form:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} + \frac{S(x,t)}{c\rho} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots . (9)$$

This is 1D form of heat equation. We can get the 2D and 3D version of heat equation by using Laplacian operator to the first term in right hand side of equation (9)

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T + \frac{S(x,t)}{c\rho} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots . (10)$$

Tow-Temperature Model

The Two Temperature Model (TTM) or Parabolic Two Step (PTS) model is given by:

$$C_e(T_e) \frac{\partial T_e}{\partial t} = \frac{\partial}{\partial x}\left(K_e(T_e, T_l) \frac{\partial T_e}{\partial x}\right) - g(T_e - T_l) + S(x,t) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (10)$$

$$C_l \frac{\partial T_l}{\partial t} = g(T_e - T_l) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (11)$$

Where,

C_e	:	Heat capacity of electrons
C_l	:	Heat capacity of lattice
g	:	Electron-phonon coupling factor
K_e	:	Thermal conductivity
$S(x,t)$	:	Laser source term, heat energy generated per unit volume per unit time.

The electron-phonon coupling factor and the lattice heat capacity are assumed to be constant. The electron heat capacity is a strong function of the electron temperature and thermal conductivity is obtained from electron and lattice temperature and equilibrium electron thermal conductivity measured at room temperature.

$$C_e = C_e^{\iddots} T_e \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (12)$$

$$K_e(T_e, T_l) = k \frac{T_e}{T_l} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (13)$$

Where, k is the equilibrium electron thermal conductivity measured at room temperature. The laser source term has an exponential decay in space to account for absorption in a nontransparent media, and a Gaussian shape in time. Neglecting the temperature dependence of the optical properties a reasonable approximation of the source term is given as.

$$S(x,t) = (1-R)\frac{J}{t_p d} * \exp\left[-\frac{x}{d} - 2.77\left(\frac{t}{t_p}\right)^2\right] \quad \text{.....(14)}$$

Where,

R	:	Reflectivity of the material
J	:	Laser fluence
d	:	Radiation penetration depth
t_p	:	Pulse width

Here R and α are material properties and J and t_p are laser parameters.

Numerical Solution of 1D Heat Equation using Finite Difference Methods for Homogeneous Dirichlet Boundary Condition

The heat equation provides a model for transient heat conduction in a slab of material with finite thickness. Although not exact, numerical solution obtained by methods of numerical analysis gives a very good approximation of the intended model. The Finite Difference method (FDM) is one of the most frequently used numerical tools for solving heat equation. The basic idea behind FDM is to replace continuous partial differential equation (PDE) by difference formulas that involves discrete values associated with nodes on a mesh. The mesh is a set of locations where discrete values are computed. Since heat equation involves both time and space derivatives the mesh can be defined by two key parameters: (i) the local distance between adjacent points in space (Δx) and (ii) the local distance between adjacent time steps (Δt). Depending on the combinations of mesh points used in difference formulas, the heat equation can be solved for three FDM schemes:

I. **Forward Time Centered Space (FTCS)**
II. **Backward Time Centered Space (BTCS)**
III. **Crank-Nicolson Scheme (CNS)**

The heat we will be solving here does not involve the source term and taking the room temperature as initial condition and using thermally insulated boundary condition.

Heat Equation: $\boldsymbol{\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}}$

IC: $T(x, t=0) = T_{Room} = 300$

BC at x = 0: $T(x=0, t) = T_{S1} = 0$

BC at x = L: $(x = L, t) = T_{S2} = 0$

Scheme I: Forward Time Centered Space (FTCS)

FTCS approximation to heat equation is:

$T_i^{m+1} = rT_{i+1}^m + (1-2r)T_i^m + rT_{i-1}^m$; where, $r = \frac{\alpha \Delta t}{\Delta x^2}$

In matrix multiplication form: $T^{m+1} = AT^m$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ r & (1-2r) & r & 0 & 0 & 0 \\ 0 & r & (1-2r) & r & 0 & 0 \\ & & \dots\ \dots & & & \\ & & \dots\ \dots & & & \\ 0 & 0 & r & (1-2r) & r & 0 \\ 0 & 0 & 0 & r & (1-2r) & r \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For Nb, $\alpha = \frac{\kappa}{\rho C_p} = \frac{55\ Wm^{-1}K^{-1}}{(8560\ kgm^{-3})(260\ Jkg^{-1}K^{-1})} = 2.47 \times 10^{-5}\ m^2s^{-1}$

For Stable solution $r < \frac{1}{2}$; $\frac{\alpha \Delta t}{\Delta x^2} < \frac{1}{2}$; $\Delta t < \frac{\Delta x^2}{2\alpha}$

Let, $t_{end} = 1\ ns,$ L = 1 µm and $n_x = 101$; $\Delta x = \frac{L}{(n_x - 1)} = \frac{10^{-6}}{100} = 10^{-8}$

$$\Delta t < \frac{(10^{-8})^2}{4.95 \times 10^{-5}};\ \Delta t \cong \frac{(10^{-8})^2}{1 \times 10^{-4}} = 10^{-12}\ sec$$

$$n_t = \frac{t_{end}}{\Delta t} + 1 = \frac{10^{-9}}{10^{-12}} + 1 = 1001$$

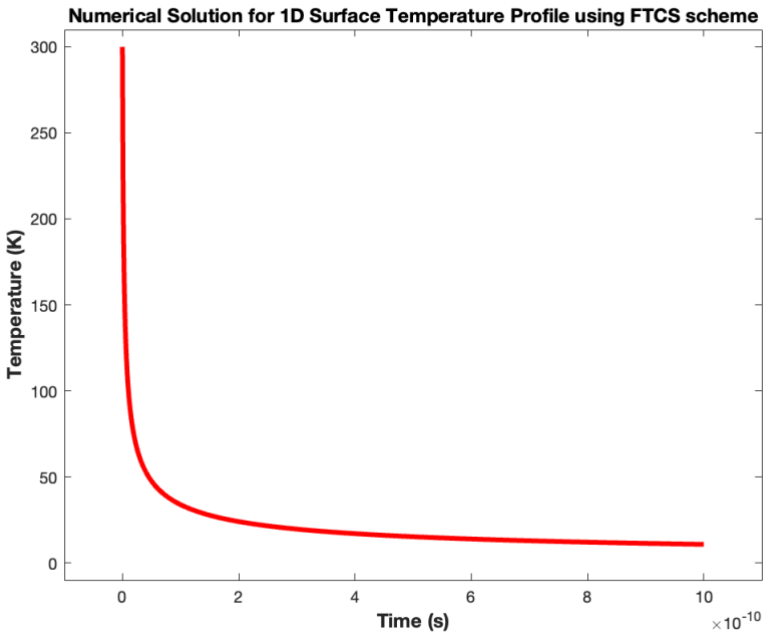
Matlab Code:

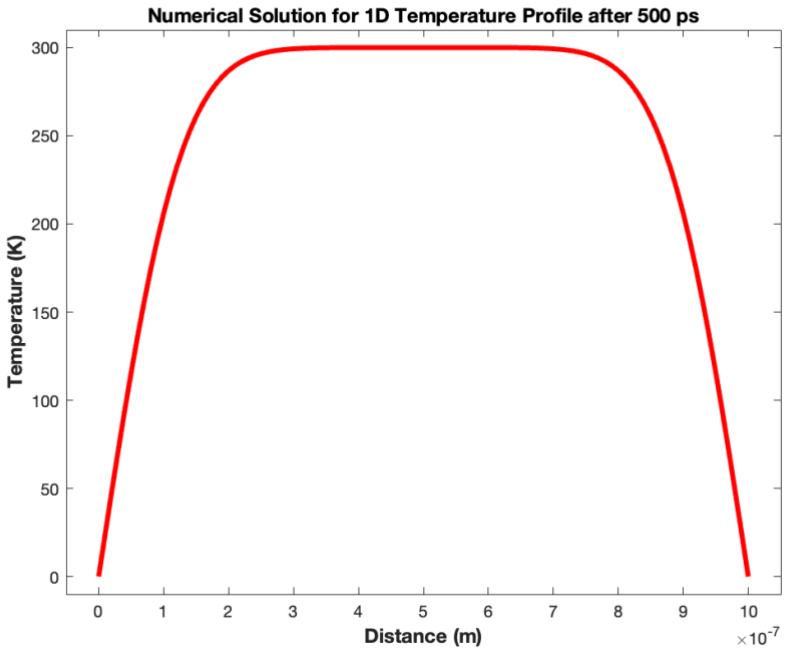
```
clc
clear

% Putting Constant Values
L=1e-6;     % Thickness of metal sample is 1 um
tend=1e-9;  % Diffusion upto 1 ns
alpha=2.47e-5;

%Mesh spacing and time steps
```

```
nx=101;
nt=1001;
dx=L/(nx-1);
dt=tend/(nt-1);

% Calculation of other constant
r=(alpha*dt)/dx^2;
r2=1-2*r;

% Creating arrays to save data
x=linspace(0,L,nx);
t=linspace(0,tend,nt);

% Memory preallocation
u=zeros(nx,nt);

%Initial and Boundary Condition
u(:,1)=300;     % The slab of material is kept at room temperature
u(1,:)=0;       % Homogenious Dirichlet BC at the front
u(nx,:)=0;      % % Homogenious Dirichlet BC at the rear

% Determining FTCS values for interior nodes
   for m=2:nt
      for i=2:(nx-1)
         u(i,m)=r*u(i-1,m-1)+r2*u(i,m-1)+r*u(i+1,m-1);
      end
   end

% Plotting temporal profile of Temperature
      plot(t,u(2,:),'r','linewidth', 3)
      axis([-0.1e-9 1.1e-9 -10 310]);
      title('Numerical Solution for 1D Surface Temperature Profile using FTCS scheme','fontweight',
'bold','FontSize',12)
```

```
xlabel('Time (s)','fontweight','bold','FontSize',12)
ylabel('Temperature (K)','fontweight', 'bold','FontSize',12)


% Plotting temperature profile as a function of Distance
plot(x,u(:, 200),'r','linewidth', 3)
axis([-0.05e-6 1.05e-6 -10 310]);
title('Numerical Solution for 1D Temperature Profile after 500 ps','fontweight', 'bold','FontSize',12)
xlabel('Distance (m)','fontweight','bold','FontSize',12)
ylabel('Temperature (K)','fontweight', 'bold','FontSize',12)
```

Results:





Scheme II: Backward Time Centered Space (BTCS)

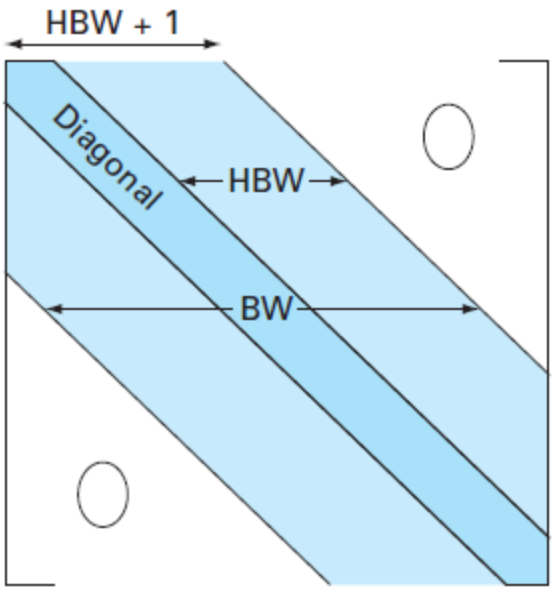
BTCS approximation to heat equation is:

$$\frac{T_i^m - T_i^{m-1}}{\Delta t} = \alpha \frac{T_{i-1}^m - 2T_i^m + T_{i+1}^m}{\Delta x^2}$$

$$-\frac{\alpha}{\Delta x^2} T_{i-1}^m + \left(\frac{1}{\Delta t} + \frac{2\alpha}{\Delta x^2}\right) T_i^m - \frac{\alpha}{\Delta x^2} T_{i+1}^m = \frac{1}{\Delta t} T_i^{m-1}$$

$$a_i T_{i-1}^m + b_i T_i^m + c_i T_{i+1}^m = d_i$$

Therefore, the system of equations can be represented as a tridiagonal system. A tridiagonal system is one with a bandwidth of 3.



$$\begin{bmatrix} b_1 & c_1 & & & & & & \\ a_2 & b_2 & c_2 & & & & & \\ & a_3 & b_3 & c_3 & & & & \\ & & \dots & \dots & \dots & & & \\ & & & \dots & \dots & \dots & & \\ & & & & \dots & \dots & \dots & \\ & & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & & a_n & b_n \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \dots \\ \dots \\ \dots \\ T_{n-1} \\ T_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \dots \\ \dots \\ \dots \\ d_{n-1} \\ d_n \end{bmatrix}$$

Where,

$$a_i = -\frac{\alpha}{\Delta x^2};\ b_i = \frac{1}{\Delta t} + \frac{2\alpha}{\Delta x^2};\ c_i = -\frac{\alpha}{\Delta x^2}, and\ d_i = \frac{1}{\Delta t} T_i^{m-1}$$

So in matrix form the system of equations can be written as follows:

$$A\{T\} = \{d\}$$

This system can be efficiently solved using LU factorization with backward substitution. The LU factorization is such that the system of equations can be written as:

$$[L]\big\{[U]\{T\} - \{D\}\big\} = [A]\{T\} - \{d\}$$

Using rules of matrix multiplication

$$[A] = [L][U] \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\ (1)\ and$$

$$[L]\{D\} = \{d\} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2)$$

The upper triangular matrix is such that

$$[U]\{T\} - \{D\} = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

To solve the tridiagonal system, we must follow two steps:

(i) LU factorization
(ii) Substitution step

LU Factorization: For the coefficient matrix being positive definite the LU factorization can be done without pivoting. The LU factorization can be done as follows:

$[A] = [L][U]$ where,

$$A = \begin{bmatrix} b_1 & c_1 & & & & & & \\ a_2 & b_2 & c_2 & & & & & \\ & a_3 & b_3 & c_3 & & & & \\ & & \ldots & \ldots & \ldots & & & \\ & & & \ldots & \ldots & \ldots & & \\ & & & & \ldots & \ldots & \ldots & \\ & & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & & a_n & b_n \end{bmatrix}$$

$$L = \begin{bmatrix} e_1 & & & & & & & \\ a_2 & e_2 & & & & & & \\ & a_3 & e_3 & & & & & \\ & & \ldots & \ldots & & & & \\ & & & \ldots & \ldots & & & \\ & & & & \ldots & \ldots & & \\ & & & & & a_{n-1} & e_{n-1} & \\ & & & & & & a_n & e_n \end{bmatrix}$$

And,

$$U = \begin{bmatrix} 1 & f_1 & & & & & & \\ & 1 & f_2 & & & & & \\ & & 1 & f_3 & & & & \\ & & & \ldots & \ldots & & & \\ & & & & \ldots & \ldots & & \\ & & & & & \ldots & \ldots & \\ & & & & & & 1 & f_{n-1} \\ & & & & & & & 1 \end{bmatrix}$$

Evaluating each nonzero term in the product LU and setting it equal to the corresponding entry in A gives:

$$e_1 = b_1$$

$$e_1 f_1 = c_1$$

$$a_2 = a_2$$

$$a_2 f_1 + e_2 = b_2$$

$$e_2 f_2 = c_2$$

$$a_i = a_i$$

$$a_i f_{i-1} + e_i = b_i$$

$$e_i f_i = c_i$$

… … … … ….

… … … … ….

$$a_n = a_n$$

$$a_n f_{n-1} + e_n = b_n$$

Solving for the unknown e_i and f_i gives

$$e_1 = b_1$$

$$f_1 = {}^{c_1}/_{e_1} = {}^{c_1}/_{b_1}$$

$$e_i = b_i - a_i f_{i-1}$$

$$f_i = {}^{c_i}/_{e_i}$$

Substitution Steps: First, Eq. (2) is used to generate an intermediate vector {D} by forward substitution. Then, the result is substituted into Eq. (3), which can be solved by back substitution for {T}.

$$\{\boldsymbol{D}\} = {}^{\{\boldsymbol{d}\}}\Big/_{[\boldsymbol{L}]}$$

Forward substitution results in:

$$D_1 = {}^{d_1}/_{e_1}$$

$$D_i = {}^{(d_i - a_i D_{i-1})}/_{e_i}$$

Backward substitution solves for T

$$\{T\} = {\{D\}}/{\{U\}}$$

Which gives:

$$T_n = D_n$$

$$T_i = D_i - f_i D_{i+1}$$

For homogeneous Dirichlet boundary condition where both boundaries are at zero temperature we have:

$$b_1 = 1 \qquad c_1 = 0 \qquad d_1 = T_{s1} = 0$$

$$b_n = 0 \qquad b_n = 1 \qquad d_n = T_{s2} = 0$$

MATLAB Codes:

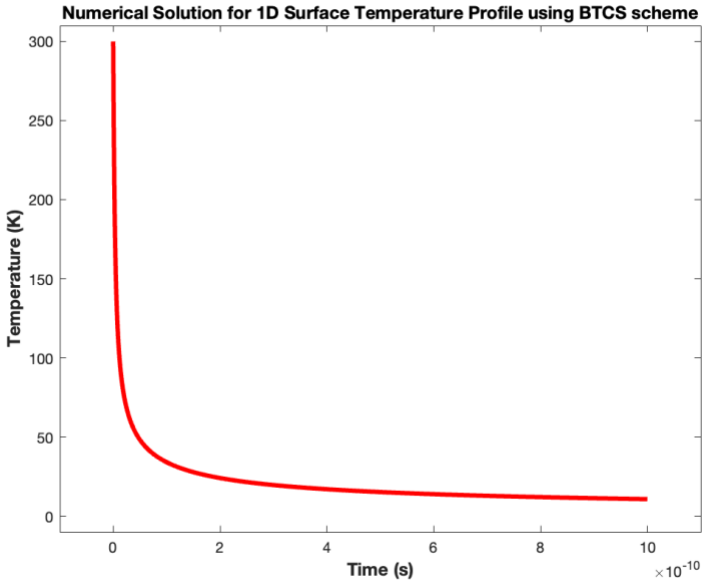
```
% Solve 1D Heat Equation with BTCS scheme

clc
clear

% Putting constant values
L=1e-6;
tend=1e-9;
nx=101;
nt=1001;
alpha=2.47e-5;

%Mesh spacing and time steps
dx=L/(nx-1);
dt=tend/(nt-1);

% Creating arrays to save data
x=linspace(0,L,nx);
t=linspace(0,tend,nt);
```

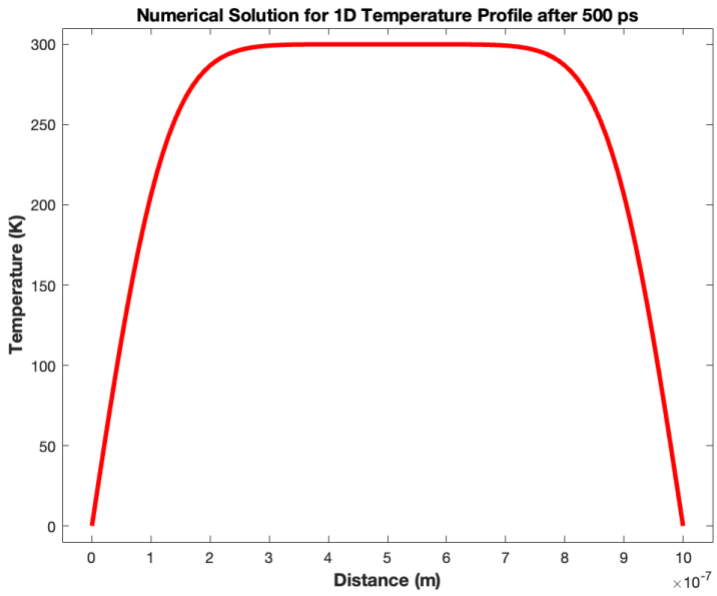
```
% Memory preallocation
T=zeros(nx,nt);

%Setting IC and BC
T(:,1)=300;
T(1,:)=0;
T(nx,:)=0;

% LU factorization and getting coefficient matrix of tridiagonal system
a=(-alpha/dx^2)*ones(nx,1);
c=a;
b=((1/dt)*ones(nx,1))-2*a;
b(1)=1;
c(1)=0;
a(end)=0;
b(end)=1;
[e,f]=tridiagLU(a,b,c);

%Solve for temperature profile by forward and backward substitution
for m=2:nt
   d=T(:,m-1)./dt;
   d(1)=T(1);
   d(end)=T(nx);
   T(:,m)= tridiagLUsolve(d,a,e,f,T(:,m-1));
end

% Plotting temperature profile as a function of time
      plot(t,T(2,:),'r','linewidth', 3)
      axis([-0.1e-9 1.1e-9 -10 310]);
      title('Numerical Solution for 1D Surface Temperature Profile using BTCS scheme','fontweight',
'bold','FontSize',12)
      xlabel('Time (s)','fontweight','bold','FontSize',12)
```

```
        ylabel('Temperature (K)','fontweight', 'bold','FontSize',12)

% Plotting temperature profile as a function of Distance
        plot(x,T(:, 200),'r','linewidth', 3)
        axis([-0.05e-6 1.05e-6 -10 310]);
        title('Numerical Solution for 1D Temperature Profile after 500 ps','fontweight', 'bold','FontSize',12)
        xlabel('Distance (m)','fontweight','bold','FontSize',12)
        ylabel('Temperature (K)','fontweight', 'bold','FontSize',12)


% Define function for LU factorization
function [e,f]=tridiagLU(a,b,c)
n=length(a);
e=zeros(n,1);
f=e;


e(1)=b(1);
f(1)=c(1)/b(1);
    for i=2:n
        e(i)=b(i)-a(i)*f(i-1);
        f(i)=c(i)/e(i);
    end
end


% Define function for solving Tridiagonal system of equations
function D=tridiagLUsolve(d,a,e,f,D)
n=length(d);

    if nargin<5
        D=zeros(n,1);
    end
```

```
    D(1)=d(1)/e(1);

        for i=2:n
            D(i)=(d(i)-a(i)*D(i-1))/e(i);
        end

        for i=n-1:-1:1
            D(i)=D(i)-f(i)*D(i+1);
        end
end
```

Results:





Scheme III: Crank-Nicolson Scheme (CNS)

Like BTCS, CNS is an implicit method and is unconditionally stable. In FTCS and BTCS scheme the right-hand side of the heat equation is approximated by the central difference evaluated at the current time step but in CNS, the right-hand side of the heat equation is approximated with the average of the central differences evaluated at the current and the previous time steps. The left side of the heat equation is approximated with the backward time difference used in the BTCS scheme.

$$\frac{T_i^m - T_i^{m-1}}{\Delta t} = \frac{\alpha}{2}\left[\frac{T_{i-1}^m - 2T_i^m + T_{i+1}^m}{\Delta x^2} + \frac{T_{i-1}^{m-1} - 2T_i^{m-1} + T_{i+1}^{m-1}}{\Delta x^2}\right]$$

$$-\frac{\alpha}{2\Delta x^2}T_{i-1}^m + \left(\frac{1}{\Delta t} + \frac{\alpha}{\Delta x^2}\right)T_i^m - \frac{\alpha}{2\Delta x^2}T_{i+1}^m = \frac{\alpha}{2\Delta x^2}T_{i-1}^{m-1} + \left(\frac{1}{\Delta t} - \frac{\alpha}{\Delta x^2}\right)T_i^{m-1} + \frac{\alpha}{2\Delta x^2}T_{i+1}^{m-1}$$

$$a_i T_{i-1}^m + b_i T_i^m + c_i T_{i+1}^m = d_i$$

Where,

$$a_i = -\frac{\alpha}{2\Delta x^2}\ ; \qquad i = 2, 3, \ldots\ldots\ldots\ldots, N-1$$

$$b_i = \frac{1}{\Delta t} + \frac{\alpha}{\Delta x^2}$$

$$c_i = -\frac{\alpha}{2\Delta x^2}, and$$

$$d_i = -a_i T_{i-1}^{m-1} + \left(\frac{1}{\Delta t} + a_i + c_i\right)T_i^{m-1} - c_i T_{i+1}^{m-1}$$

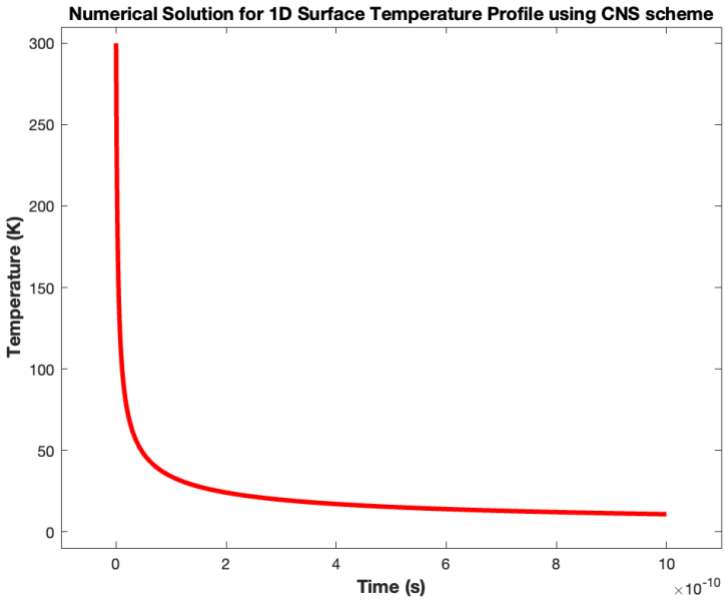
MATLAB Codes:

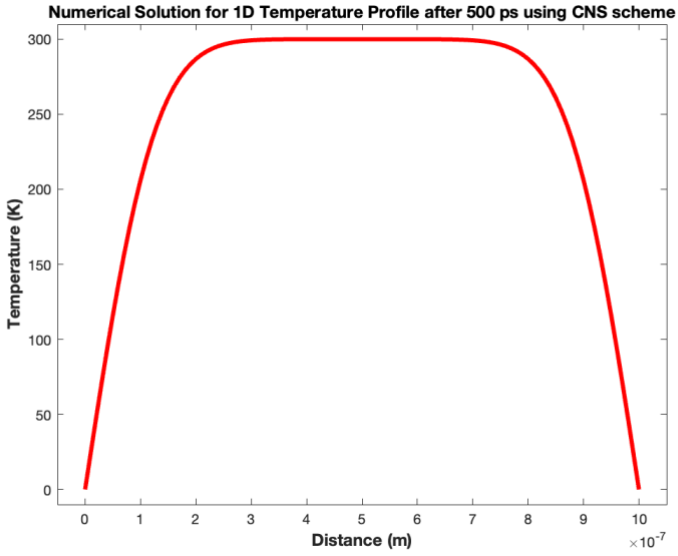
```
% Solve 1D Heat Equation with Crank-Nicolson scheme
clc
clear

% Putting constant values
L=1e-6;
tend=1e-9;
alpha=2.47e-5;

%Mesh spacing and time steps
nx=101;
nt=1001;
dx=L/(nx-1);
dt=tend/(nt-1);

% Creating arrays to save data
x=linspace(0,L,nx);
t=linspace(0,tend,nt);

% Memory preallocation
T=zeros(nx,nt);

%Setting IC and BC
T(:,1)=300;
T(1,:)=0;
T(nx,:)=0;

% LU factorization and getting coefficient matrix of tridiagonal system
a=(-alpha/(2*dx^2))*ones(nx,1);
c=a;
b=((1/dt)*ones(nx,1))-(a+c);
```

```
b(1)=1;
c(1)=0;
a(end)=0;
b(end)=1;
[e,f]=tridiagLU(a,b,c);

%Solve for temperature profile by forward and backward substitution
for m=2:nt
   d=T(:,m-1)./dt-[0;a(2:end-1).*T(1:end-2,m-1);0]+[0;(a(2:end-1)+c(2:end-1)).*T(2:end-1,m-1);0]-
[0;c(2:end-1).*T(3:end,m-1);0];
   d(1)=T(1);
   d(end)=T(nx);
   T(:,m)= tridiagLUsolve(d,a,e,f,T(:,m-1));
end

% Plotting temperature profile as a function of time
      plot(t,T(2,:),'r','linewidth', 3)
      axis([-0.1e-9 1.1e-9 -10 310]);
      title('Numerical Solution for 1D Surface Temperature Profile using CNS scheme','fontweight',
'bold','FontSize',12)
      xlabel('Time (s)','fontweight','bold','FontSize',12)
      ylabel('Temperature (K)','fontweight', 'bold','FontSize',12)

% Plotting temperature profile as a function of Distance
      plot(x,T(:, 200),'r','linewidth', 3)
      axis([-0.05e-6 1.05e-6 -10 310]);
      title('Numerical Solution for 1D Temperature Profile after 500 ps using CNS scheme','fontweight',
'bold','FontSize',12)
      xlabel('Distance (m)','fontweight','bold','FontSize',12)
      ylabel('Temperature (K)','fontweight', 'bold','FontSize',12)

% Define function for LU factorization
function [e,f]=tridiagLU(a,b,c)
```

```
n=length(a);
e=zeros(n,1);
f=e;


e(1)=b(1);
f(1)=c(1)/b(1);
   for i=2:n
      e(i)=b(i)-a(i)*f(i-1);
      f(i)=c(i)/e(i);
   end
end


% Define function for solving Tridiagonal system of equations
function D=tridiagLUsolve(d,a,e,f,D)
n=length(d);

   if nargin<5
      D=zeros(n,1);
   end


   D(1)=d(1)/e(1);

   for i=2:n
      D(i)=(d(i)-a(i)*D(i-1))/e(i);
   end

   for i=n-1:-1:1
      D(i)=D(i)-f(i)*D(i+1);
   end
end
```

Results:


Numerical Solution for 1D Surface Temperature Profile using CNS scheme
Temperature (K)
300
250
200
150
100
50
0
0
2
4
6
8
10
Time (s)
×10^-10


Numerical Solution for 1D Temperature Profile after 500 ps using CNS scheme
Temperature (K)
300
250
200
150
100
50
0
0
1
2
3
4
5
6
7
8
9
10
Distance (m)
×10^-7