ME201/MTH281/ME400/CHE400
Heat Flow in a Slab
Mathematica 7

■ 1. Introduction

In this notebook, we use *Mathematica* to visualize the solutions obtained in class for transient heat conduction in a slab. In the first example, the boundary conditions are zero temperature on both faces of the slab, and the initial temperature is uniform. The formulation of the boundary value problem is given below.

$$\frac{\partial T}{\partial t} = D_f \frac{\partial^2 T}{\partial x^2}, \; T(x,0) = T_0, \; T(0,t) = 0, \text{ and } T(L,t) = 0 . \qquad (1)$$

In class we used separation of variables and a Fourier sine series to solve this problem. The result is

$$T(x,t) = \sum_{n=1}^{\infty} \frac{4 T_0}{(2n-1)\pi} e^{-(2n-1)^2 \pi^2 D_f t/L^2} \sin((2n\text{-}1)\pi x/L) \; . \qquad (2)$$

The second example, there is a zero flux boundary condition at each end of the slab, and the initial temperature varies linearly from zero at the left of the slab to $2T_0$at the right of the slab. The formulation of the boundary value problem is given below.

$$\frac{\partial T}{\partial t} = D_f \frac{\partial^2 T}{\partial x^2}, \; T(x,0) = 2T_0(x/L), \; \frac{\partial T}{\partial x}(0,t) = 0, \text{ and } \frac{\partial T}{\partial x}(L,t) = 0 . \qquad (3)$$

The solution obtained in class by separation of variables is

$$T(x,t) = T_0 - \sum_{n=1}^{\infty} \frac{8 T_0}{(2n-1)^2 \pi^2} e^{-(2n-1)^2 \pi^2 D_f t/L^2} \cos((2n\text{-}1)\pi x/L) \; . \qquad (4)$$

We will use the formulas (2) and (4) to construct graphs of the solutions as a function of *x* for various times *t*. In each graph, we will show the initial condition, the "exact" solution (obtained by keeping 10 terms of the series), and the long-time approximation, obtained by keeping only the first term of the series. The initial condition is dashed, the exact solution is red, and the long-time approximation is blue.

■ 2. Values of Parameters

As in class, we consider an aluminum slab of thickness 0.1 m. We take T_0 to be 100 °C, (the initial temperature for the first problem and the average temperature for the second problem) and we use the value of 70 x 10^{-6}m^2/s for the thermal diffusivity of aluminum.

```
To = 100.0;(** °C **)
```

```
Df = 70.0 * 10^-6;(** m^2/s **)
```

The thickness of the slab is taken to be 0.1 m.

```
L = 0.1 (** m **);
```

For this set of parameters, the basic diffusion time (in seconds) is

```
tau = L^2/(π^2 Df) (** Diffusion Time **)
```

$$tau = \frac{L^2}{\pi^2 Df}$$

```
14.4745
```

In the first problem, this is the time over which considerable cooling of the slab takes place. In the second problem, it is the time over which considerable internal rearrangement of heat enegy takes place. We will make graphs for times up to 20 seconds.

3. Example With Zero Temperature Boundary Conditions

3a. Definition of Functions and Graphs

```
SetOptions[Plot, ImageSize → 250];
```

The initial temperature is given by

```
f[x_] := To
```

The function fs[x,t,n], defined below, gives the nth partial sum of the solution, as a function of x and t.

```
fs[x_,t_,n_]:=((4*To)/π) Sum[(Exp[-(Df*(((2i-1)*π)/L)^2)*t])
(Sin[(π*(2 i - 1)*x)/L]/(2 i - 1)),{i,1,n}]
```

The function b[t, j], defined below, produces a composite graph. The graph shows the initial distribution in red, the solution calculated from j terms of the series solution in blue (solid), and the long-time approximation (the first term only of the series) in blue (dashed). The function b[0, j] is just the special form of this function for the initial time.

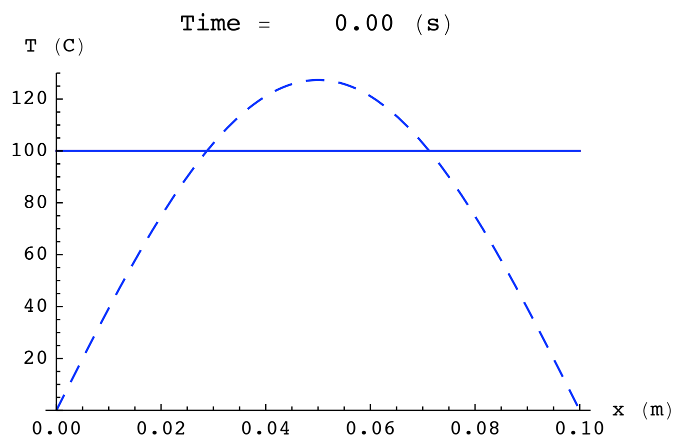
```
 b[0.,j_]:=Plot[{fs[x,0,1],f[x],f[x]},{x,0,L},PlotStyle->
{{RGBColor[0,0,1],Dashing[{0.03}],Thickness[0.004]},{RGBColor[1,0,0],Thickness[
PlotRange->{0,1.3*To},AxesLabel->{"x (m)","T (C)"},
PlotLabel -> Row[{"Time = ",PaddedForm[0,{5,2}]," (s)"}]]
```

```
 b[t_,j_]:=Plot[{fs[x,t,1],f[x],fs[x,t,j]},{x,0,L},PlotStyle->
{{RGBColor[0,0,1],Dashing[{0.03}],Thickness[0.004]},{RGBColor[1,0,0],Thickness[
PlotRange->{0,1.3*To},AxesLabel->{"x (m)","T (C)"},
PlotLabel -> Row[{"Time = ",PaddedForm[t,{5,2}]," (s)"}]]
```

■ 3b. Graph Sequence

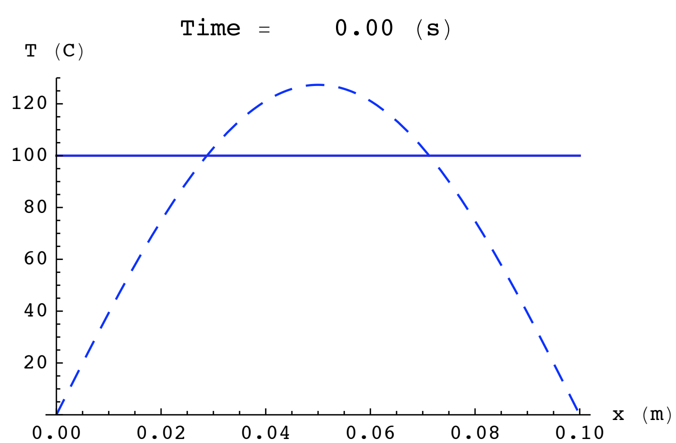
We use a Do loop to construct a sequence of graphs showing the time evolution of the cooling process. We plot the solution at every 0.25 s, starting at $t = 0$, and ending at $t = 20$ seconds. You can animate the graph sequence by selecting it and then using the menu sequence **Graphics -> Rendering -> Animate Selected Graphics**. In the printed version of this notebook, only the first graph of the sequence is shown.

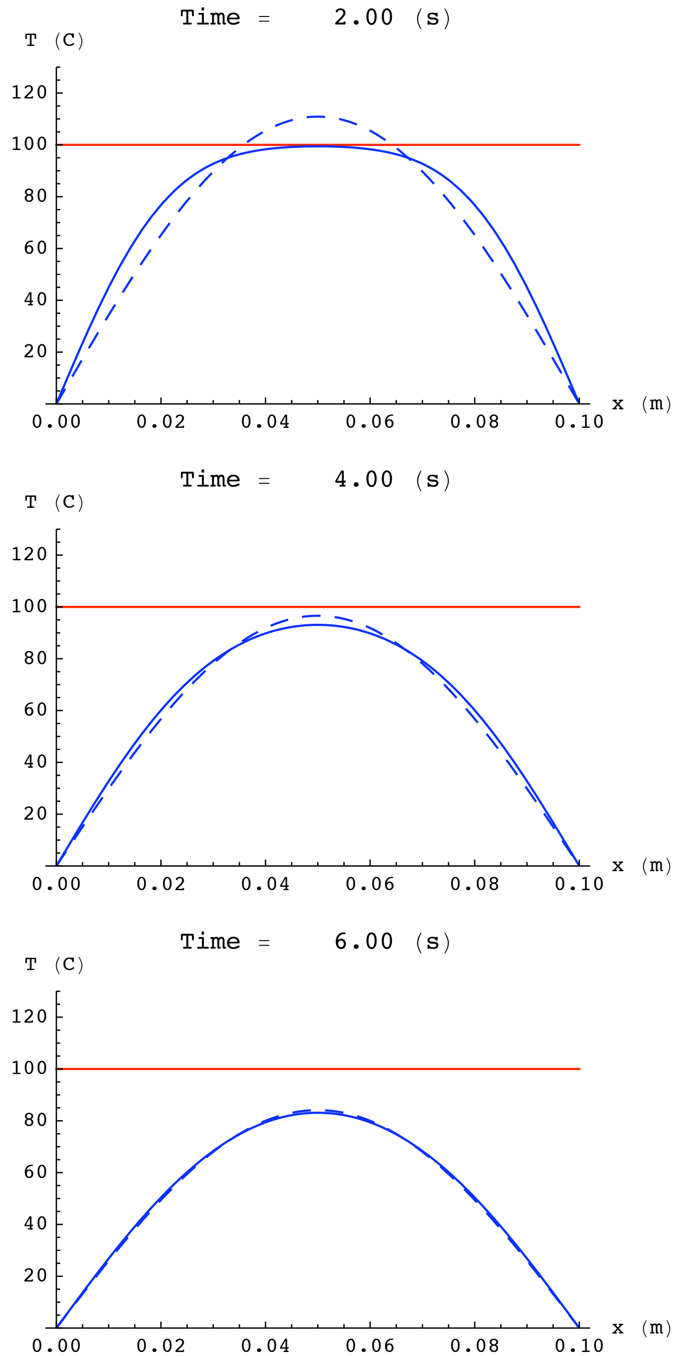
```
Do[Print[b[n,10]],{n,0,20,0.25}];
```

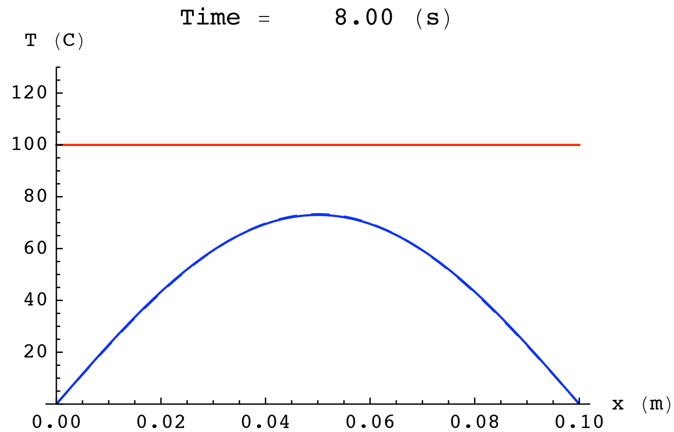


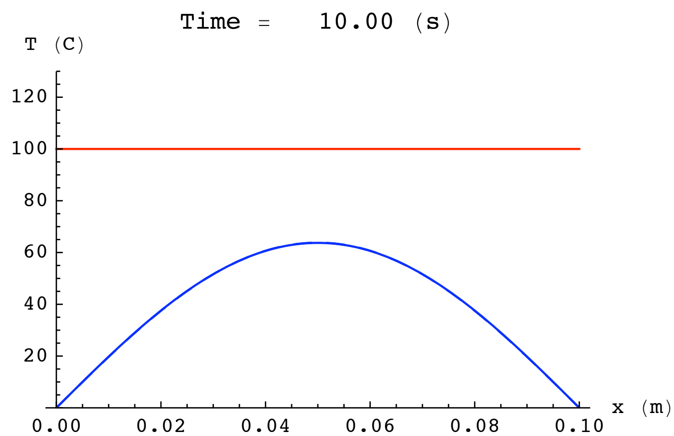
For visualization in the printed version of the notebook, we construct a sequence of solution graphs at time intervals of 2 s.

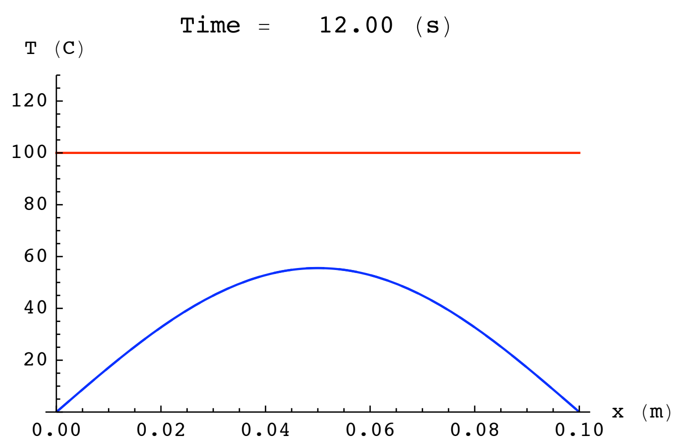
```
Do[Print[b[n,10]],{n,0,20,2.}];
```

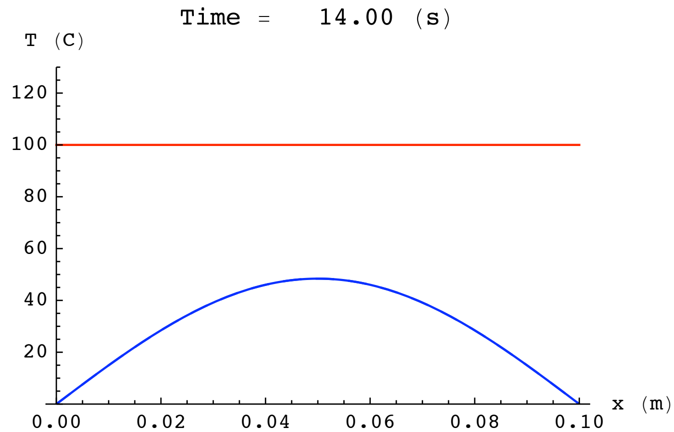


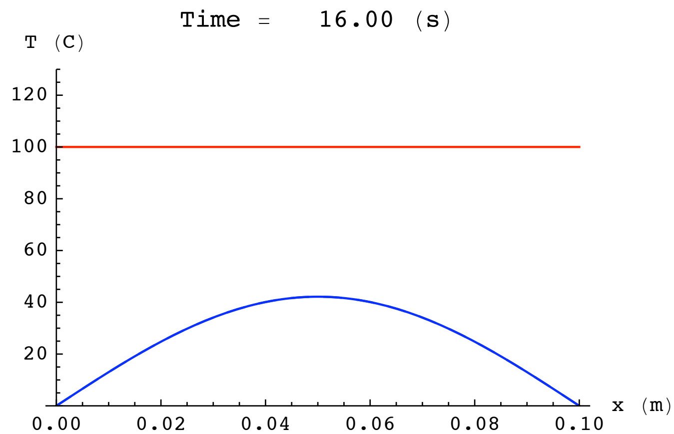

Time = 2.00 (s)
T (C)
120
100
80
60
40
20
0.00
0.02
0.04
0.06
0.08
0.10
x (m)
Time = 4.00 (s)
T (C)
120
100
80
60
40
20
0.00
0.02
0.04
0.06
0.08
0.10
x (m)
Time = 6.00 (s)
T (C)
120
100
80
60
40
20
0.00
0.02
0.04
0.06
0.08
0.10
x (m)

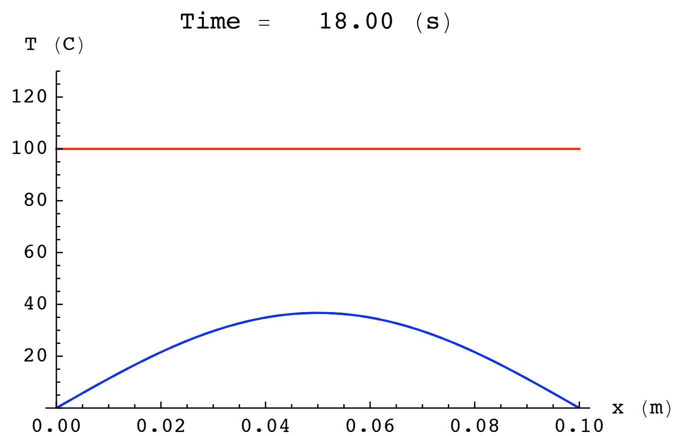

Time = 8.00 (s)
T (C)
120
100
80
60
40
20
0.00
0.02
0.04
0.06
0.08
0.10
x (m)

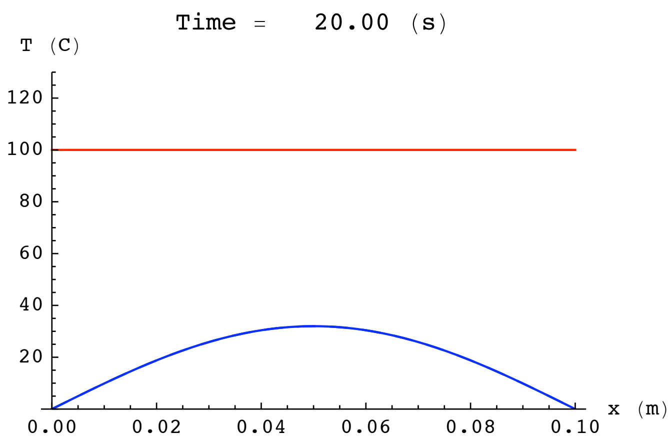

Time = 10.00 (s)
T (C)
120
100
80
60
40
20
0.00
0.02
0.04
0.06
0.08
0.10
x (m)


Time = 12.00 (s)
T (C)
120
100
80
60
40
20
0.00
0.02
0.04
0.06
0.08
0.10
x (m)


Time = 14.00 (s)
T (C)
120
100
80
60
40
20
0.00
0.02
0.04
0.06
0.08
0.10
x (m)


Time = 16.00 (s)
T (C)
120
100
80
60
40
20
0.00
0.02
0.04
0.06
0.08
0.10
x (m)


Time = 18.00 (s)
T (C)
120
100
80
60
40
20
0.00
0.02
0.04
0.06
0.08
0.10
x (m)



3c. Discussion of the Graph Sequence

Consider first the red curve, showing the exact solution. At the early times, heat has been lost only in the regions nearest the cool boundaries. For those times the central part of the red curve still coincides with the initial curve. For later times, the temperature has dropped throughout the slab, with the largest drops occurring near the boundaries. The distinction between small and large times is determined qualitatively by the diffusion time $L^2/(\pi^2 D_f)$, which we calculated above to be 14.5 seconds. The graph sequence is consistent with this. For times around 14 seconds or longer, the drops in temperature are appreciable throughout the slab. For times much smaller than this, the drops are confined to the boundary regions.

The blue curve shows the first term only of the series. As we saw in class, this should approximate the exact solution for large times. The graph sequence shows this. For the larger times, the blue curve and the red curve coincide completely. By animating the graph sequence, you can see that the approximation provided by the blue curve is quite good for any time larger than about 5 seconds.

4. Example With Zero Flux Boundary Conditions

4a. Definition of Functions and Graphs

The initial temperature is given by

```
f[x_] := 2 * To * (x / L)
```

The function fs[x,t,n], defined below, gives the nth partial sum of the solution, as a function of x and t.

```
fs[x_,t_,n_]:=To-((8*To)/π^2) Sum[(Exp[-(Df*(((2i-1)*π)/L)^2)*t])
(Cos[(π*(2 i - 1)x)/L]/(2 i - 1)^2),{i,1,n}]
```

The function b[t, j], defined below produces a composite graph. The graph shows the initial distribution dashed, the solution calculated from the series solution, using j terms, in red, and the long-time approximation (the first term only of the series) in blue. The function b[0,j] is just the special form of this function for the initial time.

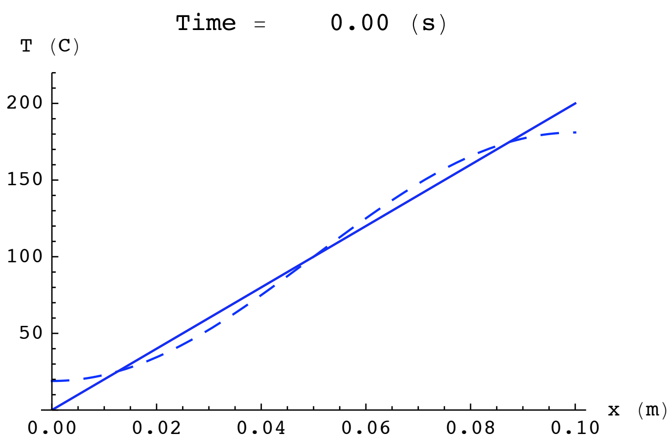
```
b[0.,j_]:=Plot[{fs[x,0,1],f[x],f[x]},{x,0,L},PlotStyle->
{{RGBColor[0,0,1],Dashing[{0.03}],Thickness[0.004]},{RGBColor[1,0,0],Thickness[
PlotRange->{0,2.2*To},AxesLabel->{"x (m)","T (C)"},
PlotLabel -> Row[{"Time = ",PaddedForm[0,{5,2}]," (s)"}]]

b[t_,j_]:=Plot[{fs[x,t,1],f[x],fs[x,t,j]},{x,0,L},PlotStyle->
{{RGBColor[0,0,1],Dashing[{0.03}],Thickness[0.004]},{RGBColor[1,0,0],Thickness[
PlotRange->{0,2.2*To},AxesLabel->{"x (m)","T (C)"},
PlotLabel -> Row[{"Time = ",PaddedForm[t,{5,2}]," (s)"}]]
```

■ 4b. Graph Sequence

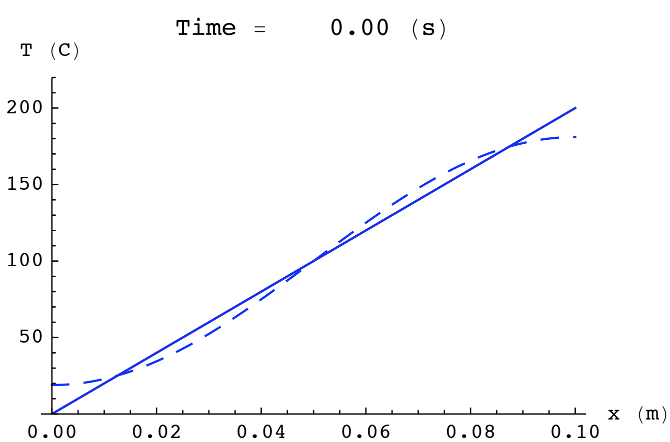
We use a Do loop to construct a sequence of graphs showing the time evolution of the cooling process. We plot the solution at every 0.25 s, starting at t = 0, and ending at t = 20 seconds. You can animate the graph sequence by selecting it and then using the menu sequence **Graphics -> Rendering -> Animate Selected Graphics**. In the printed version of this notebook, only the first graph of this sequence is shown.

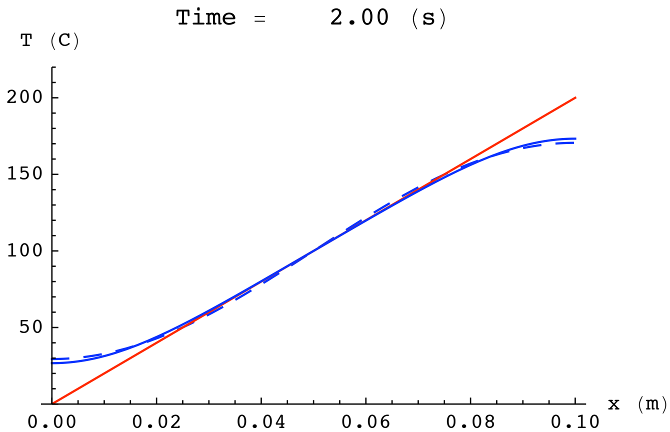
```
Do[Print[b[n,10]],{n,0,20,0.25}];
```

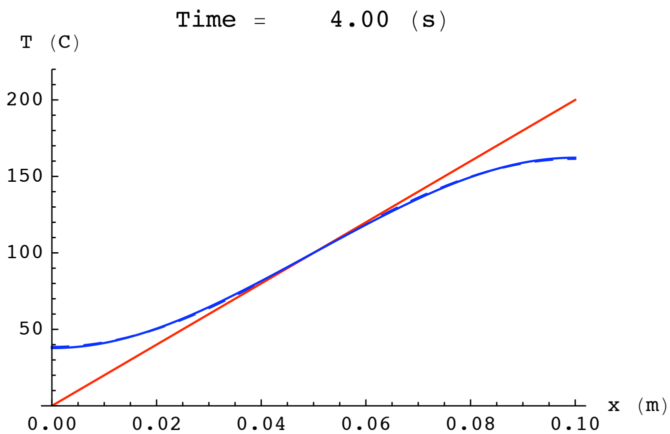


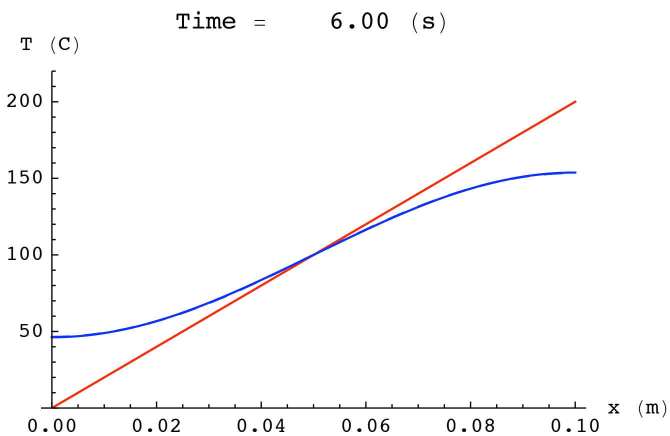
For visualization in the printed version of this notebook, we construct an abbreviated sequence from t = 0 to t = 20 at intervals of 2 s.

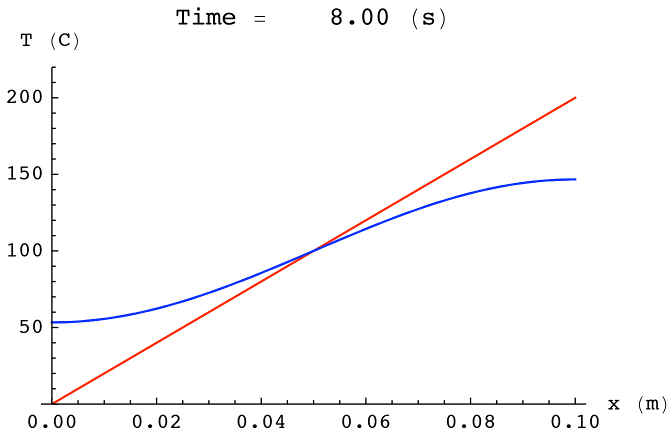
```
Do[Print[b[n,10]],{n,0,20,2.}];
```

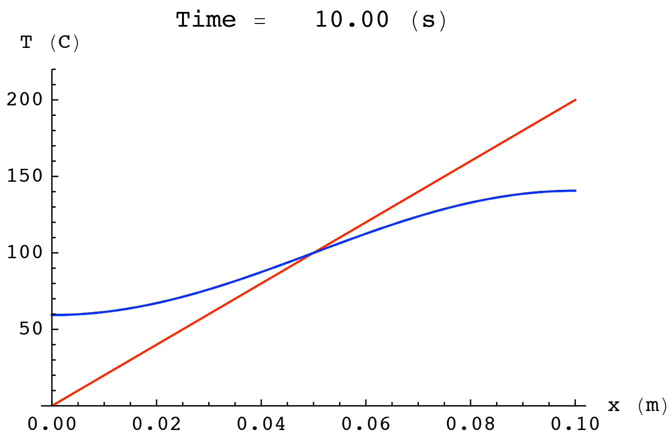


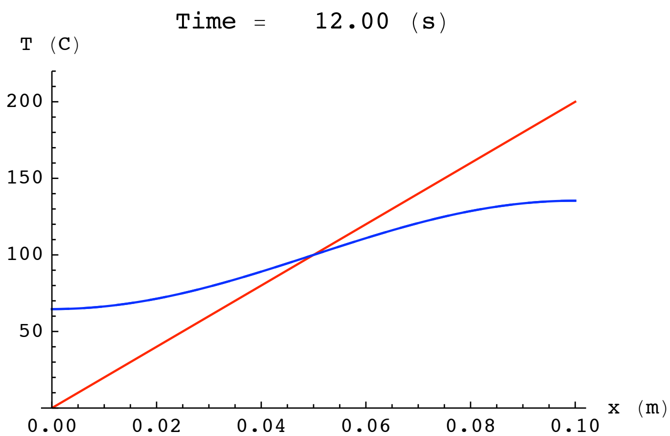

Time = 2.00 (s)
T (C)
200
150
100
50
0.00
0.02
0.04
0.06
0.08
0.10
x (m)


Time = 4.00 (s)
T (C)
200
150
100
50
0.00
0.02
0.04
0.06
0.08
0.10
x (m)

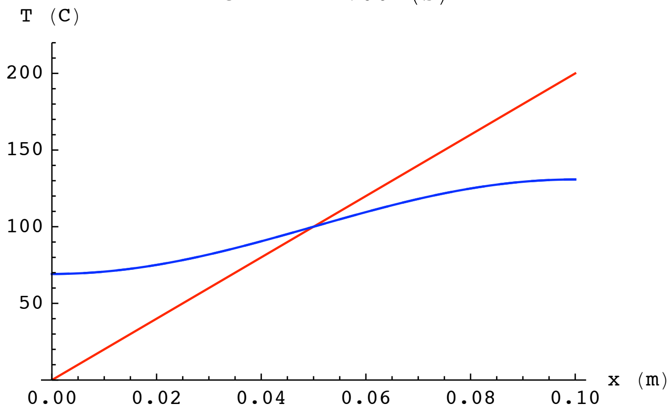

Time = 6.00 (s)
T (C)
200
150
100
50
0.00
0.02
0.04
0.06
0.08
0.10
x (m)


Time = 8.00 (s)
T (C)
200
150
100
50
x (m)
0.00
0.02
0.04
0.06
0.08
0.10

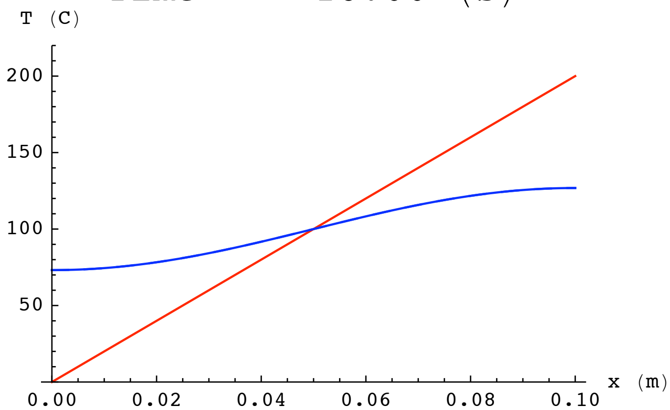

Time = 10.00 (s)
T (C)
200
150
100
50
x (m)
0.00
0.02
0.04
0.06
0.08
0.10


Time = 12.00 (s)
T (C)
200
150
100
50
x (m)
0.00
0.02
0.04
0.06
0.08
0.10

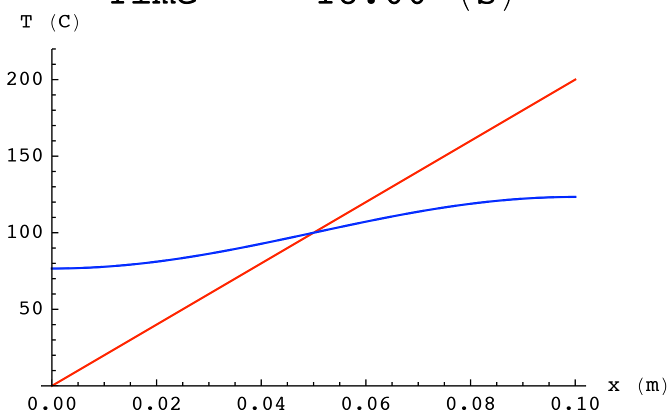
Time = 14.00 (s)

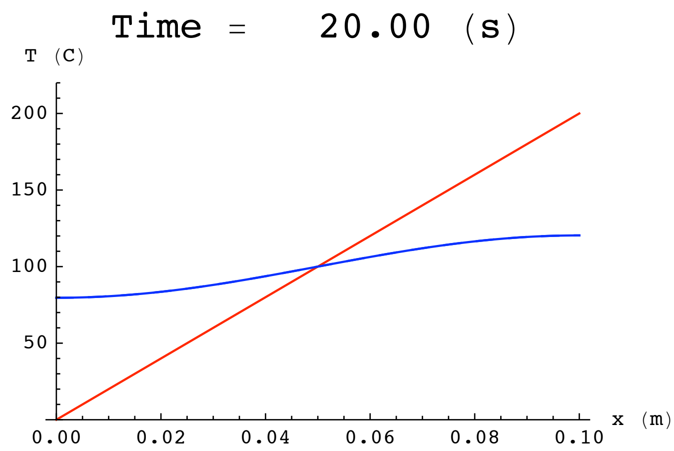


Time = 16.00 (s)



Time = 18.00 (s)

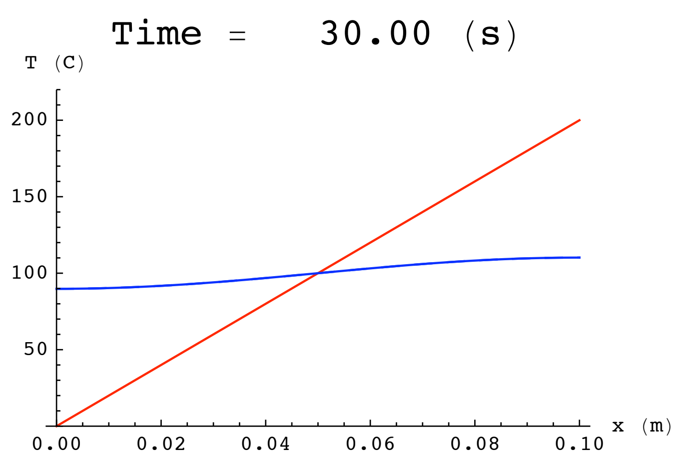




■ 4c. Discussion of the Graph Sequence

The red curve, showing the exact solution, exhibits the zero flux boundary condition (zero slope) at the sides of the slab. The temperature makes a gradual transition from the initial linear distribution toward the final steady state of a constant temperature throughout the slab. The blue curve, representing the constant term plus the first non-constant term in the series, essentially coincides with the exact solution (red curve) after about 5 seconds. The characteristic diffusion time $L^2/\pi^2 D_f$ was calculated above to be 14.5 s. It's significance is that the equilibration has progressed substantially when that amount of time has elapsed. We can see the truth of that from the above graph sequence. For the equilibration to appear complete on the graphs, we would have to go beyond 20 s. We check this by looking at a few graphs for longer times.

```
b[30, 10]
```



```
b[50, 10]
```