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.

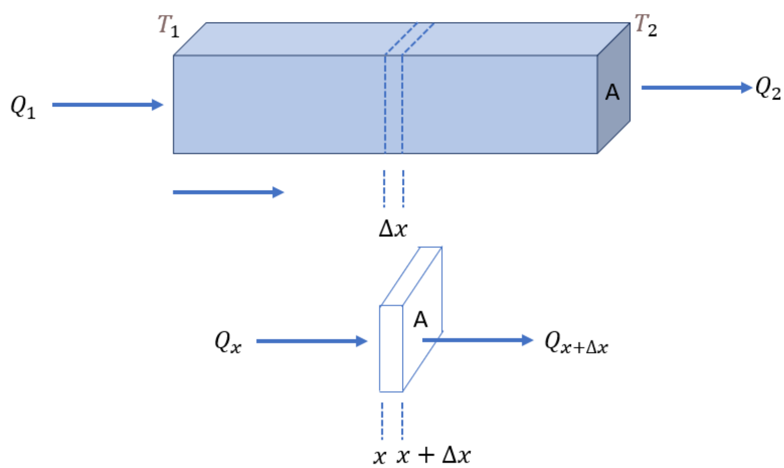


Fig. 1: A rectangular metallic rod with insulated lateral surface and nonuniform heat distribution along length.

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:

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

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:

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

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

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

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

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

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.

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

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,

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

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

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

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:

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

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:

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

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)

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

Tow-Temperature Model

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

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

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

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.

$$C_e = C_e' T_e \dots \dots \dots (12)$$

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

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.

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

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 CNS with Thermal Insulation Boundary Condition:

A 1-D PDE includes a function u(x,t) that depends on time t and one spatial variable x. The MATLAB PDE solver pdepe solves systems of 1-D parabolic and elliptic PDEs of the form:

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right) + s\left(x,t,u,\frac{\partial u}{\partial x}\right) \dots \qquad (8)$$

At the initial time $t = t_0$, for all x, the solution components satisfy initial conditions of the form:

$$u(x,t_0) = u_0(x) \dots \qquad (9)$$

At the boundary x = a or x = b, for all t, the solution components satisfy boundary conditions of the form:

$$p(x,t,u) + q(x,t)f\left(x,t,u,\frac{\partial u}{\partial x}\right) = 0 \dots \qquad (10)$$

The boundary conditions are expressed in terms of the flux f. q(x,t) is a diagonal matrix with elements that are either zero or never zero.

In our case at x = 0, the equation (10) takes the form
pl + ql * f = 0;
$pl + ql * \ k\frac{\partial u}{\partial x} = 0$
Therefore pl = 0 and ql = 1

And at the right boundary (at x =L)

$pr + qr * k\frac{\partial u}{\partial x} = 0;$
Therefore pr = 0 and qr = 1

pl and ql are the coefficients for the left boundary, while pr and qr are the coefficients for the right boundary.

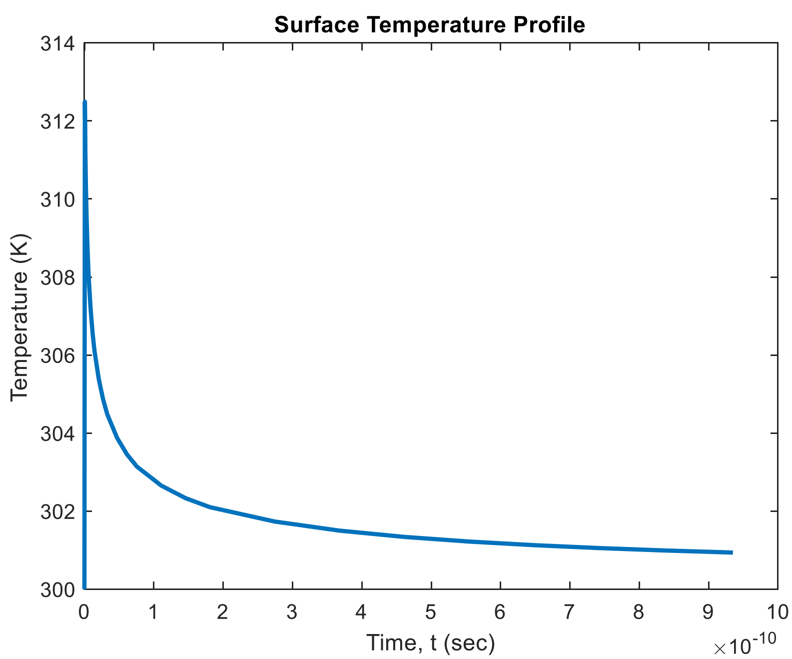
MATLAB CODE for Heat Diffusion Model of Nb:

```
function pdex4
m = 0;
x = linspace(0,400E-9,800);
t = linspace(-100E-15,935E-12,1403);

sol = pdepe(m,@pdex4pde,@pdex4ic,@pdex4bc,x,t);
u = sol(:,:,1);

% A surface plot is often a good way to study a solution.
surf(x,t,u, 'EdgeColor', 'none')
title('Temperature profile for 1D Heat equation')
xlabel('Distance x (m)')
ylabel('Time t (sec)')
zlabel('Temperature (K)')

%%%Plot surface temperature vs. time
figure, plot(t,u(:,1),'linewidth',2)
title('Surface Temperature Profile')
xlabel('Time, t (sec)')
ylabel('Temperature (K)')

%%%Plot of temperature along the length at intermediate
time
figure, plot(x,u(15,:),'linewidth',2)
title('Temperature along the length at time t = 10 ps')
xlabel('Distance, x (m)')
ylabel('Temperature (K)')
% -----------------------------------------------------------
-----


xlswrite('C:\Users\obidu\OneDrive\Desktop\matlab
example\test1.xls',t(:));
xlswrite('C:\Users\obidu\OneDrive\Desktop\matlab
example\test1.xls',u,'b1:b2000');

function [c,f,s] = pdex4pde(x,t,u,DuDx)
```

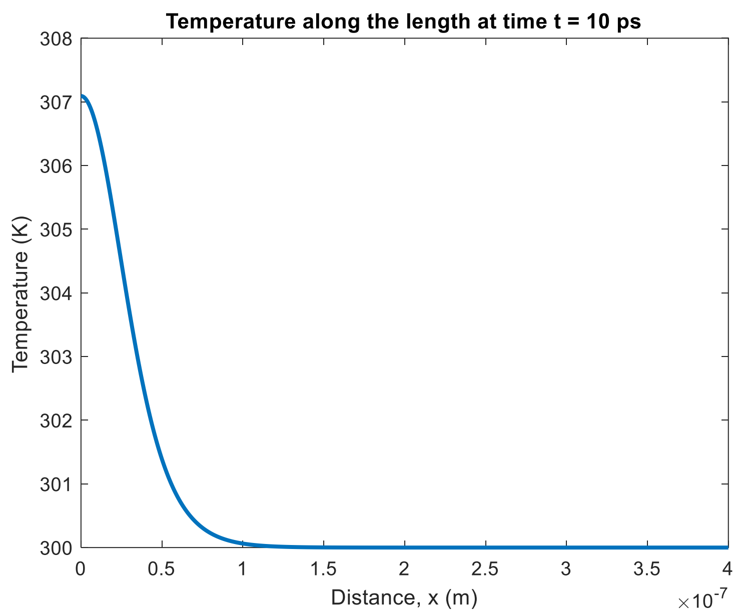
```
c = 2.3E6;
f = 55*DuDx;
s = ((1-0.9)*(3.98E13*7.4E7))*exp(-(x*7.4E7)-(2.77*(t/100E-
15)^2));


% ----------------------------------------------------------
-----
function u0 = pdex4ic(x)
u0 = 300;
% ----------------------------------------------------------
-----
function [pl,ql,pr,qr] = pdex4bc(xl,ul,xr,ur,t)
pl = 0;
ql = 1;
pr = 0;
qr = 1;
```

Results:

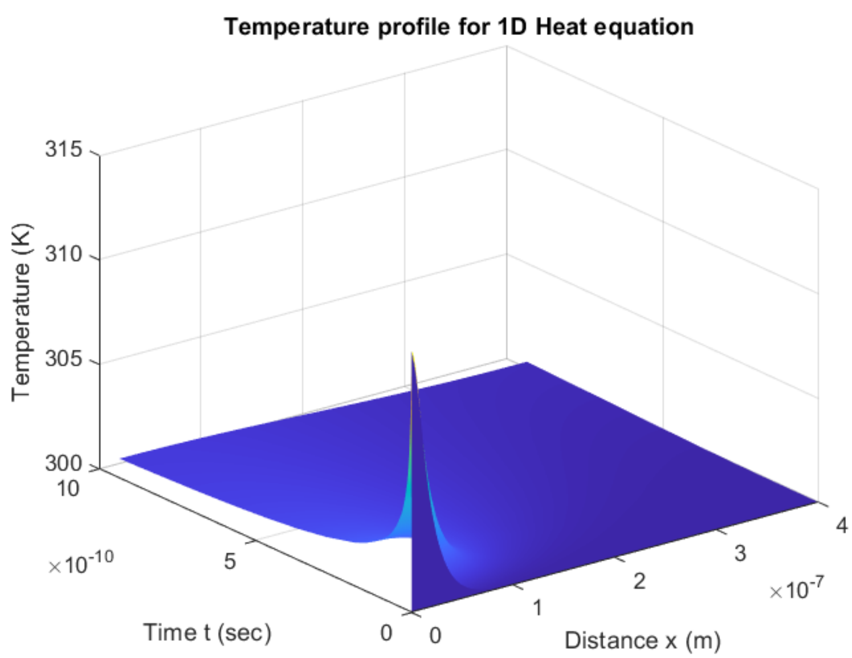
(a) Temperature as function of time



(b) Temperature as function of distance



(c) 2D Temperature Profile

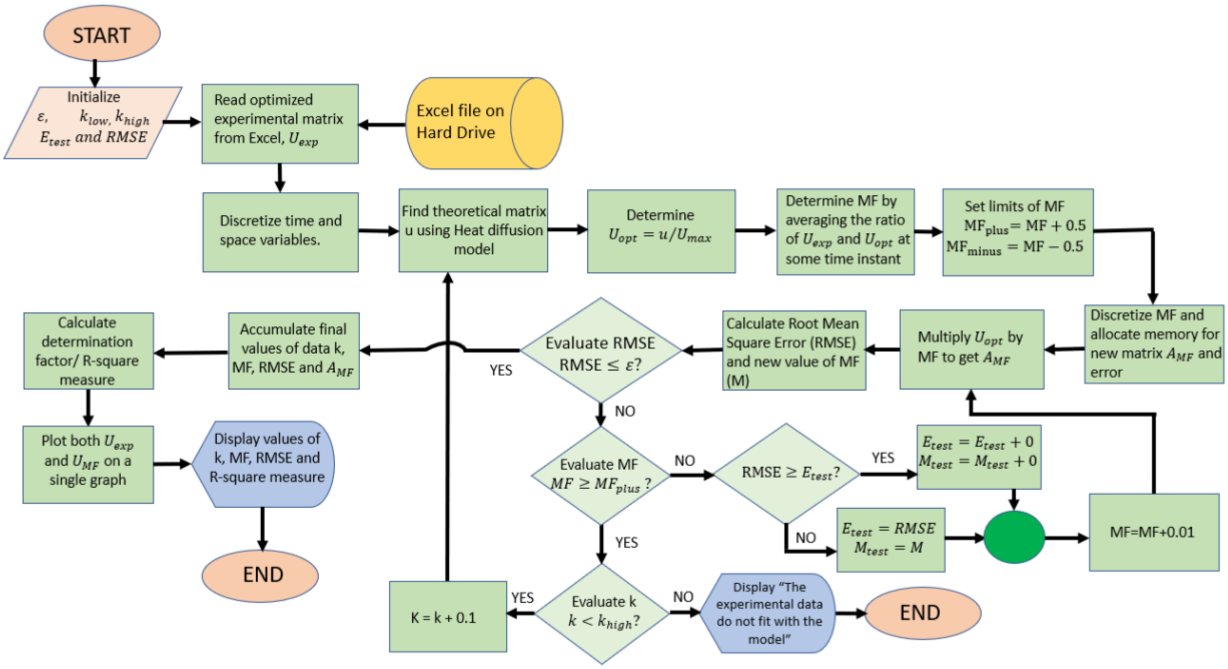


1. Fitting Model

Algorithm for fitting of thermoreflectance data with 1D Heat Model:

1. Define a global variable k and it's range from k_{low} to k_{high}. ($k_{low} = 40\text{w}m^{-1}K^{-1}$, $k_{high} = 65\text{w}m^{-1}K^{-1}$) and initialize for root mean square error (E_test=100 and E_R=0)
2. Read optimized experimental matrix, U_{exp}= ΔR/R from origin/Excel.
3. Discretize time and space variables.
4. Find theoretical matrix u using Heat diffusion model
5. Optimize theoretical matrix u dividing u by u_{max} to get U_{opt}.
6. Determine MF by averaging the ratios of U_{exp} (t) to the U_{opt} (t,1) at time 100, 200,300, 400, 500, 600, 700, 800 and 900 ps
7. Set limits for Multiplication factors $MF_{plus} = MF + 0.5$, $MF_{minus} = MF - 0.5$, and initialize for MF (MF_initial=MF_minus – 0.01 and MF_test = MF_minus)
8. Discretize MF and allocate memory for new matrix A_{MF} and error, E_R.
9. Multiply U_{opt} by MF to get A_{MF}.
10. Calculate Root Mean Square Error (RMSE) by using the formula $RMSE = \sqrt{\frac{\sum_{i=n_1}^{n_2}\left(A_{Model}-A_{Exp}\right)^2}{(n_1-n_2)}}$
11. Evaluate RMSE. If $RMSE \leq \varepsilon$, go to step 16, else go to step 12. % and finish program here. Display Multiplication factor and Error.
    (ii) %set MF=MF+0.01 and go to step 8.
12. Evaluate MF. If $MF \geq MF_{plus}$, go to step 13, else go to step 14.
    (ii) Else set MF=MF+0.01 and go to step 8.
13. Evaluate k.
    (i) If $k < k_{high}$, set $k = k + 0.1$ and go to step 4
    (ii) Else print the message **"The experimental data do not fit with the model for the given thermal conductivity range and error bar"** and finish the program.
14. Evaluate RMSE.
    (i) If $RMSE \geq E_{test}$, set $E_{test} = E_{test} + 0$ and $M_{test} = M_{test} + 0$.
    (ii) Else set $E_{test} = RMSE$ and $M_{test} = M$.
15. Accumulate values of E_{test} and M_{test}, set $MF = MF + 0.01$, and go to step 9.
16. Accumulate final values of parameters k, MF, RMSE and A_{MF}.
17. Calculate coefficient of determination/R^2 measure.
18. Plot both U_{exp} and A_{MF} on a single graph.
19. Display values of k, MF, RMSE and R^2 and finish program.

Flowchart for fitting of thermoreflectance data with 1D Heat Model:



Goodness of Fit (GoF):

The goodness of fit is measured by calculating both the root-mean-square error (RMSE) and the coefficient of determination, (R-squared measure). These parameters are calculated by using the following formulas:

$$RMSE = \sqrt{\frac{\sum_{i=n_1}^{n_2}(A_{Model} - A_{Exp})^2}{(n_1 - n_2)}} \quad \ldots \quad (11)$$

Fitting deviation measured by coefficient of determination can be calculated with two sums of squares formulas:

The total sum of squares: $SS_t = \sum_i (A_{exp}(i) - \overline{A_{exp}})^2$

Where, $\overline{A_{exp}} = \frac{1}{(n_2 - n_1)} \sum_{i=n_1}^{n_2} A_{exp}(i)$

The residual sum of squares: $SS_r = \sum_i (A_{exp}(i) - A_{model}(i))^2$

The R-squared measure is the calculated as:

$$R^2 = 1 - \frac{SS_r}{SS_t} \quad \ldots \quad (12)$$

The ideal value for R^2 is unity. For baseline model $R^2 = 0$. Negative R^2 is obtained for models which have worse predictions than this baseline.

MATLAB CODE for Fitting Model:

```
% Program for fitting Experimental thermoreflectance with 1D
Heat Model generated by PDEPE function

clear
clc
close all

% Define global variable
global k,

% Initialization
eps = 0.02;
k_high=65;
k_low=40;
E_test=100;
E_R=0;

% Read experimental thermoreflectance data from HDD
U_exp=readmatrix('Exp_Data_7_400nm');
A_exp=U_exp(151:end);


% Discretazition
x = linspace(0,400E-9,800);
t = linspace(-100E-15,935E-12,1403);

% Varies the value of thermal conductivity to obtain best fit
for k=k_low:0.1:k_high

% Heat Model solution using PDEPE function
m = 0;
sol = pdepe(m,@pdex4pde,@pdex4ic,@pdex4bc,x,t);
u = sol(:,:,1);

% Optimization of temperature profile
u_max=max(u(:,1));
U_opt=u(:,1)./u_max;
A_model=U_opt(151:end);

% Calculation of multiplication factor
MF=(U_exp(151)/U_opt(151,1)+U_exp(301)/U_opt(301,1)+U_exp(451)/U
_opt(451,1)+U_exp(601)/U_opt(601,1)+U_exp(751)/U_opt(751,1)+U_ex
p(901)/U_opt(901,1)+U_exp(1051)/U_opt(1051,1)+U_exp(1201)/U_opt(
1201,1)+U_exp(1351)/U_opt(1351,1))/9;
```

```
MF_plus=MF+0.5;
MF_minus=MF-0.5;
MF_initial=MF_minus-0.01;
M_test=MF_minus;

MF_vec=MF_minus:0.01:MF_plus;
A_MF=zeros(length(t)-150,1);
E_R=zeros(1,length(MF_vec));

% Varies the value of multiplication factor to obtain best fit
for i=1:length(MF_vec)-1
    M=MF_initial+i*0.01;
    A_MF =A_model.*M;

    % Calculation of Error
    A_R =A_MF -A_exp;
    A_R2 =A_R.^2;
    E_R2 =sum(A_R2);
    E_Ravg=E_R2/(length(t)-150);
    E_R =sqrt(E_Ravg);

    if E_R>=E_test
        E_test=E_test+0;
        M_test=M_test+0;
    else
        E_test=E_R;
        M_test=M;
    end


end

% Accumulate final values
k_final=k;
A_final=A_model.*M_test;
T=t.*1e12;

% Calculation of R-square Measure
A_expbar=sum(U_exp(151:end)/(length(t)-150));
A_SS=U_exp-A_expbar;
A_SS2=A_SS.^2;
SS_T=sum(A_SS2(151:end));
SS_model=(U_exp(151:end)-A_final).^2;
SS_R=sum(SS_model);
R_2=1-(SS_R/SS_T);
```

```
% Plotting best fit result and displaying vital parameters

        plot(T,U_exp,'b',T(151:end),A_final,'r','linewidth', 2)
        axis([-50 1000 -0.1 1.1]);
        title('Thermoreflectance data fitted with Heat Diffusion
Model','fontweight', 'bold','FontSize',12)
        xlabel('Time, t (ps)','fontweight','bold','FontSize',12)
        ylabel('Transient Response (optimized)','fontweight',
'bold','FontSize',12)
        legend({'Experimental curve','Heat Diffusion
Model'},'Location','northeast', 'FontSize',10)
        legend('boxoff')


        dim=[0.33, 0.60, 0.1, 0.1];
        str=sprintf('The root-mean-square error for best fit is:
%.5f', E_test);

annotation('textbox',dim,'String',str,'FitBoxToText','on','EdgeC
olor','none');

        dim=[0.33, 0.55, 0.1, 0.1];
        str=sprintf('The R-Square Measure for best fit is:
%.5f', R_2);

annotation('textbox',dim,'String',str,'FitBoxToText','on','EdgeC
olor','none');

        dim=[0.33, 0.65, 0.1, 0.1];
        str=sprintf('Best fit obtained for k = %.2f Wm-1K-
1',k_final);

annotation('textbox',dim,'String',str,'FitBoxToText','on','EdgeC
olor','none');

        Kp = [' Best fit obtained for thermal conductivity:   k
= ',num2str(k_final),'  Wm-1K-1'];
        disp(Kp)

        Mp = [' Multiplication factor for fitting is:   MF =
',num2str(M_test),];
        disp(Mp)

        E = [' The root-mean-square error is: RMSE =
',num2str(E_test),];
        disp(E)
```

```
        R_square= [' The determination coefficient R-Square
Measure is    R2= ',num2str(R_2),];
        disp(R_square)

disp('The experimental data do not fit with the model for the
given thermal conductivity range and error bar')
 end


% PDEPE function
function [c,f,s] = pdex4pde(x,t,u,DuDx)
c = 2.3E6;
k=52;
f = k*DuDx;
s = ((1-0.9)*5/(100E-15*15.3E-9))*exp(-(x/15.3E-9)-
(2.77*(t/100E-15)^2));
end

% Initial condition
function u0 = pdex4ic(x)
u0 = 0;
end

% Boundary condition
function [pl,ql,pr,qr] = pdex4bc(xl,ul,xr,ur,t)
pl = 0;
ql = 1;
pr = 0;
qr = 1;
end
```

Result of Fitting Model:

```
Command Window
  Best fit obtained for thermal conductivity:   k = 55   Wm-1K-1
  Multiplication factor for fitting is:   MF = 2.691
  The root-mean-square error is: RMSE = 0.0079087
  The determination coefficient R-Square Measure is   R2= 0.99245
fx >>
```

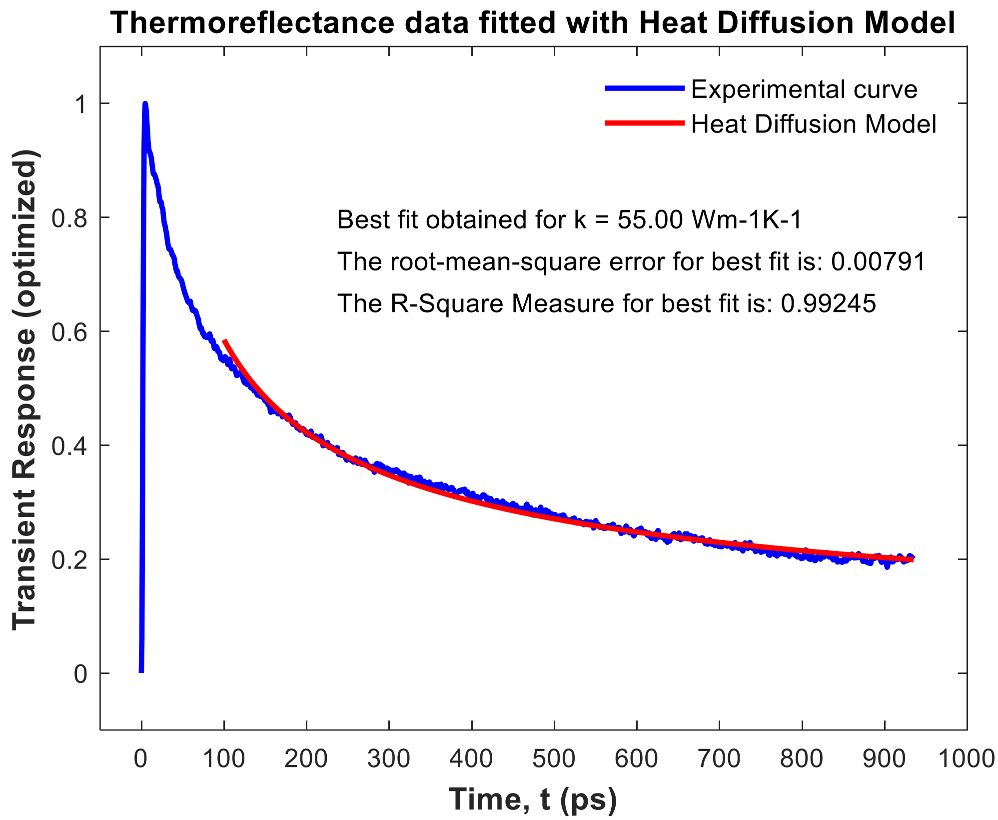


Fig: The experimental data was taken from the 800 nm Nb film deposited on Cu at 670^0 C and is averaging over 5 scans with time resolution 0.667 ps. No filtering and smoothing are used here.