ME 552 Midterm

Dominic Riccoboni

Date Created: 10/23/2021

Date Last Modified: 10/24/2021

Table of Contents


```
clear all
```

Problem 3-9 (continued)

Original Problem

Consider the 2-D, steady-state rectangular region $0 \le x \le a, 0 \le y \le b$ in which internal energy is generated at a constant rate $g_0 \ [\frac{W}{m^3}]$ and subjected to the following boundary conditions: boundaries at $x = 0$ and $y = 0$ are kept insulated, whereas the boundaries at $x = a$ and $y = b$ are kept at zero temperature. Calculate the temperature distribution $T(x, y)$.

Parameter Specifications (where thermal conductivity is for aluminum at STP):

$$g_0 = 1.00 \times 10^6 \ [\frac{W}{m^3}] \qquad a = 0.200 \ [m]$$

$$k = 240 \ [\frac{W}{m^\circ C}] \qquad b = 0.100 \ [m]$$

Fourier Coefficients

Verify that the Fourier coefficients after intregration are as follows:

$$C_n = \frac{2g_0(-1)^n}{ak\lambda_n^3 cosh(\lambda_n b)} \quad for \quad n = 1, 2, 3...$$

where

$$\lambda_n = \frac{2n-1}{2a} \quad for \quad n = 1, 2, 3...$$

Verification:

Midterm Problem 3-9 Hand Calculations Fourier Coefficients - Dominic Riccoboni

Saturday, October 23, 2021 6:07 PM

$$C_n = \frac{\frac{g_0 a}{K}\int_0^a \left[\left(\frac{x}{a}\right)^2 - 1\right]\cos(\lambda_n x)\,dx}{\cosh(\lambda_n b)} \qquad n = 0, 1, 2 \ldots$$

$$= \frac{\frac{g_0 a}{K}\left[\frac{1}{a^2}\overbrace{\int_0^a x^2\cos(\lambda_n x)\,dx}^{(1)} - \overbrace{\int_0^a \cos(\lambda_n x)\,dx}^{(2)}\right]}{\cosh(\lambda_n b)}$$

$$(1) \quad \int_0^a \underbrace{x^2}_{u}\underbrace{\cos(\lambda_n x)\,dx}_{dv} = I$$

$u = x^2 \qquad dv = \cos(\lambda_n x)\,dx$

$du = 2x\,dx \qquad v = \frac{1}{\lambda_n}\sin(\lambda_n x)$

$$I = \frac{1}{\lambda_n}\left[x^2\sin(\lambda_n x)\Big|_0^a - \int_0^a \underbrace{2x}_{u}\underbrace{\sin(\lambda_n x)\,dx}_{dv}\right]$$

$u = 2x \qquad dv = \sin(\lambda_n x)\,dx$

$du = 2\,dx \qquad v = -\frac{1}{\lambda_n}\cos(\lambda_n x)$

$$I = \frac{1}{\lambda_n}\left[x^2 \sin(\lambda_n x)\Big|_0^a - \frac{1}{\lambda_n}\left[-2x\cos(\lambda_n x)\Big|_0^a + 2\int_0^a \cos(\lambda_n x)\,dx \right] \right]$$

$$I = \frac{1}{\lambda_n}\left[x^2 \sin(\lambda_n x)\Big|_0^a - \frac{1}{\lambda_n}\left[-2x\cos(\lambda_n x)\Big|_0^a + \frac{2}{\lambda_n}\sin(\lambda_n x)\Big|_0^a \right] \right]$$

- $x^2 \sin(\lambda_n x)\Big|_0^a = a^2 \sin\left(\frac{(2n+1)\pi}{2}\right)$

  $\sin\left((n+\frac{1}{2})\pi\right) = (-1)^n$

  $= a^2 (-1)^n$

- $-2x\cos(\lambda_n x)\Big|_0^a = -2a\cos\left(\frac{(2n+1)\pi}{2}\right) - 0$

  $\cos\left(\frac{(2n+1)\pi}{2}\right) = 0$

  $= 0$

- $\frac{2}{\lambda_n}\sin(\lambda_n x)\Big|_0^a = \frac{2}{\lambda_n}(-1)^n$

$$\Rightarrow \quad I = \frac{1}{\lambda_n}\left[a^2(-1)^n - \frac{1}{\lambda_n}\left[\frac{2}{\lambda_n}(-1)^n\right]\right]$$

$$I = \left[\frac{a^2}{\lambda_n} - \frac{2}{\lambda_n^3}\right](-1)^n$$

(2) $$\int_0^a \cos(\lambda_n x)\,dx = \frac{1}{\lambda_n}\sin(\lambda_n x)\Big|_0^a$$

$$= \frac{1}{\lambda_n}(-1)^n$$

$$C_n = \frac{\frac{g_0 a}{K}\left[\frac{1}{a^2}\left[\frac{a^2}{\lambda_n} - \frac{2}{\lambda_n^3}\right](-1)^n - \frac{1}{\lambda_n}(-1)^n\right]}{\cosh(\lambda_n b)}$$

$$= \frac{\frac{g_0 a}{K}\left[\left[\frac{1}{\lambda_n} - \frac{2}{a^2\lambda_n^3} - \frac{1}{\lambda_n}\right](-1)^n\right]}{\cosh(\lambda_n b)} \qquad , n = 0, 1, 2 \ldots$$

$$C_n = \frac{-2g_0(-1)^n}{a\lambda_n^3 K \cosh(\lambda_n b)}, \quad n = 0, 1, 2 \ldots$$

$$C_n = \frac{2g_0(-1)^{n+1}}{a\lambda_n^3 K \cosh(\lambda_n b)}, \quad n = 0, 1, 2 \ldots$$

WITH $\lambda_n = \frac{(2n+1)\pi}{2a}$

OR

$$C_n = \frac{2g_0(-1)^n}{a\lambda_n^3 K \cosh(\lambda_n b)}, \quad n = 1, 2, 3 \ldots$$

WITH $\lambda_n = \frac{(2n-1)\pi}{2a}$

Produce Solution Plots

The total solution, $T(x, y)$ is decomposed as follows:

$$T(x, y) = \Psi(x, y) + \Phi(x)$$

where

$$\Psi(x, y) = \sum_{n=1}^{\infty} C_n cos(\lambda_n x) cosh(\lambda_n y)$$

and

$$\Phi(x) = \frac{g_0 a^2}{2k}[1 - (\frac{x}{a})^2]$$

```
%System Parameters
g0 = 1e6; % [W/m^3], uniform inter heat generation
k = 240; % [W/(m*K)], thermal conductivity
a = 0.2; % [m], length in x-direction
b = 0.1; % [m], length in y-direction

%Calculate Eigenvalues and Fourier Coefficients
n = 1:200; % Eignevalue index
lambda = (2*n-1)*pi/(2*a);
C = 2*g0*(-1).^n./(a*lambda.^3*k.*cosh(lambda*b));

%Allocate memory for solution's domain vectors
N = 101; %Number of data points for each dimension
x = linspace(0,a,N); %Break up the x dimension width into N data points
y = linspace(0,b,N); %Break up the y dimension width into N data points

%T(x,y) evaluated at the domain variables
phi = zeros(N,1); % Initialize 1-D ODE solution
T1 = zeros(N); % Initialize 2-D PDE solution
Tmax = g0*a^2/(2*k); % deg. C, Maximum T for 1-D ODE problem
for i = 1:N
phi(i) = Tmax*(1 - (x(i)/a)^2);
    for j = 1:N
        T1(i,j) = phi(i) + ...
        sum(C.*cos(lambda*x(i)).*cosh(lambda*y(j)));
    end
end

%Produce Plots
%Plot 1-D ODE Solution
figure('Position',[50 200 500 400])
plot(x,phi);
xlabel('x (m)') % Label x-axis
ylabel('T (deg. C)') % Label y-axis
```

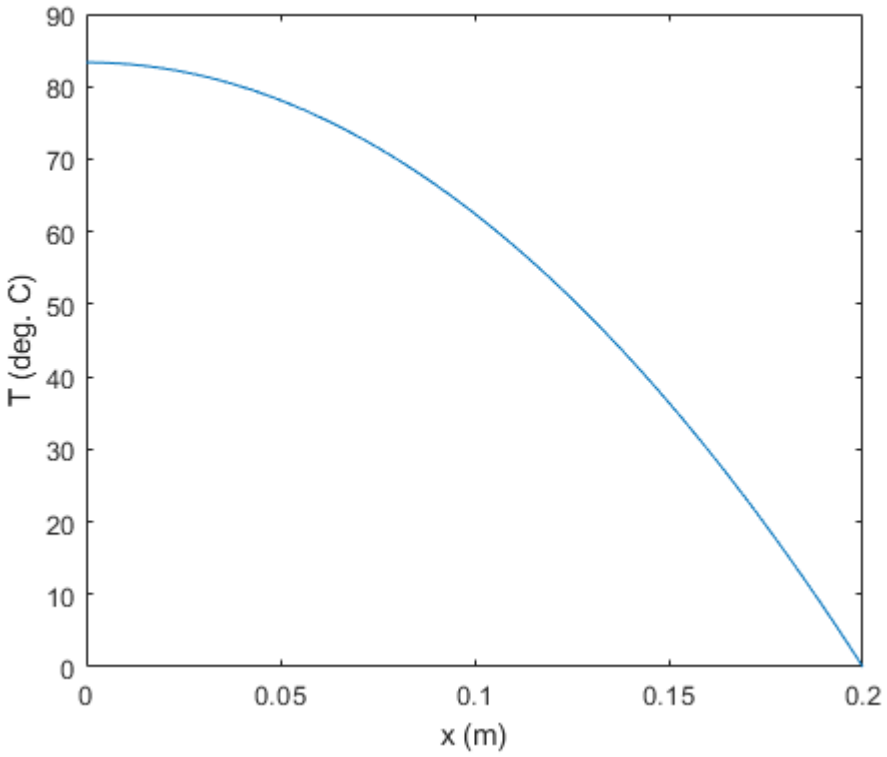


Figure 1. 1-D ODE solution $\Phi(x)$

```
figure('Position',[600 200 500 400])
contourf(x,y,T1, 'LevelStep', max(T1,[], 'all')/25);
colormap jet; % Set colors used for contours
colorbar % Add a scale to the plot
axis([0 a 0 b]) % Set axis limits
xlabel('x (m)') % Label x-axis
ylabel('y (m)') % Label y-axis
axis equal tight % Make x and y axes to scale
```

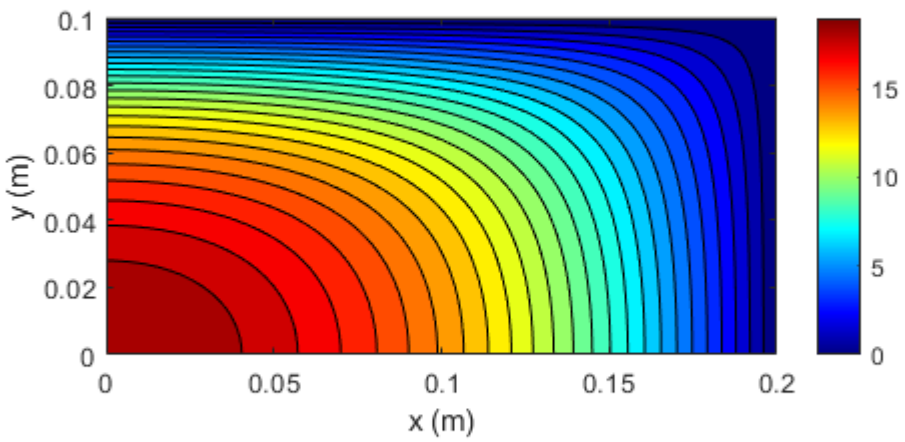


Figure 2. 2-D PDE solution $T(x, y)$

Determine How Many Terms in the Solution are Needed to Get and Accurate Result

The top right corner, $T(a, b)$, and the center of the plate, $T(\frac{a}{2}, \frac{b}{2})$, are chosen locations to calculate the percent error in the solution. Plots of percent error vs the number of Fourier coefficients are generated and discussed.

```
M = 20; % Maximum number of summation terms

%First, choose top-right corner as the location to compute errors
i = N
```

```
i = 101
```

```
j = N
```

```
j = 101
```

```

% 1-D ODE solution at top right corner
phi = Tmax*(1 - (x(i)/a)^2);

% Initialize percent error vector for the top right corner
error1 = zeros(M-1,1);

%Reference temperature for error calculation in the center of the plate. This is the value
%computed for the first 200 coefficients in the Fourier series.
T_ref_1 = T1(i,j);
```

```
fprintf('Percent Error at: x = %4.2f m, y = %4.2f m:\n',x(i),y(i));
```

```
Percent Error at: x = 0.20 m, y = 0.10 m:
```

```
for m = 1:M
    Txy = phi + ...
    sum(C(1:m).*cos(lambda(1:m)*x(i)).*cosh(lambda(1:m)*y(j)));
    error1(m) = 100*(T_ref_1-Txy)/T_ref_1;
    fprintf('%10.2e \n',error1(m));
end
```

```
  2.47e+01
  1.63e+01
  4.60e+00
  3.07e+00
  5.13e+00
  2.87e+00
  3.42e+00
  2.44e+00
  2.62e+00
  2.09e+00
  2.16e+00
  1.84e+00
  1.86e+00
  1.64e+00
  1.64e+00
  1.49e+00
  1.24e+00
  1.13e+00
  1.11e+00
  1.03e+00
```

```
%Second, choose the center as the location to compute errors
i =round(N/2)
```

```
i = 51
```

```
j =round(N/2)
```

```
j = 51
```

```
% 1-D ODE solution at top right corner
phi = Tmax*(1 - (x(i)/a)^2);

% Initialize percent error vector for the top right corner
error2 = zeros(M-1,1);

%Reference temperature for error calculation in the center of the plate. This is the value
%computed for the first 200 coefficients in the Fourier series.
T_ref_2 = T1(i,j);

fprintf('Percent Error at: x = %4.2f m, y = %4.2f m:\n',x(i),y(i));
```

```
Percent Error at: x = 0.10 m, y = 0.05 m:
```

```
for m = 1:M
```

```
    Txy = phi + ...
    sum(C(1:m).*cos(lambda(1:m)*x(i)).*cosh(lambda(1:m)*y(j)));
    error2(m) = 100*(T_ref_2-Txy)/T_ref_2;
    fprintf('%10.2e \n',error2(m));
end
```

```
 -5.47e+00
  6.34e-01
  6.93e-02
 -2.31e-02
 -3.32e-03
  1.61e-03
  2.53e-04
 -1.51e-04
 -2.46e-05
  1.67e-05
  2.78e-06
 -2.07e-06
 -3.47e-07
  2.76e-07
  4.66e-08
 -3.89e-08
 -6.61e-09
  5.74e-09
  9.78e-10
 -8.78e-10
```

Plot the percent error at each location as a function fo the number of terms

```
figure
plot(1:M,error1)
xlabel('Number of Fourier Coefficients')
ylabel('Percent error')
```

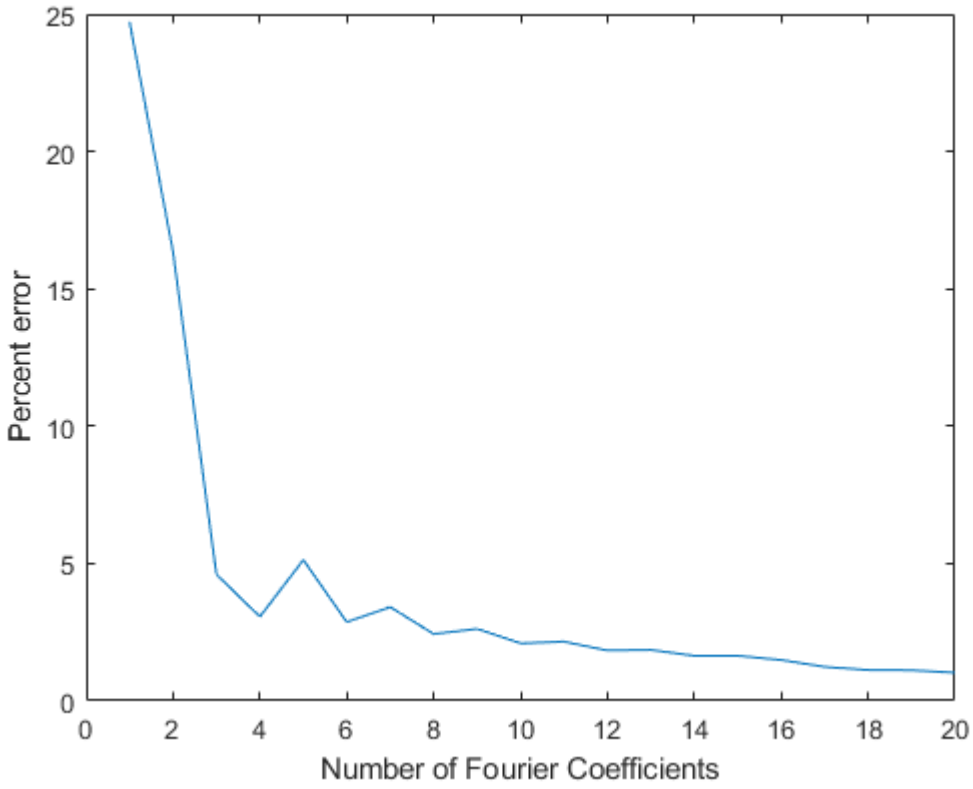


Figure 3. Percent error versus the number of Fourier coefficients for the top right corner, at $x = a$ and $y = b$.

```
figure
plot(1:M,error2)
xlabel('Number of Fourier Coefficients')
ylabel('Percent error')
```

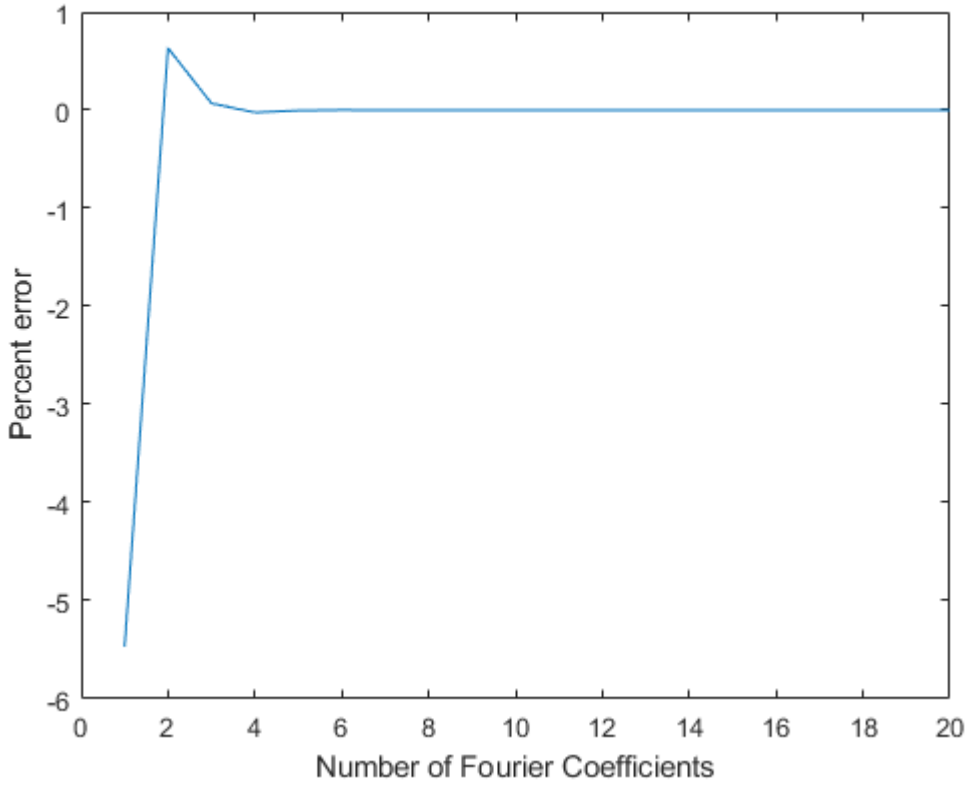


Figure 4. Percent error versus the number of Fourier coefficients for the center, at $x = \frac{a}{2}$ and $y = \frac{b}{2}$.

Update Thermal Conductivity and Reproduce Plots

```
%Update k
k = 2*240; % [W/(m*K)], new thermal conductivity


%Calculate Eigenvalues and Fourier Coefficients again
C = 2*g0*(-1).^n./(a*lambda.^3*k.*cosh(lambda*b));


%T(x,y) evaluated at the domain variables
T2 = zeros(N); % Initialize 2-D PDE solution
Tmax = g0*a^2/(2*k); % deg. C, Maximum T for 1-D ODE problem
for i = 1:N
phi(i) = Tmax*(1 - (x(i)/a)^2);
    for j = 1:N
        T2(i,j) = phi(i) + ...
        sum(C.*cos(lambda*x(i)).*cosh(lambda*y(j)));
    end
end

%Produce Plots
%Plot 1-D ODE Solution
figure('Position',[50 200 500 400])
```

```
plot(x,phi);
xlabel('x (m)') % Label x-axis
ylabel('T (deg. C)') % Label y-axis
```

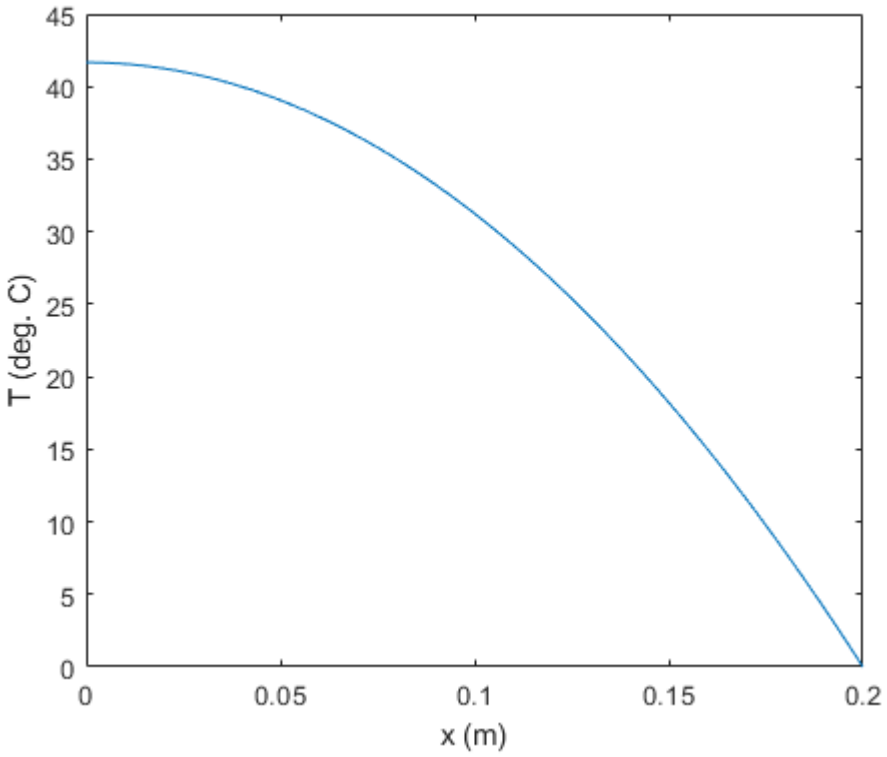


Figure 5. 1-D ODE solution $\Phi(x)$ for new k value.

```
figure('Name','2-D PDE Solution','Position',[600 200 500 400])
contourf(x,y,T2, 'LevelStep', max(T2,[], 'all')/25);
colormap jet; % Set colors used for contours
colorbar % Add a scale to the plot
axis([0 a 0 b]) % Set axis limits
xlabel('x (m)') % Label x-axis
ylabel('y (m)') % Label y-axis
axis equal tight % Make x and y axes to scale
```

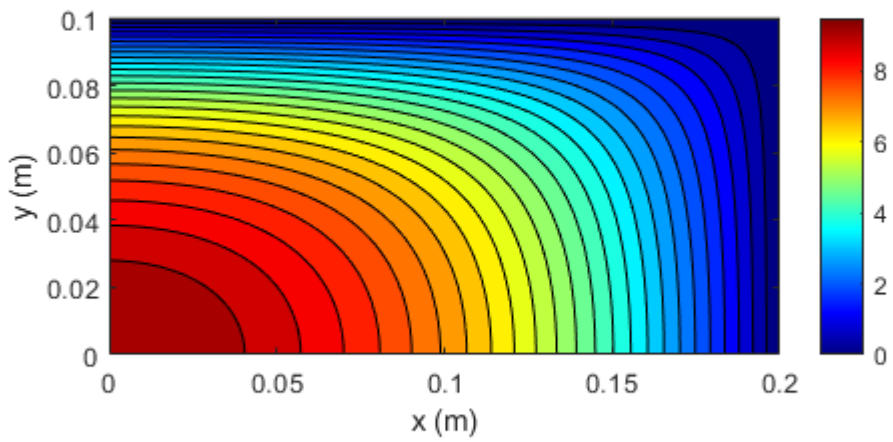


Figure 6. 2-D PDE solution $T(x, y)$ for new k value.

Discussion

It is clear from the percent error plots that the solution at the center converges much faster than the solution at the top right corner of the plate. Typically, an uncertainty of $\pm 1^{\circ}C$ is acceptable for engineering applications. For both cases, the first term in the Fourier expansion is within this tolerance.

```
fprintf('T(a,b) is off by %e degrees celcius at the first Fourier coefficient.', error1(1)/100
```

```
T(a,b) is off by -1.728760e-15 degrees celcius at the first Fourier coefficient.
```

```
fprintf('T(a/2,b/2) is off by %e degrees celcius at the first Fourier coefficient.', error2(1),
```

```
T(a/2,b/2) is off by -6.742317e-01 degrees celcius at the first Fourier coefficient.
```

The solution $T(x, y)$ appears correct. Figure 2 shows that the boundaries at $x = a$ and $y = b$ are very close to 0 $^{\circ}C$, and the boundaries at $x = 0$ and $y = 0$ have temperature gradients of 0 (the contour lines intersect those walls at 90°). Figures 5 and 6 show expected behavior for a material with double the thermal conductivity of aluminum (at STP). Since the thermal conductivity factors out of every term in the solution, all temperatures are exactly half those of the original solution.

```
T1./T2
```

```
ans = 101×101
     2     2     2     2     2     2     2     2     2     2     2     2     2 ⋯
     2     2     2     2     2     2     2     2     2     2     2     2     2
     2     2     2     2     2     2     2     2     2     2     2     2     2
     2     2     2     2     2     2     2     2     2     2     2     2     2
     2     2     2     2     2     2     2     2     2     2     2     2     2
     2     2     2     2     2     2     2     2     2     2     2     2     2
     2     2     2     2     2     2     2     2     2     2     2     2     2
     2     2     2     2     2     2     2     2     2     2     2     2     2
     2     2     2     2     2     2     2     2     2     2     2     2     2
     2     2     2     2     2     2     2     2     2     2     2     2     2
     ⋮
```

Problem 3-1 (continued)

Original Problem

Original Problem: A 1-D slab, $0 \le x \le L$, is initially at a temperature $F(x)$. For times $t > 0$, the boundary surface at $x = 0$ is kept at zero temperature, and the boundary surface at $x = L$ dissipates heat by convection into a medium with fluid temperature of zero and with a heat transfer coefficient h. Obtain an expression for the temperature distribution $T(x,t)$ in the slab for times $t > 0$, and for the heat flux at the boundary surface at $x = L$.

Parameter Specifications (where properties are for aluminum at STP):

$$L = 0.100 \ [m] \qquad \rho = 240 \ [\frac{kg}{m^3}]$$

$$T_0 = 100 \ [^\circ C] \qquad c = 900 \ [\frac{J}{kg^\circ C}]$$

$$F(x) = T_0 sin(\frac{\pi x}{L}) \qquad k = 240 \ [\frac{W}{m^\circ C}]$$

$$h = 10 \ [\frac{W}{m^{2\circ} C}]$$

Fourier Coefficients and Norm

Verify that the Fourier coefficients after intregration are as follows:

$$C_n = \frac{T_0}{N(\lambda_n)} \frac{\pi L sin(\lambda_n L)}{[\pi^2 - (\lambda_n L)^2]} \ \ for \ \ n = 1, 2, 3...$$

where

$$N(\lambda_n) = (\frac{L}{2})\left[\frac{(\lambda_n L)^2 + Bi^2 + Bi}{(\lambda_n L)^2 + Bi^2}\right] \quad for \;\; n = 1, 2, 3...$$

Verification:

Midterm Problem 3-1 Hand Calculations Fourier Coefficients and Norm - Dominic Riccoboni

Saturday, October 23, 2021 10:06 PM

FOURIER COEFFICIENTS

$$C_n = \frac{\int_0^L T_0 \sin(\frac{\pi x}{L}) \sin(\lambda_n x)\, dx}{N(\lambda_n)}$$

TRIG IDENTITY:

$$\sin(A)\sin(B) = \frac{1}{2}\left[\cos(A-B) - \cos(A+B)\right]$$

$$C_n = \frac{T_0}{2N(\lambda_n)}\left[\int_0^L \cos((\frac{\pi}{L} - \lambda_n)x)\, dx - \int_0^L \cos((\frac{\pi}{L} + \lambda_n)x)\, dx\right]$$

$$C_n = \frac{T_0}{2N(\lambda_n)}\left[\frac{1}{(\frac{\pi}{L} - \lambda_n)} \sin((\frac{\pi}{L} - \lambda_n)x)\Big|_0^L - \frac{1}{(\frac{\pi}{L} + \lambda_n)} \sin((\frac{\pi}{L} + \lambda_n)x)\Big|_0^L\right]$$

- $\sin((\frac{\pi}{L} - \lambda_n)x)\Big|_0^L = \sin(\pi - \lambda_n L)$

- $\sin\left(\left(\frac{\pi}{L}+\lambda_n\right)x\right)\Big|_0^L = \sin(\pi+\lambda_n L)$

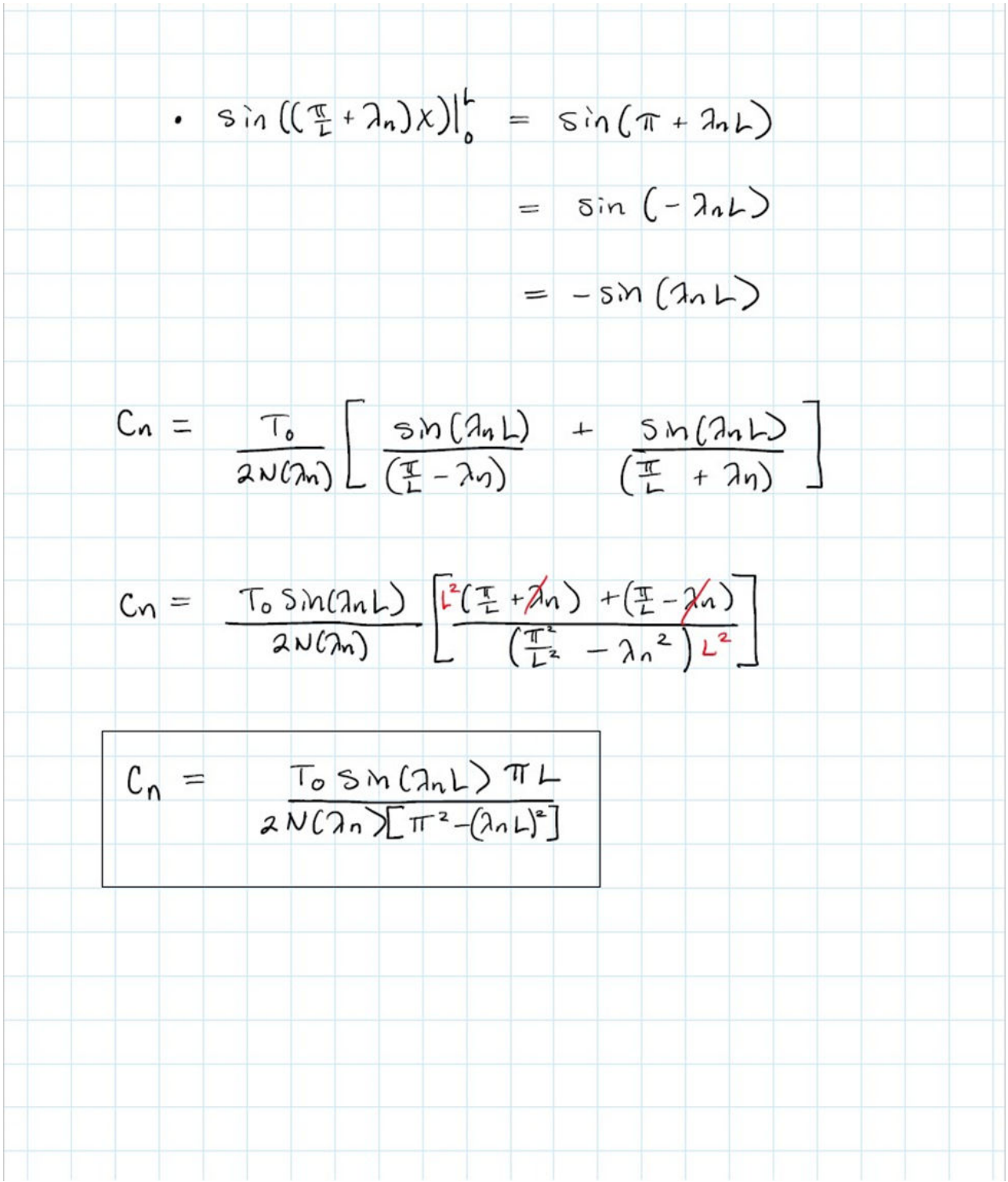
$= \sin(-\lambda_n L)$

$= -\sin(\lambda_n L)$

$$C_n = \frac{T_0}{2N(\lambda_n)}\left[\frac{\sin(\lambda_n L)}{\left(\frac{\pi}{L}-\lambda_n\right)} + \frac{\sin(\lambda_n L)}{\left(\frac{\pi}{L}+\lambda_n\right)}\right]$$

$$C_n = \frac{T_0 \sin(\lambda_n L)}{2N(\lambda_n)}\left[\frac{L^2\left(\frac{\pi}{L}+\lambda_n\right)+\left(\frac{\pi}{L}-\lambda_n\right)}{\left(\frac{\pi^2}{L^2}-\lambda_n^2\right)L^2}\right]$$

$$\boxed{C_n = \frac{T_0 \sin(\lambda_n L)\,\pi L}{2N(\lambda_n)\left[\pi^2-(\lambda_n L)^2\right]}}$$



Find Eigenvalues Using Root Finder

```
%System Parameters
L = 0.100; %[m] Length of rod
T0 = 100; %[deg C] Amplitude of initial temperature distribution
h = 10; %[W/(m^2*degC)] Convection coefficient
rho = 240; %[kg/m^3] Density
c = 900; %[J/(kg*degC)] Specific heat
k = 240; %[W/(m*degC)] Thermal conductivity
```

```
alpha = k/(rho*c); % [m^2/s] Thermal diffusivity
Bi = h*L/k; %Biot number

% Create an anonymous function for eigenvalue equation set equal
% to zero where fun_lambda is its “function handle”.
fun_lambda = @(lambda) (lambda*L).*cot(lambda*L) + Bi;

% Solve for eigenvalues using fzero() for specified intervals.
N = 200; % Number of eigenvalues
lambda = zeros(N,1); % Initialize eigenvalues

for n = 1:N
    lambda(n) = fzero(fun_lambda,[(n-0.99)*pi/L (n-0.01)*pi/L]);
end

%For each lambda, compute the norm and the Fourier coefficients
norm = (L/2)*((lambda*L).^2 + Bi^2 + Bi)./((lambda*L).^2 + Bi^2);
Cn = T0*pi*L*sin(lambda*L)./(norm.*(pi^2 - (lambda*L).^2));
```

Produce Solution Plot

```
Nx = 101; %Number of data points in the x direction
Nt = 6; %Number of points in time to plot
tmax = 10; %[s] Max time to plot solution for

%Allocate memory for solution's domain and range
x = linspace(0,L,Nx); % [m] x-locations for calculations
t = [linspace(0,tmax,Nt),200]; % [s] times for calculations
T = zeros(Nx,length(t)); %Temperature as function of x and t

%Evaluate analytical solution over the domain to produce T
for p = 1:Nt
    for i = 1:Nx
        T(i,p) = sum(Cn.*sin(lambda*x(i)).*exp(-alpha*lambda.^2*t(p)));
    end
end

%Produce plots of T vs x as various time values
figure('Name','Temperature versus Time and Location')
hold on % Turn on plot hold - want to plot T at various times on same axis
for p = 1:length(t)
    label = ['t = ', num2str(t(p),'%3.0f'),' s']; % Label string
    plot(x,T(:,p),'DisplayName',label) % Generate plot
end
axis([0 L -10 T0]) % Set axis limits
xlabel('x (m)') % Label x-axis
ylabel('Temperature (deg. C)') % Label y-axis
legend() % Add legend to plot
hold off % Turn off plot hold
```

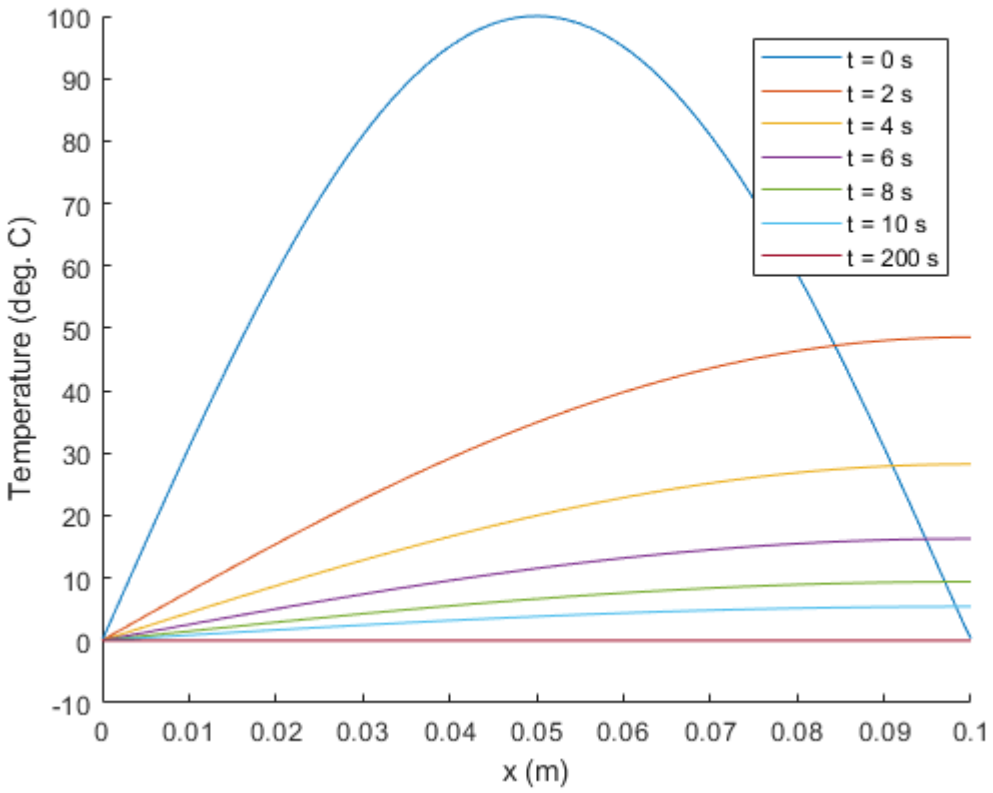


Figure 7. Temperature vs location along the rod for various moments in time.

Produce Plots for Altered Parameters

Three plots to produce:

1. Doubled conductivity
2. Half density
3. Half specific heat

Double the Conductivity

```
k = 2*240; %[W/(m*degC)] Thermal conductivity
rho = 240; %[kg/m^3] Density is the same
c = 900; %[J/(kg*degC)] Specific heat is the same
alpha = k/(rho*c); % [m^2/s] Thermal diffusivity
Bi = h*L/k; %Biot number

% Create an anonymous function for eigenvalue equation set equal
% to zero where fun_lambda is its “function handle”.
fun_lambda = @(lambda) (lambda*L).*cot(lambda*L) + Bi;

% Solve for eigenvalues using fzero() for specified intervals.
N = 200; % Number of eigenvalues
lambda1 = zeros(N,1); % Initialize eigenvalues

for n = 1:N
```

```
    lambda1(n) = fzero(fun_lambda,[(n-0.99)*pi/L (n-0.01)*pi/L]);
end

%For each lambda, compute the norm and the Fourier coefficients
norm = (L/2)*((lambda1*L).^2 + Bi^2 + Bi)./((lambda1*L).^2 + Bi^2);
Cn = T0*pi*L*sin(lambda1*L)./(norm.*(pi^2 - (lambda1*L).^2));

%Allocate memory for solution's domain and range
x = linspace(0,L,Nx); % [m] x-locations for calculations
t = [linspace(0,tmax,Nt),200]; % [s] times for calculations
T = zeros(Nx,length(t)); %Temperature as function of x and t

%Evaluate analytical solution over the domain to produce T
for p = 1:Nt
    for i = 1:Nx
        T(i,p) = sum(Cn.*sin(lambda1*x(i)).*exp(-alpha*lambda1.^2*t(p)));
    end
end

%Produce plots of T vs x as various time values
figure('Name','Temperature versus Time and Location')
hold on % Turn on plot hold - want to plot T at various times on same axis
for p = 1:length(t)
    label = ['t = ', num2str(t(p),'%3.0f'),' s']; % Label string
    plot(x,T(:,p),'DisplayName',label) % Generate plot
end
axis([0 L -10 T0]) % Set axis limits
xlabel('x (m)') % Label x-axis
ylabel('Temperature (deg. C)') % Label y-axis
legend() % Add legend to plot
hold off % Turn off plot hold
```

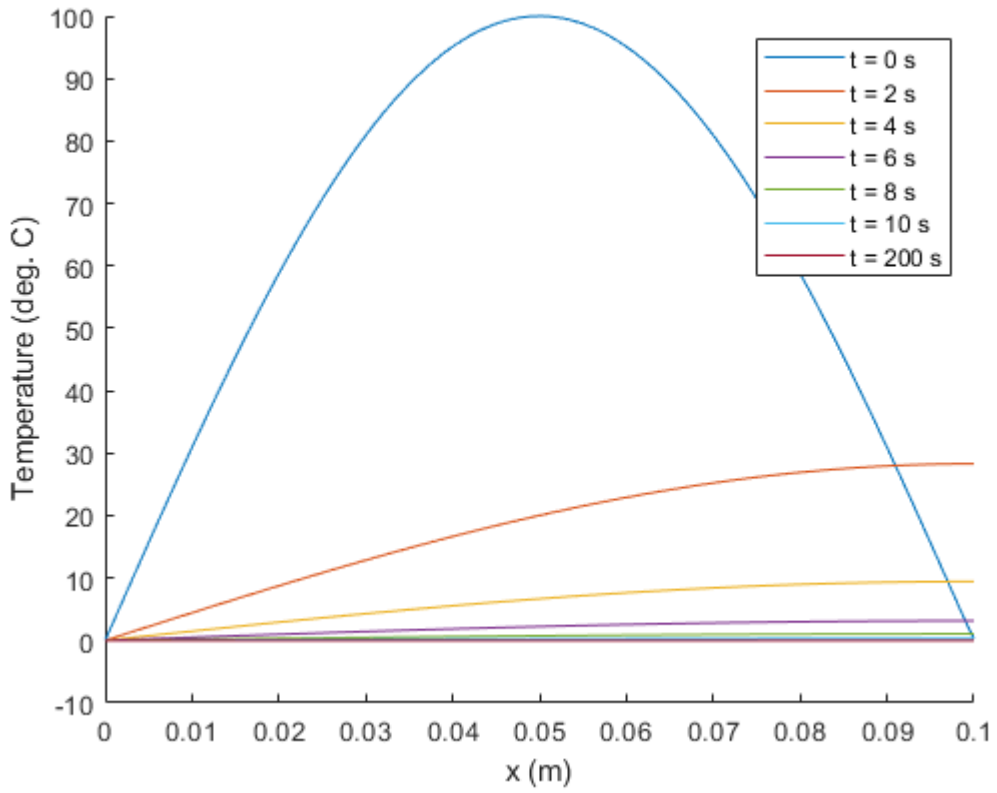


Figure 8. Temperature vs location along the rod for various moments in time. Conductivity has been doubled from figure 7 results.

Half the Density

```
k = 240; %[W/(m*degC)] Thermal conductivity set back to original value
rho = 240/2; %[kg/m^3] Density is halved
c = 900; %[J/(kg*degC)] Specific heat is the same
alpha = k/(rho*c); % [m^2/s] Thermal diffusivity recomputed
Bi = h*L/k; %Biot number recomputed with original k value

% Create an anonymous function for eigenvalue equation set equal
% to zero where fun_lambda is its “function handle”.
fun_lambda = @(lambda) (lambda*L).*cot(lambda*L) + Bi;

% Solve for eigenvalues using fzero() for specified intervals.
N = 200; % Number of eigenvalues
lambda2 = zeros(N,1); % Initialize eigenvalues

for n = 1:N
    lambda2(n) = fzero(fun_lambda,[(n-0.99)*pi/L (n-0.01)*pi/L]);
end

%For each lambda, compute the norm and the Fourier coefficients
norm = (L/2)*((lambda2*L).^2 + Bi^2 + Bi)./((lambda2*L).^2 + Bi^2);
Cn = T0*pi*L*sin(lambda2*L)./(norm.*(pi^2 - (lambda2*L).^2));
```

```
%Allocate memory for solution's domain and range
x = linspace(0,L,Nx); % [m] x-locations for calculations
t = [linspace(0,tmax,Nt),200]; % [s] times for calculations
T = zeros(Nx,length(t)); %Temperature as function of x and t

%Evaluate analytical solution over the domain to produce T
for p = 1:Nt
    for i = 1:Nx
        T(i,p) = sum(Cn.*sin(lambda2*x(i)).*exp(-alpha*lambda2.^2*t(p)));
    end
end

%Produce plots of T vs x as various time values
figure('Name','Temperature versus Time and Location')
hold on % Turn on plot hold - want to plot T at various times on same axis
for p = 1:length(t)
    label = ['t = ', num2str(t(p),'%3.0f'),' s']; % Label string
    plot(x,T(:,p),'DisplayName',label) % Generate plot
end
axis([0 L -10 T0]) % Set axis limits
xlabel('x (m)') % Label x-axis
ylabel('Temperature (deg. C)') % Label y-axis
legend() % Add legend to plot
hold off % Turn off plot hold
```

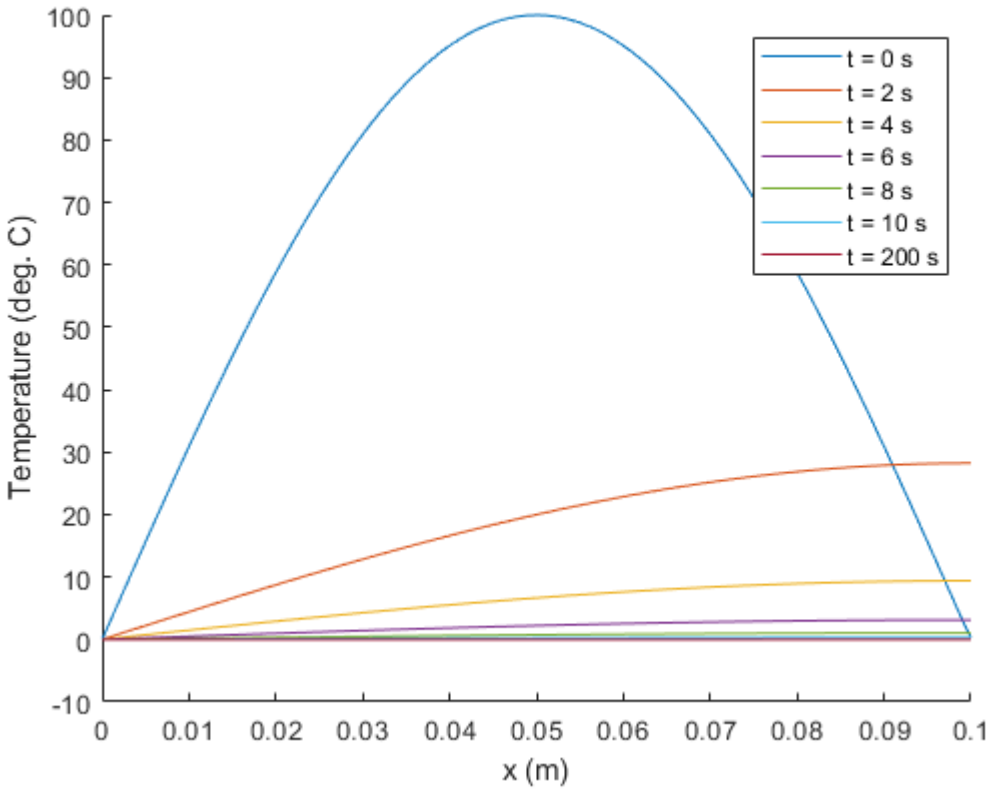


Figure 9. Temperature vs location along the rod for various moments in time. Density is half that of the figure 7 results.

Half Specific Heat

```
k = 240; %[W/(m*degC)] Thermal conductivity set back to original value
rho = 240; %[kg/m^3] Density is set back to original value
c = 900/2; %[J/(kg*degC)] Specific heat is halved
alpha = k/(rho*c); % [m^2/s] Thermal diffusivity recomputed
Bi = h*L/k; %Biot number recomputed with original k value

% Create an anonymous function for eigenvalue equation set equal
% to zero where fun_lambda is its “function handle”.
fun_lambda = @(lambda) (lambda*L).*cot(lambda*L) + Bi;

% Solve for eigenvalues using fzero() for specified intervals.
N = 200; % Number of eigenvalues
lambda3 = zeros(N,1); % Initialize eigenvalues

for n = 1:N
    lambda3(n) = fzero(fun_lambda,[(n-0.99)*pi/L (n-0.01)*pi/L]);
end

%For each lambda, compute the norm and the Fourier coefficients
norm = (L/2)*((lambda3*L).^2 + Bi^2 + Bi)./((lambda3*L).^2 + Bi^2);
Cn = T0*pi*L*sin(lambda3*L)./(norm.*(pi^2 - (lambda3*L).^2));

%Allocate memory for solution's domain and range
x = linspace(0,L,Nx); % [m] x-locations for calculations
t = [linspace(0,tmax,Nt),200]; % [s] times for calculations
T = zeros(Nx,length(t)); %Temperature as function of x and t

%Evaluate analytical solution over the domain to produce T
for p = 1:Nt
    for i = 1:Nx
        T(i,p) = sum(Cn.*sin(lambda3*x(i)).*exp(-alpha*lambda3.^2*t(p)));
    end
end

%Produce plots of T vs x as various time values
figure('Name','Temperature versus Time and Location')
hold on % Turn on plot hold - want to plot T at various times on same axis
for p = 1:length(t)
    label = ['t = ', num2str(t(p),'%3.0f'),' s']; % Label string
    plot(x,T(:,p),'DisplayName',label) % Generate plot
end
axis([0 L -10 T0]) % Set axis limits
xlabel('x (m)') % Label x-axis
ylabel('Temperature (deg. C)') % Label y-axis
legend() % Add legend to plot
hold off % Turn off plot hold
```

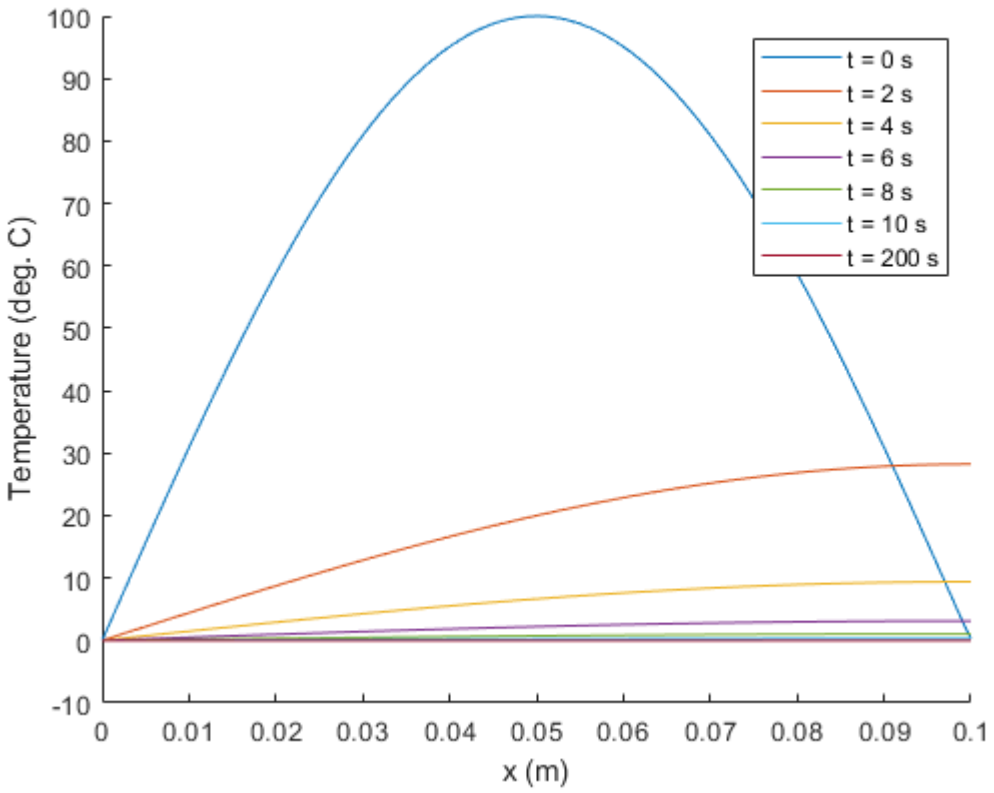


Figure 10. Temperature vs location along the rod for various moments in time. Specific heat is half that of the figure 7 results.

Discussion

Discuss the Validity of the Results for Original System Parameters:

The initial condition is satisfied - the zeros of $F(x) = T_0 sin(\frac{\pi x}{L})$ occur at $x = 0$ and $x = L$ for the problem domain.

The amplitude is also clearly present, $T_0 = 100\ [^\circ C]$, and the shape is sinusoidal by inspection. The steady state value at $t = 200\ [s]$ is also as expected. The entire rod reaches thermal equilibrium with the fluid at $T_\infty = 0$; the steady state value being $T_{ss} = T_\infty = 0$.

Discuss Why All Three Plots for the Altered System Parameters are the Same:

All three plots are the same because α is affected in the same way by each of the alterations. Since the governing equation's only parameter is this constant and the BC's and IC's are the same for each problem, the solutions are identical.

Discuss the Difference Between the New Results and the Original Results and if T is Valid:

When varying the system parameters, expected results are found. For a larger k, there is an increased efficiency of transport of thermal energy and the temperature gradients are increased. For smaller ρ or c, less internal energy can be stored for a given temperature gradient and therefore more energy transport occurs. In each case, and for each eigenvalue, the exponential decay is increased by an increase in α. This can be seen in:

$$\Gamma_n(t) = c_1 e^{-\alpha\lambda_n^2 t}$$

For smaller c and ρ, the eigenvalues are unaffected and therefore a smaller time constant is directly responsible for the faster response. However, for larger k, it is still expected that the steady state value is reached earlier, but since changing it affects the eigenvalues as well, the time constant is not so easily ascertained. For this case, a combination of a change in the Fourier coefficients and basis functions along with the altered time constant, is responsible for the decreased time to steady state.