ME 201/MTH 281 ASSIGNMENT #7 SOLUTIONS OCT 30, 2008

(1) (a) Let $T_s(x)$ be the steady-state solution. The equation is

$$0 = D_f \frac{d^2 T_s}{dx^2}$$

so $T_s = Ax + B$. The boundary condition at $x=0$ gives

$$\frac{dT_s}{dx}(0) = 0 \Rightarrow A = 0.$$

The boundary condition at L is

$$k\frac{dT_s}{dx} + h[T_s - T_A] = 0 \Rightarrow B = T_A,$$

so $T_s(x) = T_A = 10°C$. The average initial temperature is $\frac{1}{2}(T_L + T_R) = \frac{1}{2}(-30°C + 50°C) = 10°C$

which is the same.

(b) See Mathematica notebook.

(c) See Mathematica notebook.

(2) We start with the hint: $T(x,t) = \sum_{n=1}^{\infty} C_n(t) \sin\left(\frac{n\pi x}{L}\right)$.

We substitute this into the equation. Because T satisfies the same BC's as the sine functions, we may differentiate the series termwise. We get

$$\sum_{n=1}^{\infty} \frac{dC_n}{dt} \sin\left(\frac{n\pi x}{L}\right) = -\sum_{n=1}^{\infty} D C_n \frac{n^2\pi^2}{L^2} \sin\left(\frac{n\pi x}{L}\right) + \gamma(x,t).$$

If we expand γ in a sine series, we will be able to balance coefficients in the equation. We have

$$\gamma(x,t) = \sum_{n=1}^{\infty} \gamma_n(t) \sin\left(\frac{n\pi x}{L}\right)$$

where $\gamma_n = \frac{2}{L}\int_0^L \gamma \sin\left(\frac{n\pi x}{L}\right) dx = \frac{2T_0 e^{-\alpha t}}{L^2}\int_0^L x \sin\left(\frac{n\pi x}{L}\right)$

(2) (continued) so $\gamma_n = T_0 \alpha e^{-\alpha t} \frac{2(-1)^{n+1}}{n\pi}$. Then

$$\sum \frac{dC_n}{dt} \sin\left(\frac{n\pi x}{L}\right) = \sum (-D) \frac{n^2\pi^2}{L^2} C_n \sin\left(\frac{n\pi x}{L}\right) + \sum \left(T_0 \alpha e^{-\alpha t}\right) \frac{2(-1)^{n+1}}{n\pi} \sin\left(\frac{n\pi x}{L}\right).$$

We balance the coefficients of $\sin(n\pi x/L)$ to get

$$\frac{dC_n}{dt} = -D \frac{n^2\pi^2}{L^2} C_n + T_0 \alpha e^{-\alpha t} \frac{2(-1)^{n+1}}{n\pi},$$

a set of first-order, inhomogeneous ODE's for the coefficients C_n. Because $T(x,0)=0$, we have the initial conditions

$$C_n(0) = 0.$$

The equation is solved by superposing a particular solution C_{np} and the solution C_{nH} of the homogeneous equation. We have

$$C_{nH}(t) = A_n e^{-\frac{n^2\pi^2}{L^2} D t}, \quad C_{np}(t) = \frac{2T_0 \alpha (-1)^{n+1}}{n\pi} \frac{e^{-\alpha t}}{D\frac{n^2\pi^2}{L^2} - \alpha}.$$

(We get C_{np} by trying $C_{np} = \text{const} \cdot e^{-\alpha t}$.)

The initial condition requires that $C_{nH}(0) + C_{np}(0) = 0$, so

$$A_n = -\frac{2T_0 \alpha (-1)^{n+1}}{n\pi} \frac{1}{D\frac{n^2\pi^2}{L^2} - \alpha}$$

Then $$T(x,t) = \sum_{n=1}^{\infty} \frac{2T_0 \alpha (-1)^{n+1}}{n\pi} \frac{e^{-\alpha t} - e^{-\frac{n^2\pi^2}{L^2} D t}}{\frac{n^2\pi^2}{L^2} D - \alpha} \sin\left(\frac{n\pi x}{L}\right)$$

(3) An appropriate set of eigenfunctions is

$$\sin\left(\frac{l\pi x}{a}\right) \sin\left(\frac{m\pi y}{b}\right) \sin\left(\frac{n\pi z}{c}\right).$$

(3) (continued) These are eigenfunctions of the spatial operator and they satisfy the zero BC on the boundary. We try

$$N(x,y,z,t) = \sum_{\ell,m,n=1}^{\infty} C_{\ell mn}(t)\sin\left(\frac{\ell\pi x}{a}\right)\sin\left(\frac{m\pi y}{b}\right)\cdot$$
$$\cdot\sin\left(\frac{n\pi z}{c}\right)$$

We substitute this into the equation to get

$$\sum\left\{\frac{dC_{\ell mn}}{dt} + \left[D_x\frac{\ell^2\pi^2}{a^2} + D_y\frac{m^2\pi^2}{b^2} + D_z\frac{n^2\pi^2}{c^2}\right]\cdot C_{\ell mn} + (\beta-\alpha)C_{\ell mn}\right\}\sin\left(\frac{\ell\pi x}{a}\right)\sin\left(\frac{m\pi y}{b}\right)\sin\left(\frac{n\pi z}{c}\right)$$
$$= 0$$

We balance coefficients to get

$$\frac{dC_{\ell mn}}{dt} + \lambda_{\ell mn}C_{\ell mn} = 0,$$

where $\lambda_{\ell mn} = \pi^2\left\{D_x\frac{\ell^2}{a^2} + D_y\frac{m^2}{b^2} + D_z\frac{n^2}{c^2}\right\} - (\alpha-\beta)$.

If ~~all $\lambda_{\ell mn}$~~

The solution is $C_{\ell mn}(t) = C_{\ell mn}(0)\,e^{-\lambda_{\ell mn}t}$.
If all $\lambda_{\ell mn} > 0$, the distribution decays and the reactor is subcritical. If any $\lambda_{\ell mn} < 0$, the reactor is supercritical and the neutron density will grow exponentially. If the smallest $\lambda_{\ell mn} = 0$ and all others are positive, there is a sustained steady distribution after the transients have died away. The smallest λ is associated with $\ell = m = n = 1$, so the criticality condition is

$$\frac{D_x}{a^2} + \frac{D_y}{b^2} + \frac{D_z}{c^2} = \frac{\alpha-\beta}{\pi^2}.$$

(3) (continued) We want to minimize $V = abc$ subject to the constraint that $F=0$, where

$$F = \frac{D_x}{a^2} + \frac{D_y}{b^2} + \frac{D_z}{c^2} - \frac{\alpha-\beta}{\pi^2}.$$

We use a Lagrange multiplier to do this. We form

$$G = V + \lambda F$$

and minimize G wrt a and b. We have

$$\frac{\partial G}{\partial a} = \frac{\partial V}{\partial a} + \lambda \frac{\partial F}{\partial a} = 0$$

so

$$bc - \frac{2\lambda D_x}{a^3} = 0. \qquad (1)$$

Also

$$\frac{\partial G}{\partial b} = \frac{\partial V}{\partial b} + \lambda \frac{\partial F}{\partial b} = 0,$$

so

$$ac - \frac{2\lambda D_y}{b^3} = 0. \qquad (2)$$

Multiply (1) by a and (2) by b to get

$$V = \frac{2\lambda D_x}{a^2} = \frac{2\lambda D_y}{b^2}$$

or $a = \sqrt{\sigma D_x}$, $b = \sqrt{\sigma D_y}$ where $\sigma = \frac{2\lambda}{V}$.

The constraint equation is

$$\frac{D_x}{a^2} + \frac{D_y}{b^2} + \frac{D_z}{c^2} = \frac{\alpha-\beta}{\pi^2}$$

or $\frac{2}{\sigma} + \frac{D_z}{c^2} = \frac{\alpha-\beta}{\pi^2}$ so $\sigma = \dfrac{2}{\frac{\alpha-\beta}{\pi^2} - \frac{D_z}{c^2}}$.

Then $a = \sqrt{\sigma D_x} = \sqrt{\dfrac{2D_x}{\frac{\alpha-\beta}{\pi^2} - \frac{D_z}{c^2}}}$, $b = \sqrt{\sigma D_y} = \sqrt{\dfrac{2D_y}{\frac{\alpha-\beta}{\pi^2} - \frac{D_z}{c^2}}}$

(3) (continued) The volume is

$$V = abc = \frac{2\sqrt{D_x D_y}\; c}{\frac{\alpha-\beta}{\pi^2} - \frac{D_z}{c^2}}$$

(Note that we must have $c > \pi\sqrt{\frac{D_z}{\alpha-\beta}}$ to have a critical reactor.)

ME 201/MTH 281
Newton's Law of Cooling
Assignment #7 Solutions
Oct. 30, 2008

1. INTRODUCTION

This notebook uses *Mathematica* to solve Problem 1 of Assignment #7. The notebook is a modified version of the one handed out in class. The problem is one of transient heat conduction in a slab of finite width L. The boundary condition on the left face of the slab is zero heat flux, and the condition on the right face of the slab is Newton's law of cooling, with an ambient temperature T_a. The initial temperature is the function $T_0(x)$, which in this case is a linear function of x. The relevant material properties are the thermal conductivity k, the thermal diffusivity D_f, and the heat transfer coefficient h. The specific expressions for all functions and parameters for this problem are given in section 2. The mathematical formulation of the problem is given below.

$$\begin{aligned} &\frac{\partial T}{\partial t} = D_f \frac{\partial^2 T}{\partial x^2}, \; 0 < x < L, \; t > 0, \\ &\text{with } \frac{\partial T}{\partial x}(0, t) = 0, \; k \frac{\partial T}{\partial x}(L, t) + h[T(L, t) - Ta] = 0, \\ &\text{and } T(x, 0) = T_0(x). \end{aligned} \tag{1}$$

2. PARAMETER VALUES

In this section, we define the parameter values and the initial function. This is the only place in the notebook where these quantities are given specifically. This makes it possible to change a value for the entire calculation by just changing it here. The material properties are appropriate for granite. The initial temperature is taken to be a constant.

```
h = 22.4; (** W/m^2·K **)

L = 0.5; (** m **)

k = 2.80; (** W/m·K **)

Df = 1.37 * 10^-6; (** m^2/s **)

TA = 10.0; (** °C **)

T0[x_] = TL + (TR - TL) (x / L); (** °C **)

TL = -30.0; (** °C **)
```

```
TR = 50.0; (** °C **)
```

3. STEADY STATE SOLUTION

As in all other problems of this sort, we must first find the steady-state solution $T_s(x)$, and then reformulate the problem in terms of the transient solution T_r. The steady-state solution is obtained by setting the time derivative to zero in the original equation for T. The solution of the resulting equation is a linear function of x, and we require that particular linear function which satisfies the boundary conditions at $x = 0$ and L. The result (which is intuitively obvious) is that $T_s(x)$ is constant and equal to the ambient temperature:

```
Ts[x_] := TA
```

4. FORMULATION OF PROBLEM FOR TRANSIENT

Now that we have the steady-state solution, we decompose the full solution into a steady-state part $T_s(x)$ and a transient part $T_r(x,t)$:

$$T(x,t) = T_s(x) + T_r(x,t) \quad . \tag{2}$$

By substituting this decomposition into the original equation for T, we find the following problem for the transient:

$$\frac{\partial T_r}{\partial t} = D_f \frac{\partial^2 T_r}{\partial x^2}, \; 0 < x < L, \; t > 0,$$

$$\text{with } \frac{\partial T_r}{\partial x}(0, t) = 0, \; k \frac{\partial T_r}{\partial x}(L, t) + h\, T_r(L, t) = 0, \tag{3}$$

$$\text{and } T_r(x, 0) = T_0(x) - T_s(x) \, .$$

This is the problem that we solve by separation of variables.

5. SEPARATION OF VARIABLES IN TRANSIENT PROBLEM

Following the standard approach in separation of variables, we seek functions which satisfy the equation for T_r and the homogeneous boundary conditions at $x = 0, L$. We assume a form $F(x)G(t)$ for such solutions. The two separated equations then have the form

$$F''(x) + \lambda F(x) = 0, \text{ with } F'(0) = 0 \text{ and } kF'(L) + hF(L) = 0,$$

$$\text{and } G'(t) + \lambda D G(t) = 0 \, . \tag{4}$$

The general solution for the equation for $F(x)$ is a linear combination of $\sin(\sqrt{\lambda}\, x)$ and $\cos(\sqrt{\lambda}\, x)$. Applying the two homogeneous boundary conditions to the general solution, and simplifying the resulting eigenvalue equation, we get for the eigenvalues

```
lam[n_] := (z / L) ^2 /. z -> z[n]
```

where the values z[1], z[2], .. z[n],.. are the roots of eig1 = eig2, where

```
eig1 = Tan[z];
```

```
eig2 = Bi / z;
```

with

```
Bi := (h * L) / k
```

The generic eigenfunction is

```
geneig[x_] := Cos[(z * x) / L]
```

and the nth eigenfunction is given by

```
Feig[x_, n_] := geneig[x] /. z -> z[n]
```

The generic normalization integral is

```
nor = Integrate[(geneig[x])^2, {x, 0, L}]
```

```
       0.125 Sin[2. z]
0.25 + ───────────────
              z
```

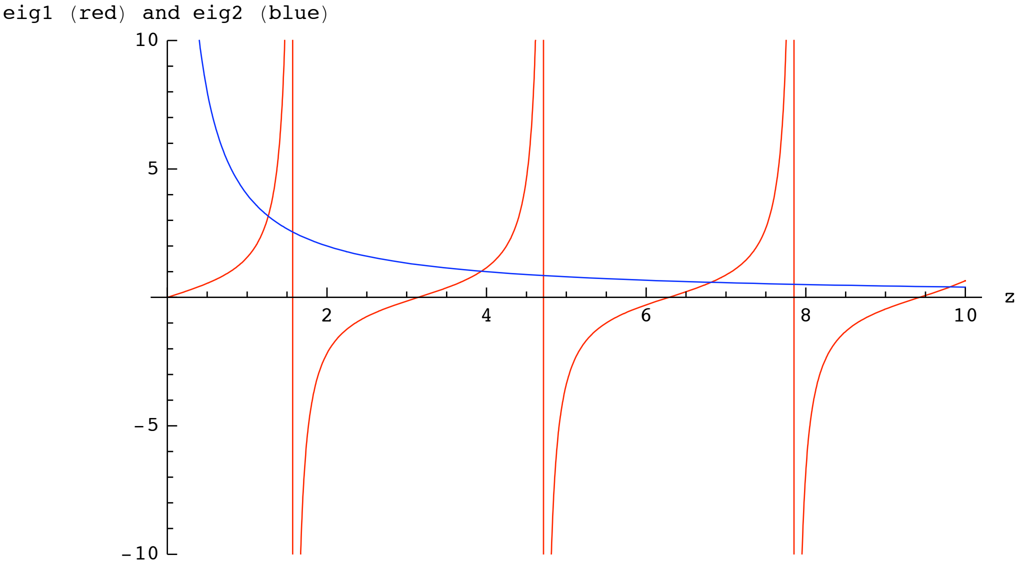
and the nth normalization integral is

```
norm[n_] := nor /. z -> z[n]
```

6. DETERMINATION OF EIGENVALUES

We now determine numerically the eigenvalues. We begin by plotting the expressions eig1 and eig2. At each crossing of these, there is an eigenvalue.

```
Plot[{eig1, eig2}, {z, 0, 10}, PlotRange → {-10, 10},
 PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]},
 ImageSize → 380, AxesLabel → {"z", "eig1 (red) and eig2 (blue)"}]
```



We see that the first root is between 0 and $\pi/2$, and each subsequent root is in an interval of length π. It is also clear that for large n, the nth root is very close to (n-1)π. We now use a Do loop to calculate and display in a table the first 100 roots. This will suffice for any calculation requiring up to 100 terms in the eigenfunction series. We also display the eigenvalue λ, and the approximate value z = (n-1)π.

```
zroot[n_] := FindRoot[eig1 == eig2, {z, (n - 0.5) * Pi - 0.1}]

zasymp[n_] := N[(n - 1) * Pi]

Do[z[n] = z /. zroot[n], {n, 1, 100}];

TableForm[
 Table[{n, PaddedForm[z[n], {10, 4}], PaddedForm[zasymp[n], {10, 4}],
   PaddedForm[lam[n], {10, 4}]}, {n, 1, 100}], TableHeadings ->
  {None, {"n", "         z[n]", "     zasymp[n]", "       lam[n]"}}]
```

```
n           z[n]     zasymp[n]         lam[n]
1         1.2646        0.0000         6.3968
2         3.9352        3.1416        61.9420
3         6.8140        6.2832       185.7229
4         9.8119        9.4248       385.0918
5        12.8678       12.5664       662.3165
6        15.9536       15.7080      1018.0727
7        19.0565       18.8496      1452.5939
8        22.1697       21.9911      1965.9744
9        25.2896       25.1327      2558.2574
10       28.4142       28.2743      3229.4647
11       31.5421       31.4159      3979.6082
12       34.6724       34.5575      4808.6949
13       37.8045       37.6991      5716.7291
14       40.9381       40.8407      6703.7134
15       44.0728       43.9823      7769.6496
16       47.2084       47.1239      8914.5391
17       50.3448       50.2655     10138.3827
18       53.4817       53.4071     11441.1809
19       56.6192       56.5487     12822.9343
20       59.7571       59.6903     14283.6432
21       62.8954       62.8319     15823.3079
22       66.0339       65.9734     17441.9285
23       69.1728       69.1150     19139.5052
24       72.3119       72.2566     20916.0381
25       75.4512       75.3982     22771.5274
26       78.5907       78.5398     24705.9731
27       81.7303       81.6814     26719.3752
28       84.8701       84.8230     28811.7339
29       88.0100       87.9646     30983.0491
30       91.1500       91.1062     33233.3210
31       94.2902       94.2478     35562.5495
32       97.4304       97.3894     37970.7346
33      100.5707      100.5310     40457.8764
34      103.7111      103.6726     43023.9750
```

```
35     106.8516    106.8142     45669.0302
36     109.9921    109.9557     48393.0422
37     113.1327    113.0973     51196.0109
38     116.2733    116.2389     54077.9364
39     119.4140    119.3805     57038.8186
40     122.5547    122.5221     60078.6576
41     125.6955    125.6637     63197.4533
42     128.8363    128.8053     66395.2059
43     131.9772    131.9469     69671.9152
44     135.1181    135.0885     73027.5813
45     138.2590    138.2301     76462.2042
46     141.4000    141.3717     79975.7839
47     144.5409    144.5133     83568.3204
48     147.6819    147.6549     87239.8137
49     150.8230    150.7964     90990.2638
50     153.9640    153.9380     94819.6708
51     157.1051    157.0796     98728.0345
52     160.2462    160.2212    102715.3551
53     163.3873    163.3628    106781.6324
54     166.5284    166.5044    110926.8666
55     169.6696    169.6460    115151.0576
56     172.8107    172.7876    119454.2054
57     175.9519    175.9292    123836.3100
58     179.0931    179.0708    128297.3715
59     182.2343    182.2124    132837.3898
60     185.3755    185.3540    137456.3649
61     188.5168    188.4956    142154.2968
62     191.6580    191.6372    146931.1855
63     194.7993    194.7787    151787.0311
64     197.9405    197.9203    156721.8335
65     201.0818    201.0619    161735.5927
66     204.2231    204.2035    166828.3088
67     207.3644    207.3451    171999.9816
68     210.5057    210.4867    177250.6113
69     213.6470    213.6283    182580.1979
70     216.7883    216.7699    187988.7412
71     219.9297    219.9115    193476.2414
72     223.0710    223.0531    199042.6984
73     226.2124    226.1947    204688.1123
74     229.3537    229.3363    210412.4830
75     232.4951    232.4779    216215.8105
76     235.6364    235.6194    222098.0948
77     238.7778    238.7610    228059.3360
78     241.9192    241.9026    234099.5340
79     245.0605    245.0442    240218.6888
80     248.2019    248.1858    246416.8005
81     251.3433    251.3274    252693.8690
82     254.4847    254.4690    259049.8943
83     257.6261    257.6106    265484.8764
84     260.7675    260.7522    271998.8154
```

```
85       263.9089      263.8938    278591.7112
86       267.0504      267.0354    285263.5639
87       270.1918      270.1770    292014.3734
88       273.3332      273.3186    298844.1397
89       276.4746      276.4602    305752.8629
90       279.6161      279.6017    312740.5428
91       282.7575      282.7433    319807.1797
92       285.8989      285.8849    326952.7733
93       289.0404      289.0265    334177.3238
94       292.1818      292.1681    341480.8311
95       295.3233      295.3097    348863.2953
96       298.4647      298.4513    356324.7162
97       301.6062      301.5929    363865.0941
98       304.7476      304.7345    371484.4287
99       307.8891      307.8761    379182.7202
100      311.0305      311.0177    386959.9685
```

We see that the asymptotic formula for z[n] has an error of less than 0.5% for any n equal to or greater than 10. It is possible to develop even more accurate, but still simple, asymptotic formulas.

■ 7. REPRESENTATION OF THE INITIAL CONDITION

We are finally ready to begin solving the boundary value for the transient $T_r(x,t)$. The form of the solution for the transient is

$$T_r(x,t) = \sum_{n=1}^{\infty} C(n)\mathrm{Exp}(-\lambda(n) D_f t)\cos[(z(n)x/L], \qquad (5)$$

where the coefficients $C(n)$ are determined by the initial conditions satisfied by T_r. The generic formula for the *n*th coefficient is

```
coeff[n_] := Integrate[Feig[x, n] * (T0[x] - Ts[x]), {x, 0, L}] / norm[n]
```

We now use a Do loop to construct the first 100 coefficients, and then print them out in a table. The *n*th numerical value is assigned to c[n].

```
Do[c[n] = coeff[n], {n, 1, 100}];

TableForm[Table[{n, PaddedForm[c[n], {8, 4}]}, {n, 1, 100}],
 TableHeadings -> {None, {"n", "      c[n]"}}]
```

```
n         c[n]
--------------
1      -7.7995
2     -28.4554
3       5.1400
4      -6.0628
5       1.7631
6      -2.4221
7       0.8441
8      -1.2766
9       0.4881
10     -0.7831
11      0.3166
```

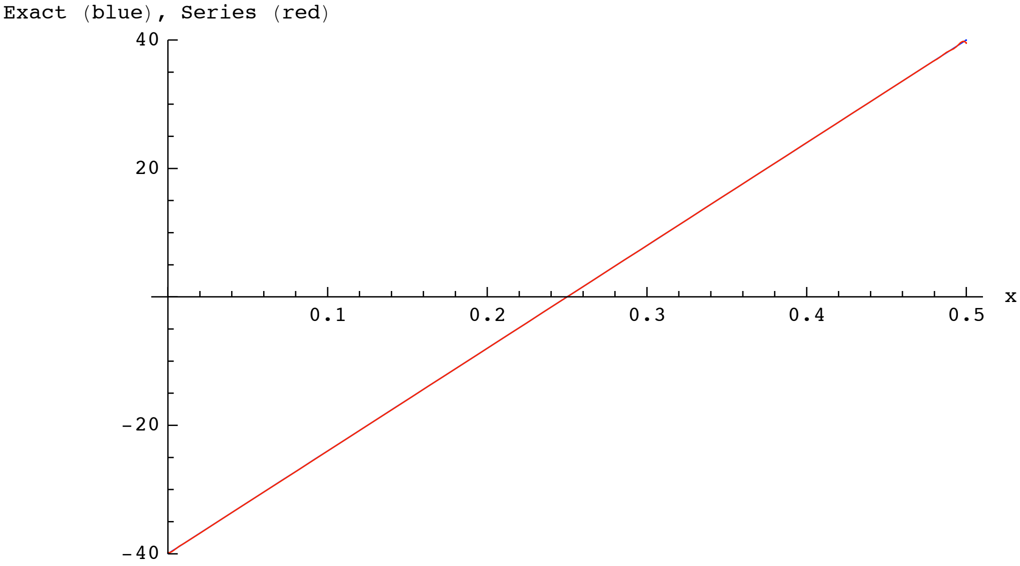
```
12   -0.5280
13    0.2214
14   -0.3796
15    0.1634
16   -0.2859
17    0.1255
18   -0.2230
19    0.0993
20   -0.1787
21    0.0806
22   -0.1464
23    0.0667
24   -0.1222
25    0.0561
26   -0.1035
27    0.0478
28   -0.0887
29    0.0412
30   -0.0769
31    0.0359
32   -0.0673
33    0.0316
34   -0.0594
35    0.0280
36   -0.0529
37    0.0250
38   -0.0473
39    0.0224
40   -0.0426
41    0.0202
42   -0.0385
43    0.0184
44   -0.0350
45    0.0167
46   -0.0320
47    0.0153
48   -0.0293
49    0.0141
50   -0.0270
51    0.0130
52   -0.0249
53    0.0120
54   -0.0231
55    0.0111
56   -0.0214
57    0.0103
58   -0.0199
59    0.0096
60   -0.0186
61    0.0090
```

```
62      -0.0174
63       0.0084
64      -0.0163
65       0.0079
66      -0.0153
67       0.0074
68      -0.0144
69       0.0070
70      -0.0136
71       0.0066
72      -0.0129
73       0.0063
74      -0.0122
75       0.0059
76      -0.0115
77       0.0056
78      -0.0109
79       0.0053
80      -0.0104
81       0.0051
82      -0.0099
83       0.0048
84      -0.0094
85       0.0046
86      -0.0090
87       0.0044
88      -0.0086
89       0.0042
90      -0.0082
91       0.0040
92      -0.0078
93       0.0038
94      -0.0075
95       0.0037
96      -0.0072
97       0.0035
98      -0.0069
99       0.0034
100     -0.0066
```

As a check on all that we have done, we see how well our series represents the initial condition, by plotting both the initial condition (in blue) and its representation by the first 100 terms of our series (in red).

```
exactinit[x_] := T0[x] - Ts[x]

seriesinit[x_] := Sum[c[n] * Feig[x, n], {n, 1, 100}]
```

```
seriesgraph =
 Plot[{exactinit[x], seriesinit[x]}, {x, 0, L}, PlotRange → {-40, 40},
  ImageSize → 380, PlotStyle → {RGBColor[0, 0, 1], RGBColor[1, 0, 0]},
  AxesLabel → {"x", "Exact (blue), Series (red)"}]
```



The agreement looks excellent, and is good evidence that our calculations are correct so far. We check the value of the temperature at the end. The exact value should be 40. The series gives

```
seriesinit[L]
```

```
39.5121
```

The error is about 2.5%. The error will of course be much less for $t > 0$ because of the exponential time factors in the series.

■ 8. SOLUTION OF THE INITIAL-BOUNDARY VALUE PROBLEM

Now that we have the eigenvalues and series coefficients, we can construct the series solution of the problem. We define Tran[x,t,n] as the nth partial sum of the series solution, and we define term[x,t,n] as the nth term in the series. As a special case of importance, we define firstterm[x,t] to be the first term in the series.

```
term[x_, t_, n_] := c[n] * Feig[x, n] * Exp[-lam[n] * Df * t]
```

```
Tran[x_, t_, n_] := Sum[term[x, t, k], {k, 1, n}]
```

```
firstterm[x_, t_] := term[x, t, 1]
```

Let's look at the first term:

```
firstterm[x, t]
```

$-7.79951\, e^{-8.76357\times10^{-6}\, t}\, \mathrm{Cos}[2.52918\, x]$

We may calculate the e-folding time for this mode -- call it $\tau1$ -- as the reciprocal of the coefficient of t in the exponent:

```
τ1 = 1 / (lam[1] * Df)
```

```
114 109.
```

This time, which is in seconds, equals about 31.7 hours. We compare this with the diffusion time $L^2 \Big/ \left(\pi^2 D_f\right)$we derived earlier for simpler boundary conditions:

$$L^2 \Big/ \left(\pi^2 \, \mathrm{Df}\right)$$

```
18 489.3
```

It is interesting that the first decay time is considerably longer than our earlier estimate of diffusion time. This happens because in the present problem there is thermal resistance at the boundary (characterized by the reciprocal of the heat transfer coefficient *h*) whereas in the earlier problem in which the temperature was specified on the boundary, there was no resistance to transfer from the surface to the surroundings. The extra resistance at the boundary in the present case gives rise to a longer decay time.

Let's look at the decay times of the second and third modes, converting them to hours as we calculate them

```
τ2 = 1 / (lam[2] * Df * 3600)
```

```
3.27335
```

```
τ3 = 1 / (lam[3] * Df * 3600)
```

```
1.09172
```

Knowledge of these decay times will help us choose time values for plotting the solution. To make this a little easier, we construct a table of the first 100 decay times.

```
τ[n_] := 1 / (lam[n] * Df)
```

```
TableForm[
 Table[{n, PaddedForm[τ[n], {10, 2}], PaddedForm[τ[n] / 3600, {10, 5}]},
  {n, 1, 100}], TableHeadings ->
  {None, {"Mode", "Decay Time (s)", "Decay Time (hr)"}}]
```

```
Mode  Decay Time (s)  Decay Time (hr)
1        114108.73        31.69687
2         11784.04         3.27335
3          3930.19         1.09172
4          1895.46         0.52652
5          1102.08         0.30613
6           716.97         0.19916
7           502.50         0.13958
8           371.28         0.10313
9           285.32         0.07926
```

```
10        226.02        0.06278
11        183.42        0.05095
12        151.79        0.04216
13        127.68        0.03547
14        108.88        0.03025
15         93.95        0.02610
16         81.88        0.02274
17         72.00        0.02000
18         63.80        0.01772
19         56.92        0.01581
20         51.10        0.01420
21         46.13        0.01281
22         41.85        0.01162
23         38.14        0.01059
24         34.90        0.00969
25         32.05        0.00890
26         29.54        0.00821
27         27.32        0.00759
28         25.33        0.00704
29         23.56        0.00654
30         21.96        0.00610
31         20.53        0.00570
32         19.22        0.00534
33         18.04        0.00501
34         16.97        0.00471
35         15.98        0.00444
36         15.08        0.00419
37         14.26        0.00396
38         13.50        0.00375
39         12.80        0.00355
40         12.15        0.00337
41         11.55        0.00321
42         10.99        0.00305
43         10.48        0.00291
44         10.00        0.00278
45          9.55        0.00265
46          9.13        0.00254
47          8.73        0.00243
48          8.37        0.00232
49          8.02        0.00223
50          7.70        0.00214
51          7.39        0.00205
52          7.11        0.00197
53          6.84        0.00190
54          6.58        0.00183
55          6.34        0.00176
56          6.11        0.00170
57          5.89        0.00164
58          5.69        0.00158
59          5.49        0.00153
```

```
60            5.31         0.00148
61            5.13         0.00143
62            4.97         0.00138
63            4.81         0.00134
64            4.66         0.00129
65            4.51         0.00125
66            4.38         0.00122
67            4.24         0.00118
68            4.12         0.00114
69            4.00         0.00111
70            3.88         0.00108
71            3.77         0.00105
72            3.67         0.00102
73            3.57         0.00099
74            3.47         0.00096
75            3.38         0.00094
76            3.29         0.00091
77            3.20         0.00089
78            3.12         0.00087
79            3.04         0.00084
80            2.96         0.00082
81            2.89         0.00080
82            2.82         0.00078
83            2.75         0.00076
84            2.68         0.00075
85            2.62         0.00073
86            2.56         0.00071
87            2.50         0.00069
88            2.44         0.00068
89            2.39         0.00066
90            2.33         0.00065
91            2.28         0.00063
92            2.23         0.00062
93            2.18         0.00061
94            2.14         0.00059
95            2.09         0.00058
96            2.05         0.00057
97            2.01         0.00056
98            1.96         0.00055
99            1.93         0.00053
100           1.89         0.00052
```

■ 9. GRAPHS OF THE SOLUTION

We define here graphs of the solution for T versus x, for various values of time. We test the code with a few graphs, and then we construct a sequence to be animated to show the evolution of the system with time. The function which produces the graph at time t is named snapshot[t]. We use the decay times to determine how many terms to keep. We agree, somewhat arbitrarily, that e^{-6} or smaller is negligible. Then we define a function nterms[t] which tells us how many terms to keep, with the understanding that we can't keep more than 100, and we will always keep at least one.

```
nterms[t_] :=
 Module[{n}, n = 1; While[((t / τ[n] < 6) && (n < 100)), n = n + 1]; n]
```

Thus, for example, for times of 20 s, 0.25 hr = 900 s and 2.0 hr = 7200 s the number of terms needed are

```
nterms[20]
```

```
76
```

```
nterms[900]
```

```
13
```

```
nterms[7200]
```

```
5
```

Now we use this to define in a practical way the sum of the series for T_r. We call this sum Trans, and the temperature we call Temp.

```
Trans[x_, t_] := Tran[x, t, nterms[t]]
```

```
Temp[x_, t_] := Trans[x, t] + Ts[x]
```

Now we are ready to define snapshot[t], which gives us a plot of temperature versus x at time t. We define the $t = 0$ case in terms of the initial condition. From the ambient temperature and the initial distribution, we can get a range of variation of temperature. We extend this range slightly on either side, and call the extended range plrange:

```
plrange = {-35, 55};
```

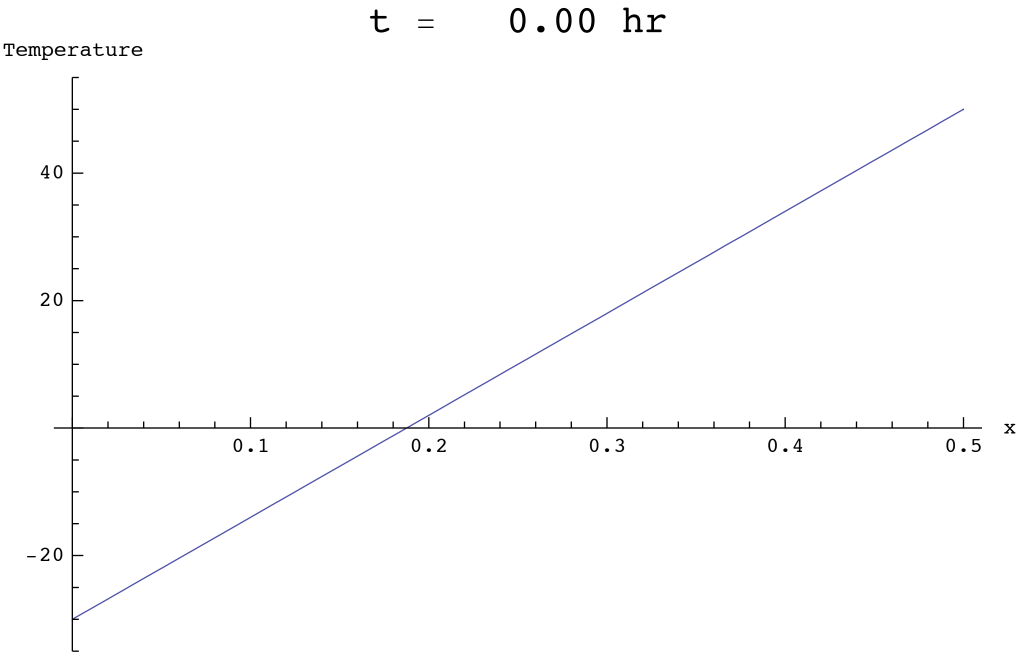
Now we define snapshot[t]. For convenience, the function snapshot takes an argument in hours, but converts it internally to seconds, to be consistent with the units of the problem.

```
snapshot[t_] := Module[{time}, time = 3600 * t;
  Plot[Temp[x, time], {x, 0, L}, PlotRange -> plrange,
   AxesLabel -> {"x", "Temperature"}, ImageSize → 380,
   PlotLabel → Row[{"t = ", PaddedForm[t, {4, 2}], " hr"}]]]
```

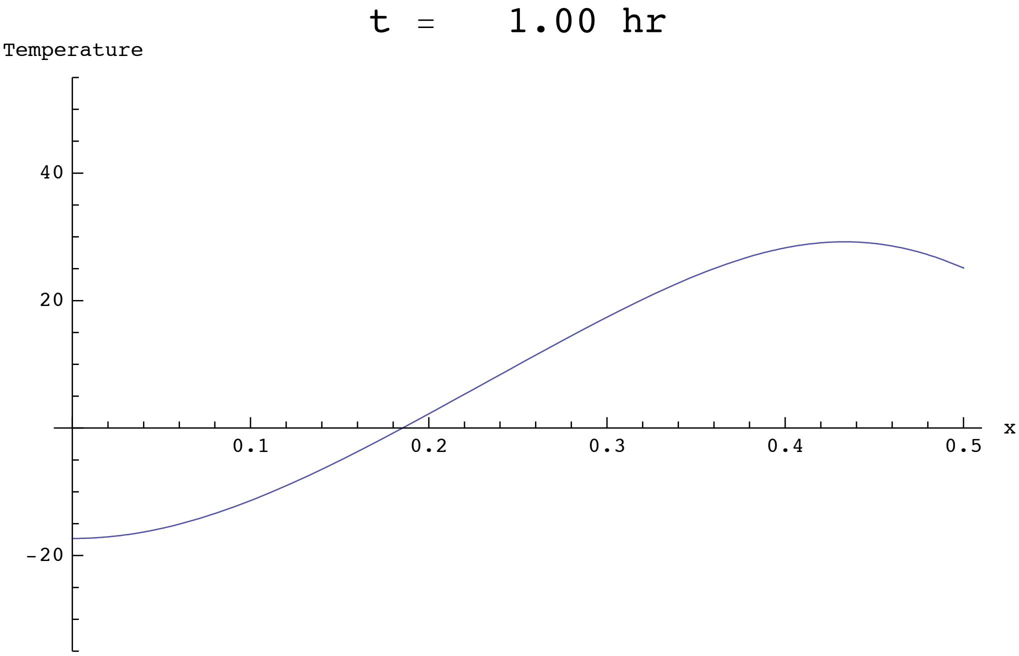
```
snapshot[0] := Module[{time}, time = 3600 * t;
  Plot[T0[x], {x, 0, L}, PlotRange -> plrange,
   AxesLabel -> {"x", "Temperature"}, ImageSize → 380,
   PlotLabel → Row[{"t = ", PaddedForm[0, {4, 2}], " hr"}]]]
```

We try it for the initial time and for 1 and 10 hr.

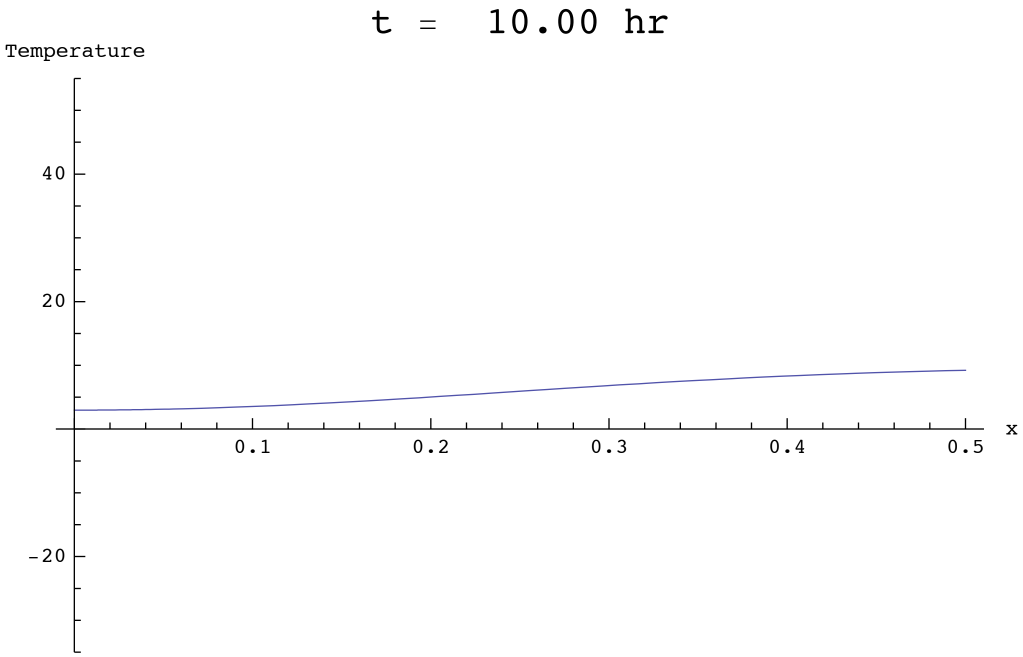
```
snapshot[0]
```



```
snapshot[1]
```

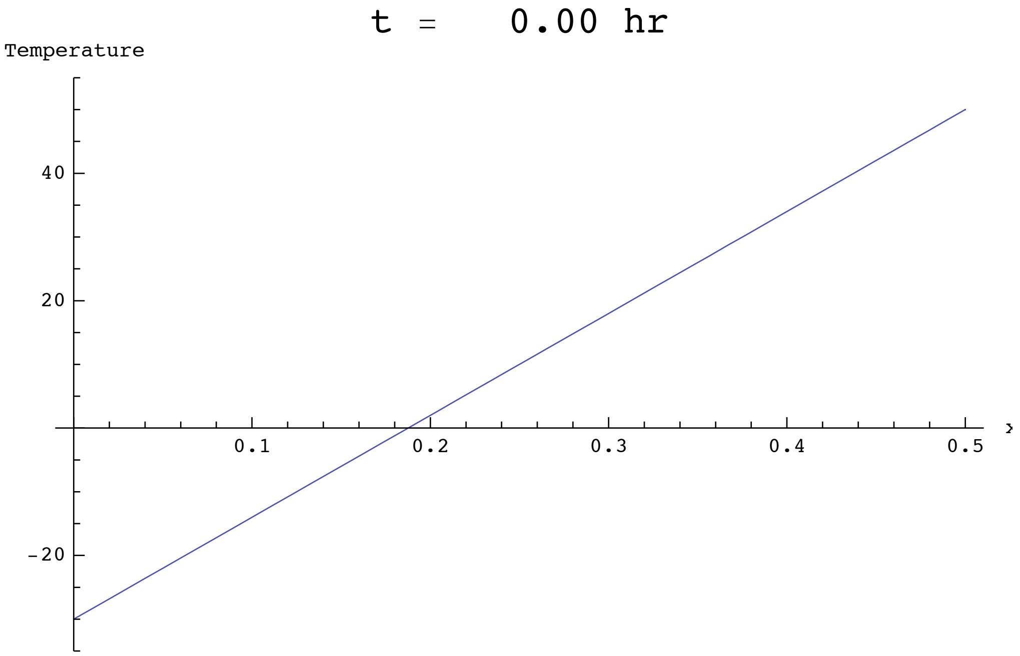


```
snapshot[10]
```



As a final task, we produce a sequence of graphs that can be animated to show the dynamics of the heat loss process. We produce a sequence of 101 graphs at 0.5 hr intervals, running from 0 hr to 50 hr. In the printed version of the notebook, only one graph in the sequence is shown.

```
Do[Print[snapshot[0.5 * i]], {i, 0, 100}]
```



■ 10. HEAT FLUX

We denote the heat flux out of the boundary at $x = L$ by $F(t)$. It is given by

$$F(t) = -k\, \partial T / \partial x \,|_{x=L}\, .$$

By using the boundary condition at $x = L$, we may write this as

$$F(t) = h\, [T(L, t)\, -\, T_A].$$

We define this for *Mathematica*.

```
F[t_] := h (Temp[L, t] - TA)
```

The exact value at $t = 0$ is

```
h (T0[L] - TA)
```

```
896.
```

We compare this with the value from the series:

```
F[0]
```

```
885.071
```

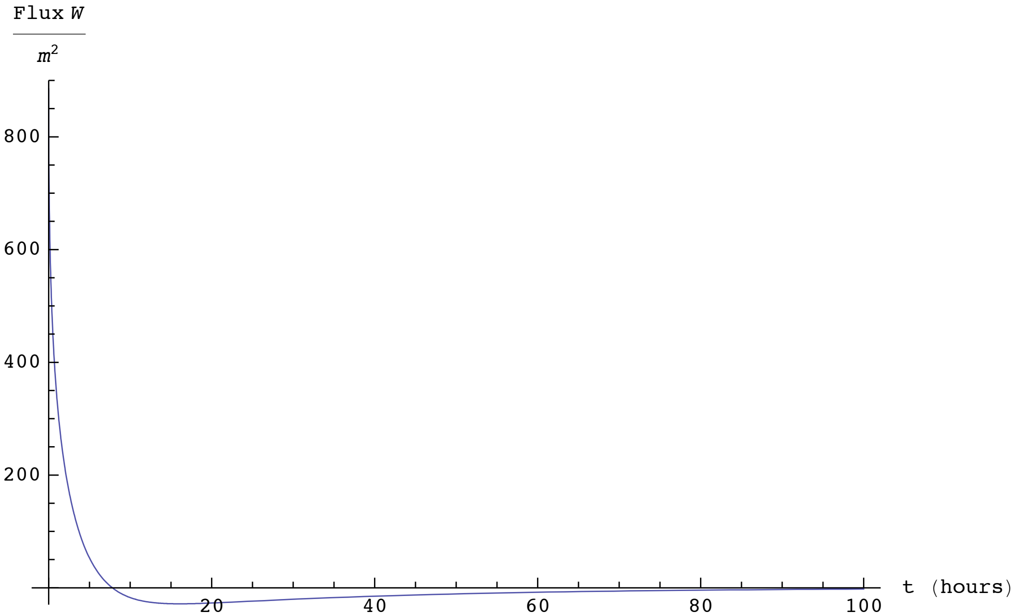
We see an error of about 1.2 % at the initial time. A few seconds after the initial time, the error will be much smaller because of the exponential time factors in the series.

For some purposes it is convenient to have the time argument in hours. We define the function with that time scale as Fh[t].

```
Fh[t_] := F[t * 3600.]
```

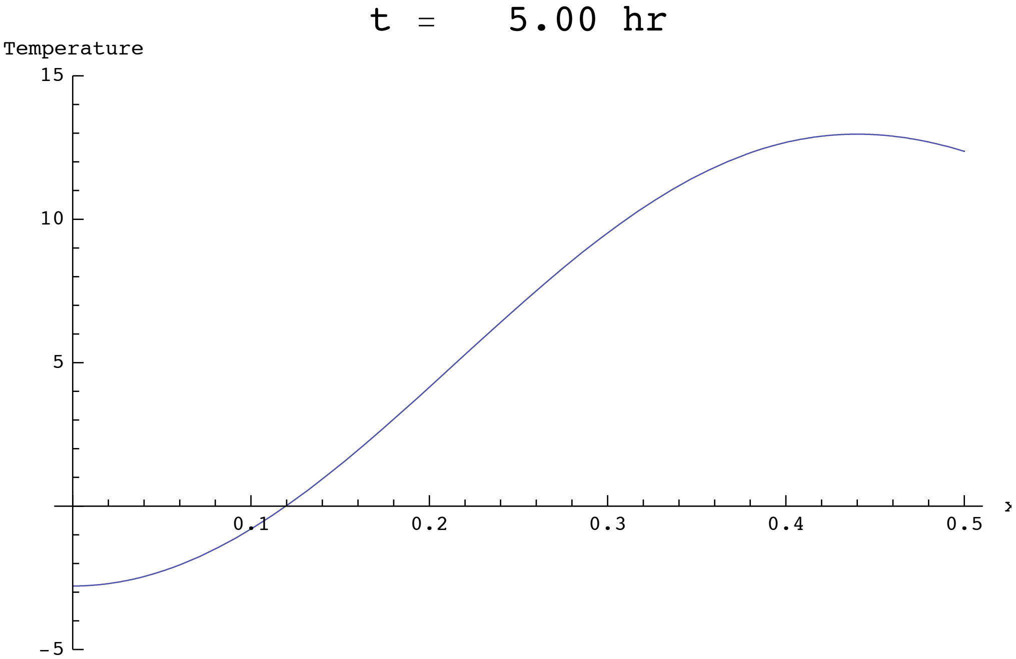
Now we plot the flux from the initial time to 100 hours.

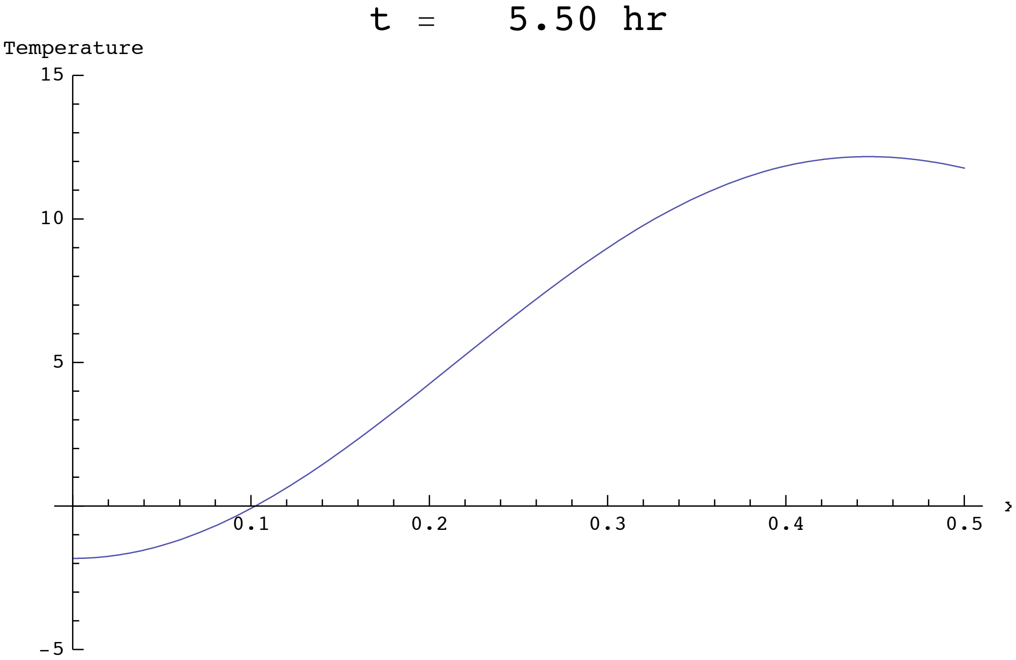
```
Plot[Fh[t], {t, 0, 100}, PlotRange → {-30, 900},
 ImageSize → 380, AxesLabel → {"t (hours)", Flux W/m^2}]
```

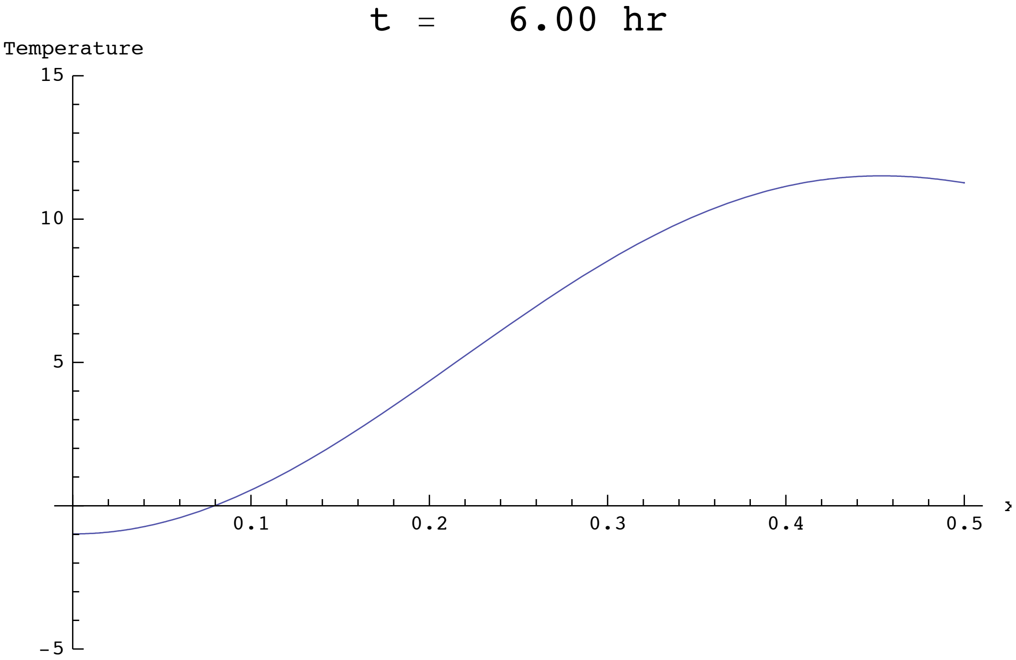


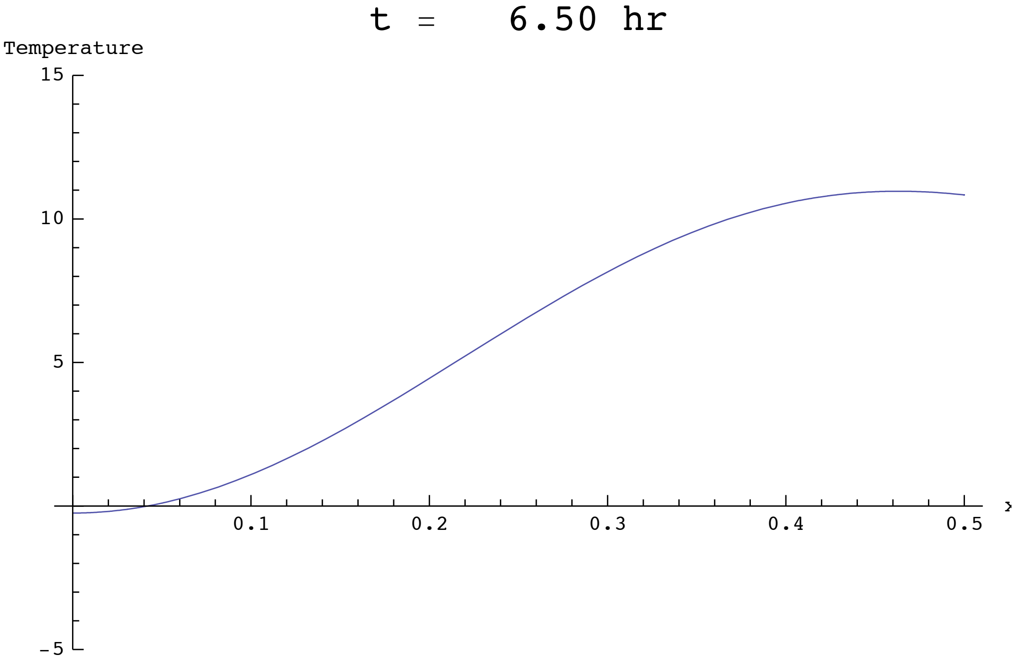
We see that the flux is positive (out of the slab) for the early times, and negative (into the slab) for the later times. The cross-over occurs between 5 and 10 hours. We look at the temperature profiles in that time range. We change the plot range to get a better view.

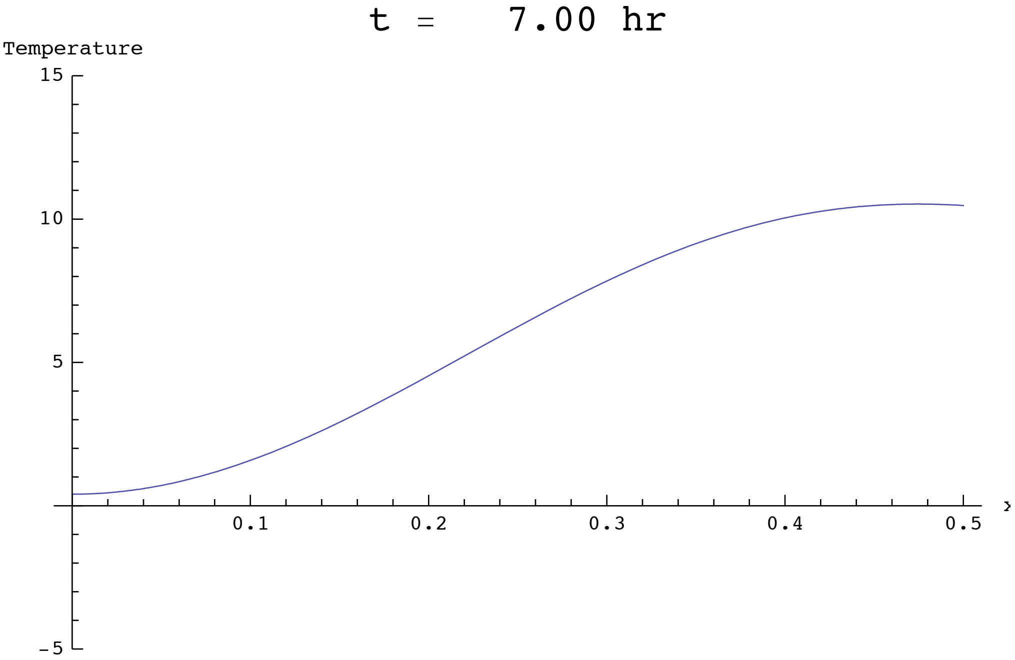
```
plrange = {-5, 15};

Do[Print[snapshot[5 + 0.5 * i]], {i, 0, 10}]
```

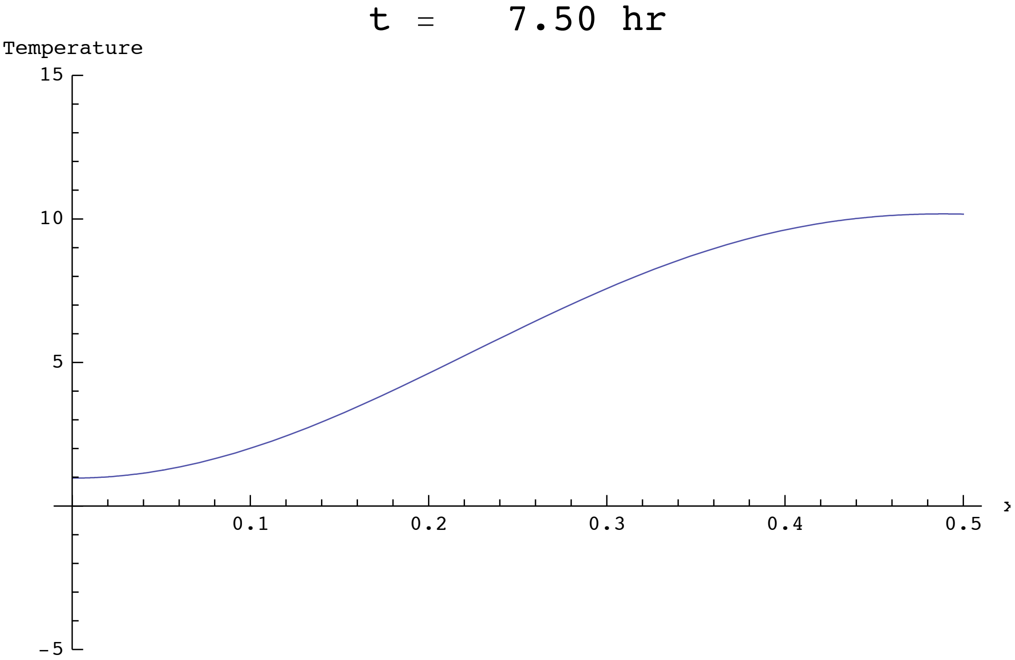

t = 5.00 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x

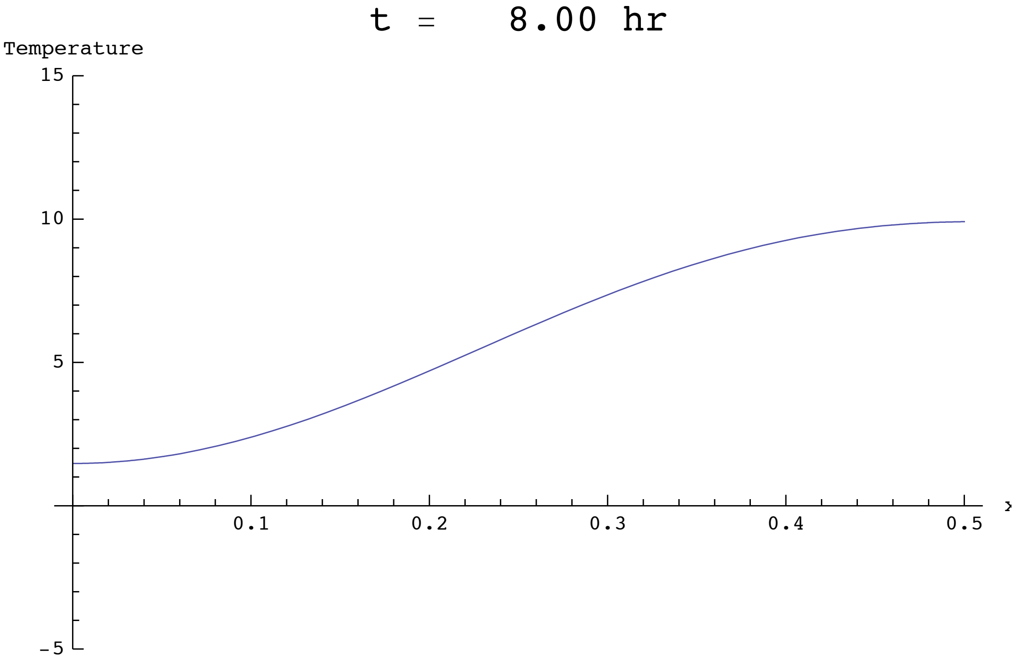

t = 5.50 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x

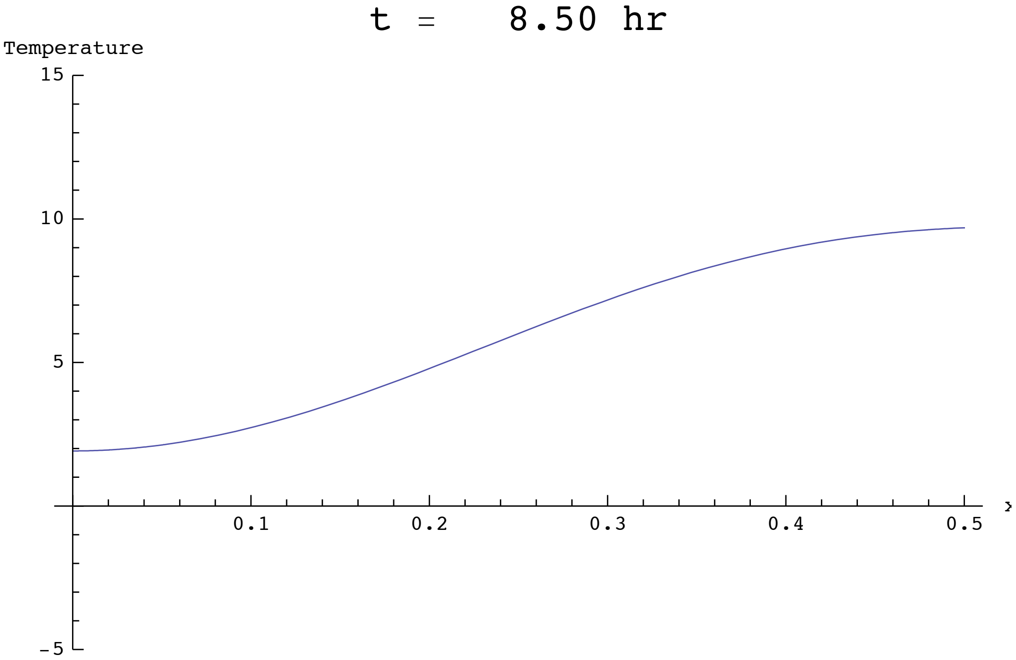

t = 6.00 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x

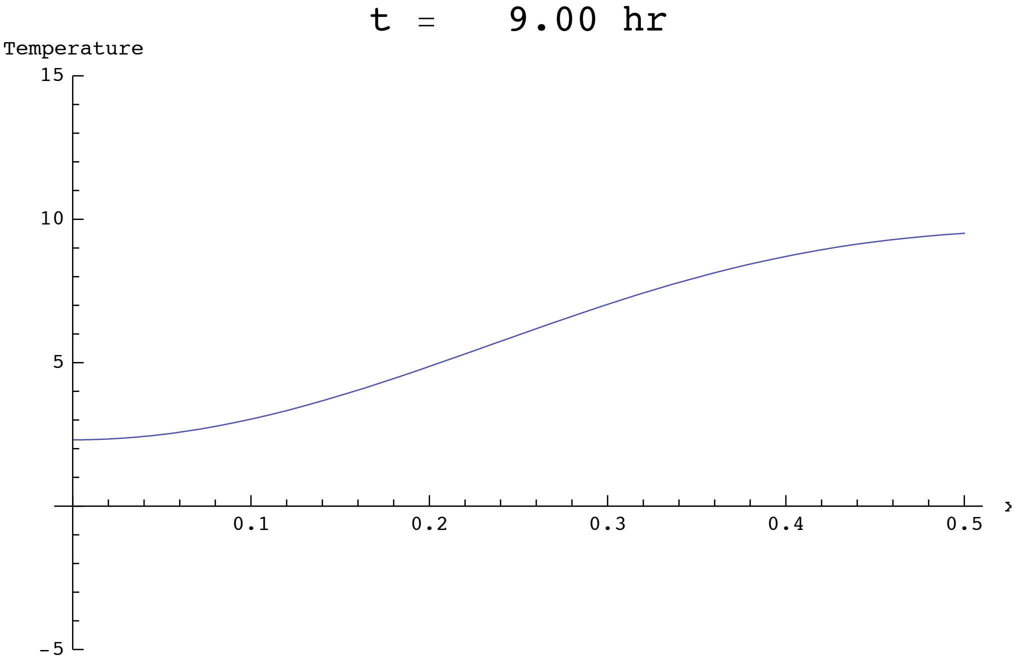

t = 6.50 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x

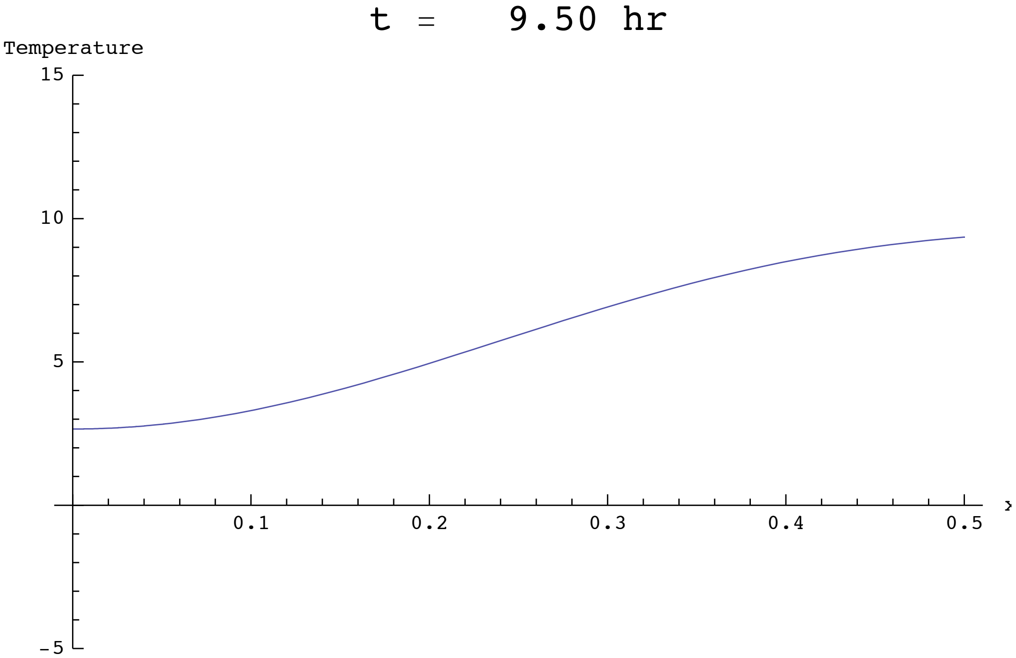

t = 7.00 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x

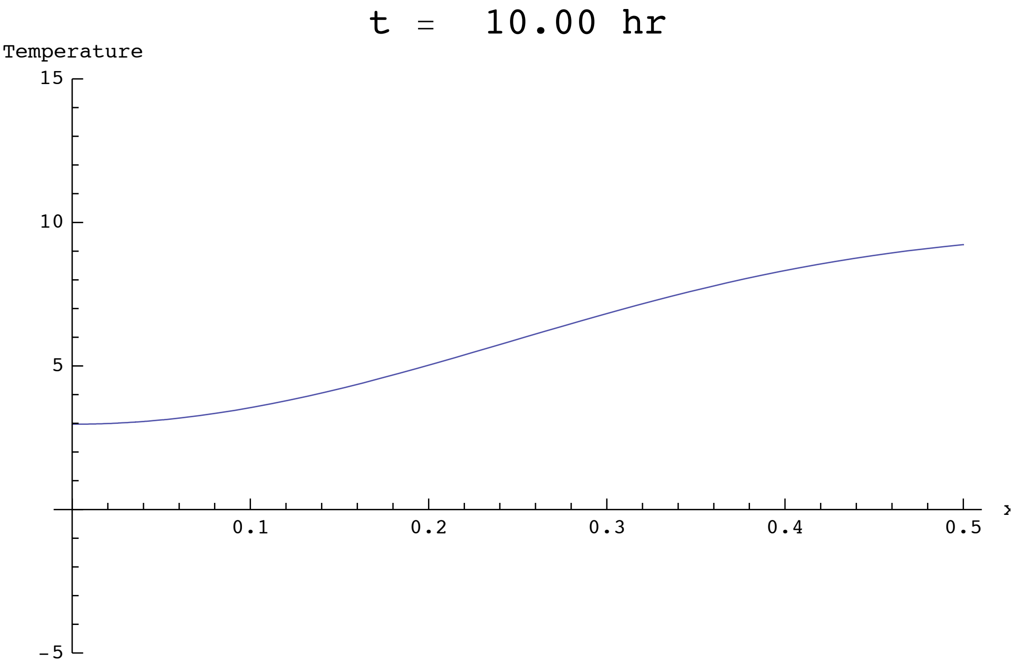

t = 7.50 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x


t = 8.00 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x


t = 8.50 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x


t = 9.00 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x


t = 9.50 hr
Temperature
15
10
5
-5
0.1
0.2
0.3
0.4
0.5
x



The graphs suggest that the flux changes sign at around 7.5 hours. We pin this down a little further by trying a few values.

```
Fh[7.5]
```

```
3.74681
```

```
Fh[7.75]
```

```
0.72783
```

```
Fh[8.0]
```

```
-2.04495
```

```
Fh[7.9]
```

```
-0.964142
```

```
Fh[7.8]
```

```
0.154203
```

```
Fh[7.84]
```

```
-0.297716
```

```
Fh[7.81]
```

```
0.0406451
```

```
Fh[7.82]
```

```
-0.0725266
```

The flux reverses sign at about 7.81 hours.

The average initial temperature is

$$\frac{1}{2}(TL + TR)$$

```
10.
```

The final steady state temperature is

```
TA
```

```
10.
```

These are the same, which means the final thermal energy is the same as the initial thermal energy. Thus the flux integrated over all times should give us zero. (Remember that the left boundary is insulated, so that no energy enters or leaves from it.)