Mech 358 – Computer Lab 3

Problem 1

The m-file used to solve problem 1 is shown below. The analytical solution is shown below in Equation 1. The graph of the solutions is shown below in Figure 1. Please note that all handwritten work for the solution to problem 1 can be found in appendix A along with some flow charts used in producing the code.

```
%this m-file will create and solve the set of linear equations  to solve
%the convective heat transfer problem

clear;
clc;

%define constants
L = 1;          %length of the domain
N = 5;          %number of INTERIOR mesh points
h = L/(N+1);    %mesh spacing

%boundary conditions
T_0 = 0;
T_1 = 0;

%create an (N)x(N) matrix to hold all the equations to be solved
A = zeros(N);

%define row and column counters
i = 1;          %row counter
j = 1;          %column counter

%create the first row of the matrix
A(1,1) = -2/h^2;
A(1,2) = 1/h^2;

%create the last row of the matrix
A(N,N-1) = 1/h^2;
A(N,N) = -2/h^2;

%create a for loop to create the rest of the matrix
for i = 2:(N-1)


    A(i,j) = 1/h^2;
    A(i,j+1) = -2/h^2;
    A(i,j+2) = 1/h^2;
```

```
    %add 1 to j to move over the column counter
    j= j + 1;

end


%create the b vector

%initialize the b vector with zeros
b = zeros(N,1);

%plug in the values at each row
for i = 1:N

    b(i) = -12*(i*h)^2;     % b = -12x^2

end

%solve the equation for the temperature Profile
%intialize the T vector with 0's for 7 points, 5 interior and 2 bc's
T = zeros(N+2,1);
T(2:N+1) = A\b

%compare it with the analytical solution
x = linspace(0,1);
Temp_analytical = -x.^4 + x;

%plot the analytical solution in red
plot(x,Temp_analytical,'r','linewidth',2);
hold on

%plot the finite difference solution
%create an x vector with the same dimensions as the temp vector
x_fd = linspace(0,1,N+2)

%plot the finite difference solution in green
plot(x_fd,T,'g','linewidth',2)

%add labels
xlabel('x');
ylabel('Temperature');
title('Temperature Profile for Rod With Fixed Ends and Spatial Heat Source')



%This part will calculate the partial sums for various values of N and
%plot them all on the same graph

%define a sum for the partial sum initially equal to zero
Sum_3 = 0;

%create a for loop to calculate the partial sum for the first three terms
```

```
for n = 1:3

    Sn = 24*((2-n^2*pi^2)*(-1)^n - 2)*sin(n*pi*x)/(n^5*pi^5)

    Sum_3 = Sum_3 + Sn

end

%plot the partial sum in dashed black lines
plot(x,Sum_3,'--k','linewidth',2)

%add a legend
legend('Analytical Solution','Finite Differences','Fourier Series')
```

Analytical solution is shown below in Equation 1. Sample calculations can be found in appendix A.

Equation 1

$$T(x) = -x^4 + x$$

The temperature profile in the rod with a spatial heat source can be seen below in Figure 1.

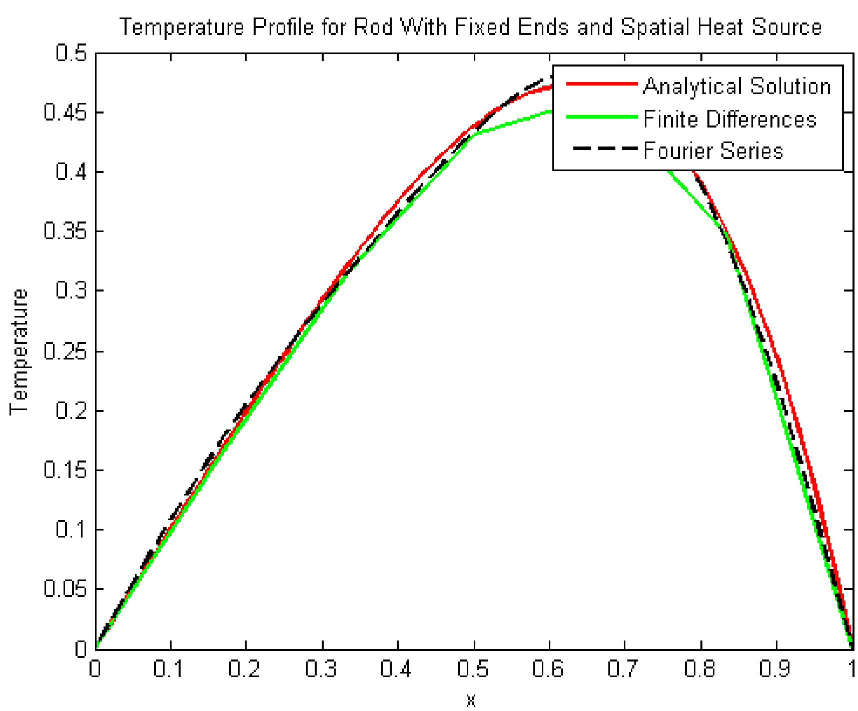


Figure 1 Plot of the temperature profile in the rod

Problem 2

The m-file used to solve this problem is shown below.

```
%this m-file will create and solve the set of linear equations  to solve
%the convective heat transfer problem

clear;
clc;

%define constants
L = 10;          %length of the domain
h = 0.1;         %mesh spacing
N = L/h;         %number of INTERIOR mesh points



%Peclet Number
Pe = 1;



%create an (N-1)x(N-1) matrix to hold all the equations to be solved we
%have N+1 variables but 2 are eliminated using BC's
A = zeros(N-1);

%define row and column counters
i = 1;          %row counter
j = 1;          %column counter

%create the first row of the matrix
A(1,1) = -2/h^2;
A(1,2) = 2/h^2;

%create the last row of the matrix
A(N-1,N-2) =1/h^2 - Pe/(2*h) ;
A(N-1,N-1) = -2/h^2;

%create a for loop to create the rest of the matrix
for i = 2:(N-2)


    A(i,j) = 1/(h^2) - Pe/(2*h);
    A(i,j+1) = -2/(h^2);
    A(i,j+2) = 1/h^2 + Pe/(2*h);

    %add 1 to j to move over the column counter
    j= j + 1;

end



%create the b vector

%initialize the b vector with zeros
```

```
b = zeros(N-1,1);

%The first row is the only non-zero term
b(1) = Pe-2/(h);


%solve the equation for the temperature Profile
%intialize the T vector with 0's for 7 points, 5 interior and 2 bc's
T = zeros(N,1);
T(1:N-1) = A\b;


%plot the finite difference solution
%create an x vector with the same dimensions as the temp vector
x_fd = linspace(0,10,N);

%plot the finite difference solution in green stars
plot(x_fd,T,'*g')
hold on

%plot it against the analytical solution to check if the code works; i can
%comment this part out after initial testing
T_analytical = 1/Pe*exp(-Pe*x_fd);

plot(x_fd,T_analytical,'g')

%add labels
xlabel('x');
ylabel('Temperature');
title('Temperature vs x for the Convective Cooling Problem')


%save the surface temperature so we can plot it vs the Pe
Temp_surf = zeros(4,1);
Temp_surf(1) = T(1);



%this part is for Pe = 2

%define constants
L = 10;          %length of the domain
h = 0.1;         %mesh spacing
N = L/h;         %number of INTERIOR mesh points


%Peclet Number
Pe = 2;

%create an (N-1)x(N-1) matrix to hold all the equations to be solved we
%have N+1 variables but 2 are eliminated using BC's
A = zeros(N-1);
```

```
%define row and column counters
i = 1;          %row counter
j = 1;          %column counter

%create the first row of the matrix
A(1,1) = -2/h^2;
A(1,2) = 2/h^2;

%create the last row of the matrix
A(N-1,N-2) =1/h^2 - Pe/(2*h) ;
A(N-1,N-1) = -2/h^2;

%create a for loop to create the rest of the matrix
for i = 2:(N-2)


    A(i,j) = 1/(h^2) - Pe/(2*h);
    A(i,j+1) = -2/(h^2);
    A(i,j+2) = 1/h^2 + Pe/(2*h);

    %add 1 to j to move over the column counter
    j= j + 1;

end


%create the b vector

%initialize the b vector with zeros
b = zeros(N-1,1);

%The first row is the only non-zero term
b(1) = Pe-2/(h);


%solve the equation for the temperature Profile
%intialize the T vector with 0's for 7 points, 5 interior and 2 bc's
T = zeros(N,1);
T(1:N-1) = A\b;


%plot the finite difference solution
%create an x vector with the same dimensions as the temp vector
x_fd = linspace(0,10,N);

%plot the finite difference solution in green stars
plot(x_fd,T,'*r')
hold on

%plot it against the analytical solution to check if the code works; i can
%comment this part out after initial testing
T_analytical = 1/Pe*exp(-Pe*x_fd);
```

```
plot(x_fd,T_analytical,'r')

%save the surface temp to plot against Pe
Temp_surf(2) = T(2);






%Now for Pe = 4


%define constants
L = 10;          %length of the domain
h = 0.1;         %mesh spacing
N = L/h;         %number of INTERIOR mesh points



%Peclet Number
Pe = 4;

%create an (N-1)x(N-1) matrix to hold all the equations to be solved we
%have N+1 variables but 2 are eliminated using BC's
A = zeros(N-1);

%define row and column counters
i = 1;          %row counter
j = 1;          %column counter

%create the first row of the matrix
A(1,1) = -2/h^2;
A(1,2) = 2/h^2;

%create the last row of the matrix
A(N-1,N-2) =1/h^2 - Pe/(2*h) ;
A(N-1,N-1) = -2/h^2;

%create a for loop to create the rest of the matrix
for i = 2:(N-2)



    A(i,j) = 1/(h^2) - Pe/(2*h);
    A(i,j+1) = -2/(h^2);
    A(i,j+2) = 1/h^2 + Pe/(2*h);

    %add 1 to j to move over the column counter
    j= j + 1;

end



%create the b vector
```

```
%initialize the b vector with zeros
b = zeros(N-1,1);

%The first row is the only non-zero term
b(1) = Pe-2/(h);


%solve the equation for the temperature Profile
%intialize the T vector with 0's for 7 points, 5 interior and 2 bc's
T = zeros(N,1);
T(1:N-1) = A\b;


%plot the finite difference solution
%create an x vector with the same dimensions as the temp vector
x_fd = linspace(0,10,N);

%plot the finite difference solution in green stars
plot(x_fd,T,'*')
hold on

%plot it against the analytical solution to check if the code works; i can
%comment this part out after initial testing
T_analytical = 1/Pe*exp(-Pe*x_fd);

plot(x_fd,T_analytical)

%save the surface temp to plot against Pe
Temp_surf(3) = T(3);





%now for Pe = 8


%define constants
L = 10;           %length of the domain
h = 0.1;          %mesh spacing
N = L/h;          %number of INTERIOR mesh points


%Peclet Number
Pe = 8;

%boundary conditions
T_inf = 0;

%create an (N-1)x(N-1) matrix to hold all the equations to be solved we
%have N+1 variables but 2 are eliminated using BC's
```

```
A = zeros(N-1);

%define row and column counters
i = 1;          %row counter
j = 1;          %column counter

%create the first row of the matrix
A(1,1) = -2/h^2;
A(1,2) = 2/h^2;

%create the last row of the matrix
A(N-1,N-2) =1/h^2 - Pe/(2*h) ;
A(N-1,N-1) = -2/h^2;

%create a for loop to create the rest of the matrix
for i = 2:(N-2)


    A(i,j) = 1/(h^2) - Pe/(2*h);
    A(i,j+1) = -2/(h^2);
    A(i,j+2) = 1/h^2 + Pe/(2*h);

    %add 1 to j to move over the column counter
    j= j + 1;

end


%create the b vector

%initialize the b vector with zeros
b = zeros(N-1,1);

%The first row is the only non-zero term
b(1) = Pe-2/(h);



%solve the equation for the temperature Profile
%intialize the T vector with 0's for 7 points, 5 interior and 2 bc's
T = zeros(N,1);
T(1:N-1) = A\b;



%plot the finite difference solution
%create an x vector with the same dimensions as the temp vector
x_fd = linspace(0,10,N);

%plot the finite difference solution in green stars
plot(x_fd,T,'*k')
hold on

%plot it against the analytical solution to check if the code works; i can
%comment this part out after initial testing
```

```
T_analytical = 1/Pe*exp(-Pe*x_fd);

plot(x_fd,T_analytical,'k')


legend('Finite Difference Pe = 1','Analytical Solution Pe = 1','Finite
Difference Pe = 2','Analytical Solution Pe = 2','Finite Difference Pe =
4','Analytical Solution Pe = 4','Finite Difference Pe = 8','Analytical
Solution Pe = 8')

%Now we need to make a plot of surface temperature (the temp at x = 0)
%versus the Pe

%save the surface temp to plot against Pe
Temp_surf(4) = T(4);



%plot the surface temperature vs. the Pe
Pe_vector = [1;2;4;8];

%plot the Pe vs the surface temperature in a new figure
figure
plot(Pe_vector, Temp_surf)

%add labels and a legend
xlabel('Pe')
ylabel('Surface Temperature (T at x=0)')
title('Surface Temperature vs. Pe')
legend('Surface Temp vs. Pe')
```

The graph of the Temperature vs. x is shown below in Figure 2. The graph of the surface temperature vs. Pe can be seen below in Figure 3.

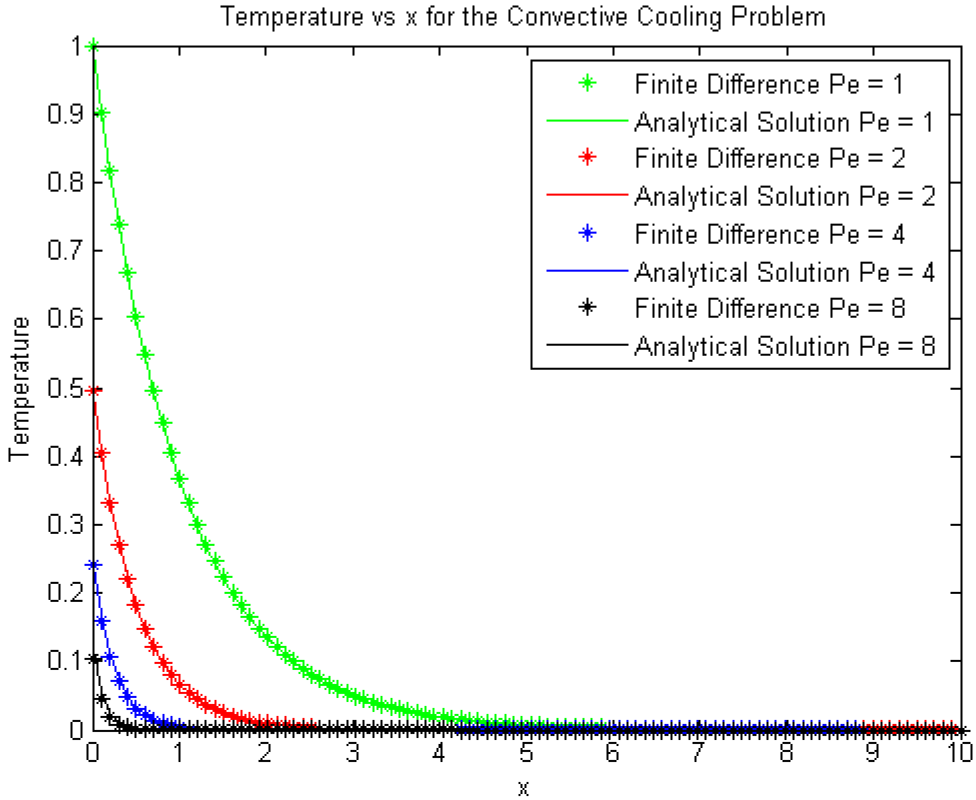


Figure 2 Temperature vs. X for the convective cooling problem

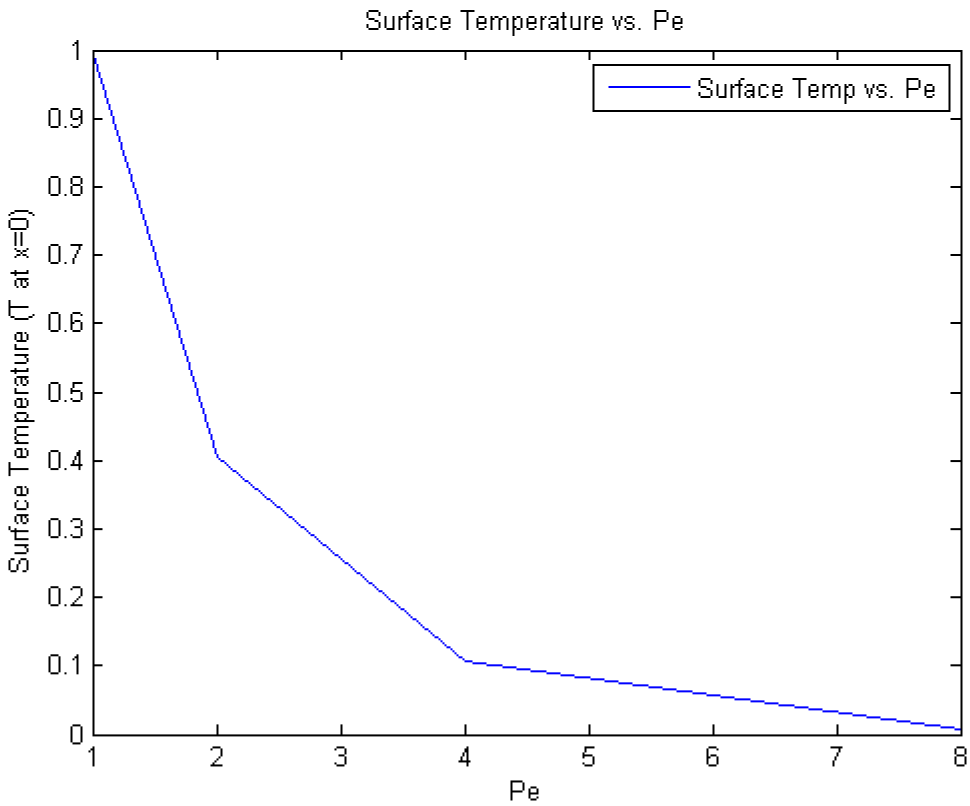


Figure 3 Surface temperature vs. Pe

From visual inspection we can see that a Pe = 2.7 gives a maximum surface temperature of 0.3 if more accuracy is desired, one should examine the Pe number closer to this value and then create a better graph to base intersect with Tmax = 0.3.

We can improve the accuracy using a linear fit between the points for Pe = 2 and Pe = 4.This was done using Matlab (shown below).

```
%fit a line between the surface temps for Pe = 2 and Pe = 4
X = [2,4]
Y = [Temp_surf(2),Temp_surf(3)]

polyfit(X,Y,1)
```

This gave:

$$Tsurface \ = \ -.1429Pe + .703$$

Plugging in Tsurface = 0.3 we could solve for Pe that gave the max surface temperature of 0.3 from the linear approximation. This was found to be

$$\boldsymbol{Pe}\ \mathbf{max} = \mathbf{2.703}$$

Problem 3

The matlab code used to solve problem three is shown below

```
%This matlab code will calculate the solution to the heat equation using
%finite differences
clear
clc

%define x vector from 0 to 1 and we want 11 points
x = 0:0.1:1;

%define the x index counter
j = 1;

%define time step and x step
dt = .001;
dx = 0.1;

%number of points in x vector
N = 11;

%define constants
time = 1;

%define temporary temperatures so we dont have store the temperature
%profile for 1000 timesteps
T_temp1 = zeros(1,N);
T_temp2 = zeros(1,N);

%create the matrix to hold the Temperature profile
T = zeros(5,N);

%Set initial conditions T = 1, at time = 0, for all x
T(1,:) = 1;

%set boundary conditions T = 0, at x=0 and x=1, for t>0
T(2,1) = 0;
T(2,N) = 0;

%initialize the temporary temperature with the next time step
for j = 2:N-1

T_temp1(j) = T(1,j) + dt/(dx^2)*( T(1,j+1) - 2*T(1,j) + T(1,j-1)  );


end

%create a for loop to calculate the temp at the next time steps, for 1000
%timesteps
for timestep = 1:1000
```

```
    %we have to calculate the T_temp 2 for all x
    for j = 2:N-1

        T_temp2(j) = T_temp1(j) + dt/(dx^2)*( T_temp1(j+1) - 2*T_temp1(j) +
T_temp1(j-1)  );

    end

    %check to see if we should store this data, we only store it 5 times
    if timestep == 50 || timestep == 100 || timestep == 200 || timestep ==
1000

        %add 1 to time so we can store the next value for temperature
        time = time +1;

        %store this temperature profile at the specified time
        T(time,:) = T_temp2(:)

    end

    %now we replace T_temp1 with T_temp2 and run the loop agian so that we
    %only have to store a limited number of temperature profiles
    T_temp1 = T_temp2;


end


%now we plot the temperature profile vs x at various times with stars
%at time =0
plot(x,T(1,:),'*k')
hold on

%at time = .05
plot(x,T(2,:),'*r')

%increase the y limit of our plot
axis([0 1 0 1.2])

%at time = 0.1
plot(x,T(3,:),'*')

%at time = 0.2
plot(x,T(4,:),'*g')

%at time = 1.0s
plot(x,T(5,:),'*m')




%Now we plot the exact solutions found using the fourier series
```

```
%define a larger x interval to get a smoother fourier curve
x = linspace(0,1);

%define a sum for the partial sum initially equal to zero
Sum_1 = 0;

%use a for loop to evaluate the first 100 terms first at t = 0
t=0;

for n = 1:100

    Sn = 4/pi* exp( -1*(2*n-1)^2*pi^2*t ) * sin( (2*n-1)*pi*x ) / (2*n-1);

    Sum_1 = Sum_1 + Sn;

end

%plot this partial sum vs. x as a solid black line
plot(x,Sum_1,'k','linewidth',2)



%create another fourier series for t=0.1
%set Sn = 0 again
Sn=0;

%define the sum for the partial sum initially equal to zero
Sum_2 = 0;

%use a for loop to evaluate the first 100 terms first at t = 0.05
t=0.05;

for n = 1:100

    Sn = 4/pi* exp( -1*(2*n-1)^2*pi^2*t ) * sin( (2*n-1)*pi*x ) / (2*n-1);

    Sum_2 = Sum_2 + Sn;

end


plot(x,Sum_2,'r','linewidth',2)



%create another fourier series for t=0.1
%set Sn = 0 again
Sn=0;

%define the sum for the partial sum initially equal to zero
Sum_3 = 0;
```

```
%use a for loop to evaluate the first 100 terms first at t = 0.05
t=0.1;

for n = 1:100

    Sn = 4/pi* exp( -1*(2*n-1)^2*pi^2*t ) * sin( (2*n-1)*pi*x ) / (2*n-1);

    Sum_3 = Sum_3 + Sn;

end


plot(x,Sum_3,'linewidth',2)




%create another fourier series for t=0.2
%set Sn = 0 again
Sn=0;

%define the sum for the partial sum initially equal to zero
Sum_4 = 0;

%use a for loop to evaluate the first 100 terms first at t = 0.05
t=0.2;

for n = 1:100

    Sn = 4/pi* exp( -1*(2*n-1)^2*pi^2*t ) * sin( (2*n-1)*pi*x ) / (2*n-1);

    Sum_4 = Sum_4 + Sn;

end


plot(x,Sum_4,'g','linewidth',2)



%create another fourier series for t=1
%set Sn = 0 again
Sn=0;

%define the sum for the partial sum initially equal to zero
Sum_5 = 0;

%use a for loop to evaluate the first 100 terms first at t = 0.05
t=1;

for n = 1:100
```

```
    Sn = 4/pi* exp( -1*(2*n-1)^2*pi^2*t ) * sin( (2*n-1)*pi*x ) / (2*n-1);

    Sum_5 = Sum_5 + Sn;

end


plot(x,Sum_5,'m','linewidth',2)

%add titles and labels to the graph along with a legend
xlabel('X')
ylabel('T')
title('Temperature vs. X at Various Times for the Transient Heating Problem')

%legend
legend('t=0','t=0.05','t=0.1','t=0.2','t=1.0','fourier solutions shown in
solid lines')
```

The plots of the final solution can be seen below in Figure 4.

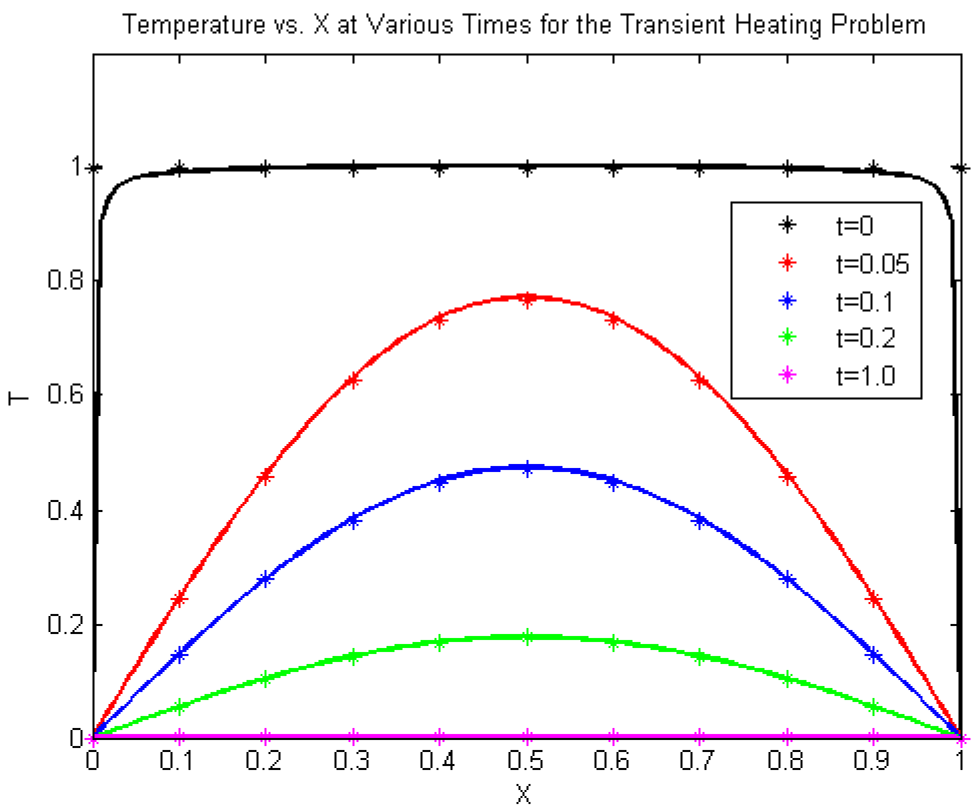


Figure 4 Plot of the final solution for the transient heating problem

Note that for Figure 4, the finite difference solutions are shown as points, the exact solutions found using the Fourier series were plotted using solid lines.

Appendix A – Sample Calculations and Rough Work

Computer Lab 3

LEE SUTTON
48507107

Problem 1

A) Solve Analytically

$$\frac{d^2T}{dx^2} + 12x^2 = 0$$

$$\int d^2T = \int -12x^2 \, dx^2$$

$$\int dT = \int -12\frac{x^3}{3} \, dx + C_1$$

$$T = \left(\frac{-12}{3}\right)\left(\frac{x^4}{4}\right) + C_1 x + C_2$$

$$\boxed{T = -x^4 + C_1 x + C_2}$$

Plug in B.C's

$$T(0) = 0$$

$$0 = 0 + 0 + C_2 \Rightarrow \boxed{C_2 = 0}$$

$$T(1) = 0 = -(1)^4 + C_1(1)$$

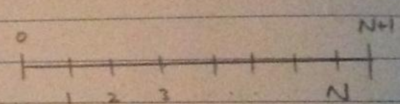
$$\Rightarrow \quad C_1 = 1$$

SOLUTION

$$\boxed{T(x) = -x^4 + x}$$

B) FINITE DIFFERENCES

- N interior points
- spacing $= h$



$$\frac{d^2T}{dx^2} \approx \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2}$$

AT $i=1$

$$\frac{T_2 - 2T_1 + T_0}{h^2} + 12(1 \cdot h)^2 = 0$$

- $T_0 = 0$ FROM B.C's

$$\boxed{\frac{T_2 - 2T_1}{h^2} = -12h^2} \rightarrow 1^{st} \text{ EQN}$$

MIDDLE ROW IN MATRIX (USES NO ENDPOINT)

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} + 12(i \times h)^2 = 0$$

$$\boxed{\frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} = -12(i \times h)^2}$$ → MIDDLE EQN

FINAL EQN (FOR $i=N$)

$$\frac{T_{N+1} - 2T_N - T_{N-1}}{h^2} = -12(N \times h)^2$$

- $T_{N+1} = 0$ FROM B.C'S

$$\boxed{\frac{-2T_N - T_{N-1}}{h^2} = -12(N \times h)^2}$$ → LAST EQN

iv) IN MATRIX VECTOR FORM

$$A\bar{x} = \bar{b}$$

$$A = \begin{bmatrix} \frac{-2}{h^2} & \frac{1}{h^2} & 0 & 0 \\ \frac{1}{h^2} & \frac{-2}{h^2} & \frac{1}{h^2} & 0 \\ & \ddots & \ddots & \ddots \\ 0 & 0 & \frac{1}{h^2} & \frac{-2}{h^2} \end{bmatrix}$$

PART C.

⇒ SOLVE USING FOURIER SERIES

$$\frac{d^2T}{dx^2} + 12x^2 = 0$$

$$T(0) = 0 = T(1)$$

SOLN.

$$T(x) = \sum_{n=1}^{\infty} \frac{24\left[(2-n^2\pi^2)(-1)^n - 2\right]}{n^5\pi^5} \sin(n\pi x)$$

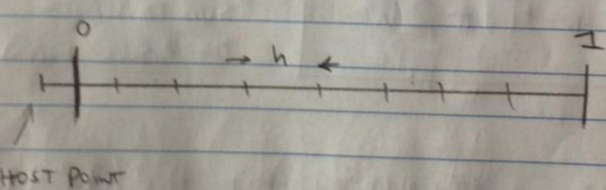
PROBLEM 2

$$\frac{d^2T}{dx^2} + Pe\frac{dT}{dx} = 0$$

B.C's

- $T \to 0$ AS $x \to \infty$
- $\frac{dT}{dx} = -1$ @ $x = 0$

MESH SPACING



SOLUTION WITH FINITE DIFFERENCES

$$\frac{d^2T}{dx^2} + Pe\frac{dT}{dx} = 0$$

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} + \frac{Pe\left(T_{i+1} - T_{i-1}\right)}{2h} = 0$$

AT $i=0$

$$\frac{T_1 - 2T_0 + T_{-1}}{h^2} + \frac{Pe\,(T_1 - T_{-1})}{2h} = 0$$

BUT: $\left.\frac{dT}{dx}\right|_{x=0} = -1$

$$\frac{(T_1 - T_{-1})}{2h} = -1$$

$$T_1 - T_{-1} = -2h$$

$$\boxed{T_{-1} = T_1 + 2h}$$ $\longrightarrow$ SUB THIS INTO THE ABOVE EQN

$$\frac{T_1}{h^2} - \frac{2T_0}{h^2} + \frac{T_1 + 2h}{h^2} + \underbrace{\frac{Pe\,(T_1 - T_{-1})}{2h}}_{=-1} = 0$$

$$\frac{T_1}{h^2} - \frac{2T_0}{h^2} + \frac{T_1}{h^2} + \frac{2}{h} - Pe = 0$$

$$\frac{T_1}{h^2} - \frac{2T_0}{h^2} + \frac{T_1}{h^2} + \frac{2}{h} - Pe$$

$$\boxed{\frac{-2T_0}{h^2} + \frac{2T_1}{h^2} = Pe - \frac{2}{h}}$$ → EQN (1) → 1st ROW OF MATRIX

FOR THE MIDDLE ROWS

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} + \frac{Pe\,(T_{i+1} - T_{i-1})}{2h} = 0$$

$$\frac{T_{i+1}}{h^2} + \frac{Pe}{2h}T_{i+1} - \frac{2T_i}{h^2} + \frac{T_{i-1}}{h^2} - \frac{Pe\,T_{i-1}}{2h} = 0$$

$$\boxed{T_{i+1}\left(\frac{1}{h^2} + \frac{Pe}{2h}\right) + T_i\left(\frac{-2}{h^2}\right) + T_{i-1}\left(\frac{1}{h^2} - \frac{Pe}{2h}\right) = 0}$$

FOR THE LAST ROW (@ $i = N-1$)

$$\frac{T_N + 2T_{N-1} + T_{N-2}}{h^2} + \frac{Pe\,(T_N - T_{N-2})}{2h} = 0$$

B.C: @ $x \to \infty$ $T \to 0$ or $\boxed{T_N = 0}$

EQN BECOMES

$$\frac{-2T_{N-1}}{h^2} + \frac{T_{N-2}}{h^2} + \frac{Pe\,(-T_{N-2})}{2h} = 0$$

Lee Sutton 48507107

$$-2\frac{T_{N-1}}{h^2} + \frac{T_{N-2}}{h^2} - Pe\frac{(T_{N-2})}{2h} = 0$$

$$\boxed{T_{N-1}\left(\frac{-2}{h^2}\right) + T_{N-2}\left(\frac{1}{h^2} - \frac{Pe}{2h}\right) = 0}$$

FINAL EQN

CREATE THE MATRIX VECTOR SOLN.

$$A\bar{x} = \bar{b}$$

$$A = \begin{bmatrix} \frac{-2}{h^2} & \frac{1}{h^2} & 0 & 0 & 0 & 0 \\ 0 & \left(\frac{1}{h^2} - \frac{Pe}{2h}\right) & \left(\frac{-2}{h^2}\right) & \left(\frac{1}{h^2} + \frac{Pe}{2h}\right) & 0 & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & \left(\frac{1}{h^2} - \frac{Pe}{2h}\right) & \left(\frac{-2}{h^2}\right) & \end{bmatrix}$$