Dubin 4.2.2 Broiling Steak 11-13-16

N. T. Gladd

Initialization: Be sure the files *NTGStylesheet2.nb* and *NTGUtilityFunctions.m* is are in the same directory as that from which this notebook was loaded. Then execute the cell immediately below by mousing left on the cell bar to the right of that cell and then typing “shift” + “enter”. Respond “Yes” in response to the query to evaluate initialization cells.

In[8]:=
```
SetDirectory[NotebookDirectory[]];
(* set directory where source files are located *)
SetOptions[EvaluationNotebook[], (* load the StyleSheet *)
 StyleDefinitions → Get["NTGStylesheet2.nb"]];
Get["NTGUtilityFunctions.m"]; (* Load utilities package *)
```

Previous version *Dubin 4.2.2 Broiling Steak 10-25-16*

Purpose

I solve a heat equation problem from Chapter 4 of *Numerical and Analytical Methods for Scientists and Engineers, Using Mathematica*, Daniel Dubin. The specific example (p 294) considers a constant flux of energy into a 1-D slab from both sides — von Neumann boundary conditions. As posed the problem has to be transformed before it is amenable to solution by by separation of variables.

(2) When one cooks using radiant heat (under a broiler, for example), there is a heat flux Γ_r due to the radiation, incident on the surface of the food. On the other hand, the food is typically suspended (on a grill or spit for example) in such a way that it cannot conduct heat very well to the environment, so that little heat is re-radiated. Under these conditions, find the time required to raise the internal temperature of a slab of meat of thickness $L = 5\,\text{cm}$ from $T = 20\,°\text{C}$ to $T = 90\,°\text{C}$. Animate the solution for $T(x, t)$ up to this time. Take $\chi = 3\times10^{-7}\,\text{m}^2/\text{s}$, $C = 3\times10^{6}\,\text{J}/(\text{m}^3\,\text{K})$, and assume that both faces of the meat are subjected to the same flux of heat, equal to $10\,\text{kW}/\text{m}^2$(a typical value in an oven).

1 Problem setup

I construct an Association that encapsulates information about this problem, and then apply the function *DSolveHeatEquation* that attempts to solve this problem directly using the Mathematica function *DSolve*.

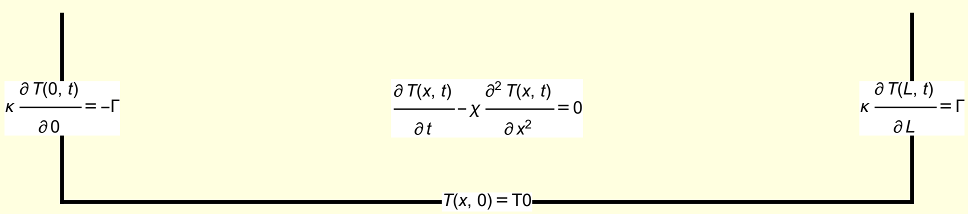
In[10]:=

```
A1 =
  Module[{description, pde, bcL, bcR, ic, eqns, depVar,
    assumptions, substitutions, simplifications, names, values},
   description = "Dubin problem 4.2.2 Broiling Steak\nHomogeneous
      heat equation, inhomogeneous Neumann boundary conditions";
   pde = D[T[x, t], t] - χ D[T[x, t], {x, 2}] == 0;
   bcL = κ Derivative[1, 0][T][0, t] == -Γ; (* inhomogeneous von Neumann *)
   bcR = κ Derivative[1, 0][T][L, t] == Γ; (* inhomogeneous von Neumann *)
   ic = T[x, 0] == T0;
   eqns = {pde, bcL, bcR, ic};
   depVar = T[x, t];
   assumptions = {L > 0, χ > 0, Γ > 0};
   substitutions = {K[1] → n};
   simplifications = {n ∈ Integers};
   values = {description, pde, bcL, bcR, ic,
     eqns, depVar, assumptions, substitutions, simplifications};
   names = {"description", "pde", "bcL", "bcR", "ic", "eqns",
     "depVar", "assumptions", "substitutions", "simplifications"};
   AssociationThread[names, values]];

Module[{soln, G},
 soln = DSolveHeatEquation[A1];
 AppendTo[A1, "soln" → soln];
 Print@ShowPDESetup[A1];
 A1["soln"]]
```

Dubin problem 4.2.2 Broiling Steak

Homogeneous heat equation, inhomogeneous Neumann boundary conditions



Out[11]=

```
T^(0,1)[x, t] == χ T^(2,0)[x, t]
```

DSolve cannot immediately solve this problem — because of the inhomogeneous boundary conditions.

The solution strategy is to write T(x,t) as the sum of a term that explicitly satisfies the boundary conditions T_{bc}(x) and a term that has homogeneous boundary conditions T_h(x,t).

In[12]:=

```
w[1] = T[x, t] == Tbc[x, t] + Th[x, t]
```

Out[12]=

```
T[x, t] == Tbc[x, t] + Th[x, t]
```

A suitable choice for T_{bc} is

In[13]:=
```
w[2] = Tbc[x] == Γ/(L κ) (x - L/2)^2
```
Out[13]=
$$T_{bc}[x] == \frac{\left(-\frac{L}{2}+x\right)^2 \Gamma}{L\,\kappa}$$

for which

In[14]:=
```
{(D[#, x]) & /@ w[2] /. x → 0, (D[#, x]) & /@ w[2] /. x → L}
```
Out[14]=
$$\left\{T_{bc}'[0] == -\frac{\Gamma}{\kappa},\ T_{bc}'[L] == \frac{\Gamma}{\kappa}\right\}$$

The equation for T_h is

In[15]:=
```
w[3] = T^(0,1)[x, t] == χ T^(2,0)[x, t] /. T → Function[{x, t}, Tbc[x, t] + Th[x, t]] /.
    Tbc → Function[{x, t}, Γ/(L κ) (x - L/2)^2] // Expand
```
Out[15]=
$$T_h^{(0,1)}[x, t] == \frac{2\,\Gamma\,\chi}{L\,\kappa} + \chi\, T_h^{(2,0)}[x, t]$$

or

In[16]:=
```
w[4] = MapEqn[(# - χ Th^(2,0)[x, t]) &, w[3]]
```
Out[16]=
$$T_h^{(0,1)}[x, t] - \chi\, T_h^{(2,0)}[x, t] == \frac{2\,\Gamma\,\chi}{L\,\kappa}$$

In[17]:=
```
w[5] = Th[x, 0] == T0 - Tbc[x] /. (w[2] // ER)
```
Out[17]=
$$T_h[x, 0] == -\frac{\left(-\frac{L}{2}+x\right)^2 \Gamma}{L\,\kappa} + T_0$$

The heat equation is now inhomogeneous, with the right hand side representing a source term. The boundary conditions are homogeneous $T_h(0, t) = T_h(L,t)$. Also, the initial condition for T_h = T - T_{bc} has become more complicated. However, it will be possible to find a solution in terms of the eigenfunctions of the homogeneous form of the PDE for T_h

I attempt direct use of DSolve for this transformed problem

In[18]:=
```
A1b =
  Module[{description, pde, bcL, bcR, ic, eqns, depVar,
    assumptions, substitutions, simplifications, names, values},
   description = "Dubin problem 4.2.2 Broiling Steak\nTransformed inhomogeneous
      heat equation\n homogeneous Neumann boundary conditions";
   pde = D[Th[x, t], t] - χ D[Th[x, t], {x, 2}] == (2 Γ χ)/(L κ);
   bcL = κ Derivative[1, 0][Th][0, t] == 0; (* homogeneous von Neumann *)
   bcR = κ Derivative[1, 0][Th][L, t] == 0; (* homogeneous von Neumann *)
   ic = Th[x, 0] == T0 - ((-L/2 + x)^2 Γ)/(L κ);
   eqns = {pde, bcL, bcR, ic};
   depVar = Th[x, t];
   assumptions = {L > 0, χ > 0, Γ > 0};
   substitutions = {K[1] → n};
   simplifications = {n ∈ Integers};
   values = {description, pde, bcL, bcR, ic,
     eqns, depVar, assumptions, substitutions, simplifications};
   names = {"description", "pde", "bcL", "bcR", "ic", "eqns",
     "depVar", "assumptions", "substitutions", "simplifications"};
   AssociationThread[names, values]];

Module[{soln, G},
 soln = DSolveHeatEquation[A1b];
 AppendTo[A1b, "soln" → soln];
 Print@ShowPDESetup[A1b];
 A1b["soln"]]
```

Dubin problem 4.2.2 Broiling Steak
Transformed inhomogeneous heat equation
homogeneous Neumann boundary conditions

$$\kappa \frac{\partial \mathrm{Th}(0,t)}{\partial 0} = 0 \qquad \frac{\partial \mathrm{Th}(x,t)}{\partial t} - \chi \frac{\partial^2 \mathrm{Th}(x,t)}{\partial x^2} = \frac{2 \Gamma \chi}{\kappa L} \qquad \kappa \frac{\partial \mathrm{Th}(L,t)}{\partial L} = 0$$

$$\mathrm{Th}(x,0) = \mathrm{T0} - \frac{\Gamma \left(x - \frac{L}{2}\right)^2}{\kappa L}$$

Out[19]=
$$\mathrm{Th}^{(0,1)}[x,t] == \chi \left(\frac{2\Gamma}{L \kappa} + \mathrm{Th}^{(2,0)}[x,t]\right)$$

The transformed problem isn't solved either.

2 Solution

A solution technique for such problems is to represent the source term and initial condition in terms of

expansions of the eigenfunctions of
the homogeneous equation for T_h(x,t). Separation of variable leads to

In[20]:=
```
w2[1] = A1b["pde"][[1]] == 0 /. Th → Function[{x, t}, 𝒯[t] ψ[x]];
w2[1] = MapEqn[(#/(𝒯[t] ψ[x])) &, w2[1]] // Expand
```

Out[21]=
$$\frac{\mathcal{T}'[t]}{\mathcal{T}[t]} - \frac{\chi\,\psi''[x]}{\psi[x]} == 0$$

The separated equations are

In[22]:=
```
w2[2] = {w2[1][[1, 1]] == -λ, w2[1][[1, 2]] == λ}
```

Out[22]=
$$\left\{\frac{\mathcal{T}'[t]}{\mathcal{T}[t]} == -\lambda,\ -\frac{\chi\,\psi''[x]}{\psi[x]} == \lambda\right\}$$

The latter constitutes a Sturm-Liouville equation problem that DSolve can handle.

In[23]:=
```
w2[3] = DSolve[{χ ψ''[x] == -λ ψ[x], ψ'[0] == 0, ψ'[L] == 0},
    ψ[x], x, Assumptions → {χ > 0, L > 0, λ ∈ Reals}][[1, 1]] /. λ → λn
```

Out[23]=
$$\psi[x] \to \begin{cases} C[1]\,\mathrm{Cos}\left[x\sqrt{\frac{\lambda_n}{\chi}}\right] & (L > 0\ \&\&\ \lambda_n == 0\ \&\&\ \chi > 0)\ ||\ (\dot{n} \in \mathrm{Integers}\ \&\& \\ & \quad ((\dot{n} \geq 0\ \&\&\ L > 0\ \&\&\ \lambda_n > 0\ \&\&\ \chi == (L^2\lambda_n)/(\pi^2 + 4\,\dot{n}\,\pi^2 + 4\,\dot{n}^2\pi^2))\ || \\ & \quad (\dot{n} \geq 1\ \&\&\ L > 0\ \&\&\ \lambda_n > 0\ \&\&\ \chi == \frac{L^2\lambda_n}{4\,\dot{n}^2\pi^2}))) \\ 0 & \mathrm{True} \end{cases}$$

In[24]:=
```
w2[4] = Refine[w2[3] /. ṇ → n, Assumptions → {n ∈ Integers, χ > 0, L > 0}]
```

Out[24]=
$$\psi[x] \to \begin{cases} C[1]\,\mathrm{Cos}\left[\frac{x\sqrt{\lambda_n}}{\sqrt{\chi}}\right] & \lambda_n == 0\ ||\ (n \geq 0\ \&\&\ \lambda_n > 0\ \&\&\ \chi == (L^2\lambda_n)/(\pi^2 + 4\,n\,\pi^2 + 4\,n^2\pi^2))\ || \\ & (n \geq 1\ \&\&\ \lambda_n > 0\ \&\&\ \chi == \frac{L^2\lambda_n}{4\,n^2\pi^2}) \\ 0 & \mathrm{True} \end{cases}$$

This result is awkward and not immediately informative. I will solve the problem manually in Appendix A

In[25]:=
$$\mathrm{w2[5]} = \left\{\lambda_n \to \frac{n^2\pi^2\chi}{L^2},\ \psi_n[x] \to \mathrm{Cos}\left[\frac{x\sqrt{\lambda_n}}{\sqrt{\chi}}\right]\right\}$$

Out[25]=
$$\left\{\lambda_n \to \frac{n^2\pi^2\chi}{L^2},\ \psi_n[x] \to \mathrm{Cos}\left[\frac{x\sqrt{\lambda_n}}{\sqrt{\chi}}\right]\right\}$$

The general solution can be written

In[26]:=
$$w2[6] = T_h[x, t] == \sum_{n=0}^{\infty} \mathcal{T}_n[t]\, \psi_n[x]$$

Out[26]=
$$T_h[x, t] == \sum_{n=0}^{\infty} \mathcal{T}_n[t]\, \psi_n[x]$$

where it remains to determine the explicit time dependence. I substitute this form into the inhomogeneous PDE for T_h

In[27]:=
$$w2[7] = A1b[\text{"pde"}] \;/.\; Th \to Function\left[\{x, t\}, \sum_{n=0}^{\infty} \mathcal{T}_n[t]\, \psi_n[x]\right]$$

Out[27]=
$$\sum_{n=0}^{\infty} \psi_n[x]\, \mathcal{T}_n'[t] - \chi \sum_{n=0}^{\infty} \mathcal{T}_n[t]\, \psi_n''[x] == \frac{2\,\Gamma\,\chi}{L\,\kappa}$$

In[28]:=
$$w2[8] = w2[7] \;/.\; \psi_n''[x] \to -\frac{\lambda_n}{\chi}\psi_n[x]$$

Out[28]=
$$-\chi \sum_{n=0}^{\infty} -\frac{\lambda_n \mathcal{T}_n[t]\, \psi_n[x]}{\chi} + \sum_{n=0}^{\infty} \psi_n[x]\, \mathcal{T}_n'[t] == \frac{2\,\Gamma\,\chi}{L\,\kappa}$$

The source term is expanded in terms of eigenfunctions

In[29]:=
$$w2[9] = w2[8] \;/.\; \frac{2\,\Gamma\,\chi}{L\,\kappa} \to \sum_{n=0}^{\infty} f_n\, \psi_n[x]$$

Out[29]=
$$-\chi \sum_{n=0}^{\infty} -\frac{\lambda_n \mathcal{T}_n[t]\, \psi_n[x]}{\chi} + \sum_{n=0}^{\infty} \psi_n[x]\, \mathcal{T}_n'[t] == \sum_{n=0}^{\infty} f_n\, \psi_n[x]$$

Consider the case n = 0

In[30]:=
$$w2[10] = w2[9] \;/.\; \infty \to 0 \;/.\; \lambda_0 \to 0$$

Out[30]=
$$\psi_0[x]\, \mathcal{T}_0'[t] == f_0\, \psi_0[x]$$

In[31]:=
$$w2[11] = DSolve[w2[10], \mathcal{T}_0[t], t][[1, 1]] \;/.\; C[1] \to A_0$$

Out[31]=
$$\mathcal{T}_0[t] \to A_0 + t\, f_0$$

For n ≥ 1

In[32]:=
$$w2[12] = w2[9] \;/.\; Sum[a\_, b\_] \to a$$

Out[32]=
$$\lambda_n \mathcal{T}_n[t]\, \psi_n[x] + \psi_n[x]\, \mathcal{T}_n'[t] == f_n\, \psi_n[x]$$

where the rule Sum[a_, b_] → a is just a Mathematica rule that has the effect of extracting the summand.

In[33]:=
```
w2[13] = DSolve[w2[12], 𝒯n[t], t][[1, 1]] /. C[1] → An
```

Out[33]=
$$\mathcal{T}_n[t] \to e^{-t\lambda_n} A_n + \frac{f_n}{\lambda_n}$$

Thus, the time dependent term $\mathcal{T}$(t) is given by

In[34]:=
```
w2[14] = 𝒯[t] == 𝒯0[t] + Sum_{n=1}^{∞} 𝒯n[t] /. w2[11] /. w2[13]
```

Out[34]=
$$\mathcal{T}[t] == A_0 + t f_0 + \sum_{n=1}^{\infty}\left(e^{-t\lambda_n} A_n + \frac{f_n}{\lambda_n}\right)$$

and T_h(x,t)

In[35]:=
```
w2[15] =
 Th[x, t] == 𝒯0[t] ψ0[x] + Sum[𝒯n[t] ψn[x], {n, 1, ∞}] /. w2[11] /. w2[13] /.
  ψ0[x] → 1
```

Out[35]=
$$T_h[x, t] == A_0 + t f_0 + \sum_{n=1}^{\infty}\left(e^{-t\lambda_n} A_n + \frac{f_n}{\lambda_n}\right) \psi_n[x]$$

The coefficients A_n are determined by the initial condition

In[36]:=
```
w2[16] = w2[15] /. t → 0
```

Out[36]=
$$T_h[x, 0] == A_0 + \sum_{n=1}^{\infty}\left(A_n + \frac{f_n}{\lambda_n}\right) \psi_n[x]$$

Explicitly

In[37]:=
```
w2[17] = w2[16] /. (A1b["ic"] /. Th → Th // ER)
```

Out[37]=
$$T0 - \frac{\left(-\frac{L}{2} + x\right)^2 \Gamma}{L \kappa} == A_0 + \sum_{n=1}^{\infty}\left(A_n + \frac{f_n}{\lambda_n}\right) \psi_n[x]$$

The initial condition is also expanded in eigenfunctions

In[38]:=
```
w2[18] = w2[17] /. T0 - ((-L/2 + x)^2 Γ)/(L κ) → g0 ψ0[x] + Sum_{n=1}^{∞} gn ψn[x] /. ψ0[x] → 1
```

Out[38]=
$$g_0 + \sum_{n=1}^{\infty} g_n \psi_n[x] == A_0 + \sum_{n=1}^{\infty}\left(A_n + \frac{f_n}{\lambda_n}\right) \psi_n[x]$$

In[39]:=
```
w2[19] = w2[18] /. ∞ → 0;
w2[19] = Solve[w2[19], A0]〚1, 1〛
```
Out[40]= $A_0 \to g_0$

For n ≥ 1

In[41]:=
```
w2[20] = w2[18] /. Sum[a_, b_] → a /. w2[19]
```
Out[41]= $g_0 + g_n\, \psi_n[x] == g_0 + \left(A_n + \frac{f_n}{\lambda_n}\right) \psi_n[x]$

In[42]:=
```
w2[21] = Solve[w2[20], An]〚1, 1〛
```
Out[42]= $A_n \to \frac{-f_n + g_n\, \lambda_n}{\lambda_n}$

With these results

In[43]:=
```
w2[22] = w2[15] /. w2[19] /. w2[21] // ExpandAll
```
Out[43]= $T_h[x, t] == t\, f_0 + g_0 + \sum_{n=1}^{\infty} \left(\frac{f_n}{\lambda_n} + \frac{e^{-t\,\lambda_n} \left(-f_n + g_n\, \lambda_n\right)}{\lambda_n}\right) \psi_n[x]$

For this particular problem, the f_n and g_n are

In[44]:=
$$\text{w2[23]} = \left\{f_n == \frac{\int_0^L \frac{2\,\Gamma\,\chi}{L\,\kappa} \mathrm{Cos}\left[\frac{n\,\pi\,x}{L}\right] dx}{\int_0^L \mathrm{Cos}\left[\frac{n\,\pi\,x}{L}\right]^2 dx},\ f_0 == \frac{1}{L} \int_0^L \frac{2\,\Gamma\,\chi}{L\,\kappa} dx\right\} \text{ // Refine[\#, n} \in \text{Integers] \& // ER}$$

Out[44]= $\left\{f_n \to 0,\ f_0 \to \frac{2\,\Gamma\,\chi}{L\,\kappa}\right\}$

In[45]:=
$$\text{w2[24]} = \left\{g_n == \frac{\int_0^L \left(T0 - \frac{\left(-\frac{L}{2}+x\right)^2 \Gamma}{L\,\kappa}\right) \mathrm{Cos}\left[\frac{n\,\pi\,x}{L}\right] dx}{\int_0^L \mathrm{Cos}\left[\frac{n\,\pi\,x}{L}\right]^2 dx},\ g_0 == \frac{1}{L} \int_0^L \left(T0 - \frac{\left(-\frac{L}{2}+x\right)^2 \Gamma}{L\,\kappa}\right) dx\right\} \text{ // Refine[\#, n} \in \text{Integers] \& // ER}$$

Out[45]= $\left\{g_n \to -\frac{2\left(1 + (-1)^n\right) L\,\Gamma}{n^2\,\pi^2\,\kappa},\ g_0 \to \frac{L\,T0 - \frac{L^2\,\Gamma}{12\,\kappa}}{L}\right\}$

The explicit form is

In[46]:=
```
w2[5]
```

Out[46]=
$$\left\{\lambda_n \to \frac{n^2 \pi^2 \chi}{L^2},\ \psi_n[x] \to \mathrm{Cos}\left[\frac{x\sqrt{\lambda_n}}{\sqrt{\chi}}\right]\right\}$$

In[47]:=
```
w2[25] =
  w2[22] /. Sum → Inactive[Sum] /. w2[23] /. w2[24] //. w2[5] // PowerExpand // Expand
```

Out[47]=
$$T_h[x, t] == T0 - \frac{L\Gamma}{12\kappa} + \frac{2 t \Gamma \chi}{L\kappa} + \sum_{n=1}^{\infty} -\frac{2\left(1+(-1)^n\right) e^{-\frac{n^2\pi^2 t \chi}{L^2}} L\Gamma\,\mathrm{Cos}\left[\frac{n\pi x}{L}\right]}{n^2\pi^2\kappa}$$

where I have declared Sum to be inactive to prevent early evaluation.

Finally, recall that the calculation started with

In[48]:=
```
w2[26] = T[x, t] == Th[x, t] + Tbc[x]
```

Out[48]=
$$T[x, t] == T_{bc}[x] + T_h[x, t]$$

So, recalling previous results

In[49]:=
```
w2[27] = w2[26] /. (w[2] // ER) /. (w2[25] // ER)
```

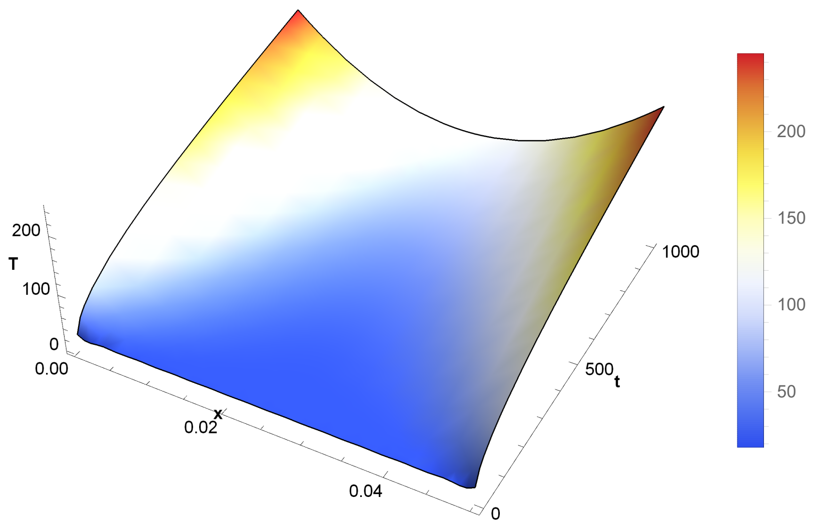
Out[49]=
$$T[x, t] == T0 - \frac{L\Gamma}{12\kappa} + \frac{\left(-\frac{L}{2}+x\right)^2 \Gamma}{L\kappa} + \frac{2 t \Gamma \chi}{L\kappa} + \sum_{n=1}^{\infty} -\frac{2\left(1+(-1)^n\right) e^{-\frac{n^2\pi^2 t \chi}{L^2}} L\Gamma\,\mathrm{Cos}\left[\frac{n\pi x}{L}\right]}{n^2\pi^2\kappa}$$

In[50]:=
```
Clear[TSoln];
TSoln[x_, t_, L_, χ_, Γ_, κ_, T0_, nMax_] :=
```
$$T0 - \frac{L\Gamma}{12\kappa} + \frac{\left(-\frac{L}{2}+x\right)^2 \Gamma}{L\kappa} + \frac{2 t \Gamma \chi}{L\kappa} + \sum_{n=1}^{nMax} -\frac{2\left(1+(-1)^n\right) e^{-\frac{n^2\pi^2 t \chi}{L^2}} L\Gamma\,\mathrm{Cos}\left[\frac{n\pi x}{L}\right]}{n^2\pi^2\kappa} \ \text{// Activate}$$

In[52]:=

```
Module[{L = 0.05 (* m *), χ = 3 × 10^-7 (*m^2/s*), Γ = 10 (*kW/m^2*) 1000
   (*J/kW*), C = 3 × 10^6 (*Joule/(m^3K)*), T0 = 20, κ, nMax = 20, α},
 κ = χ C;
 Plot3D[TSoln[x, t, L, χ, Γ, κ, T0, nMax], {x, 0, L}, {t, 0, 1000},
  ColorFunction → "TemperatureMap", AxesLabel → {Stl["x"], Stl["t"], Stl["T"]},
  Mesh → False, Boxed → False, PlotLegends → Automatic]]
```

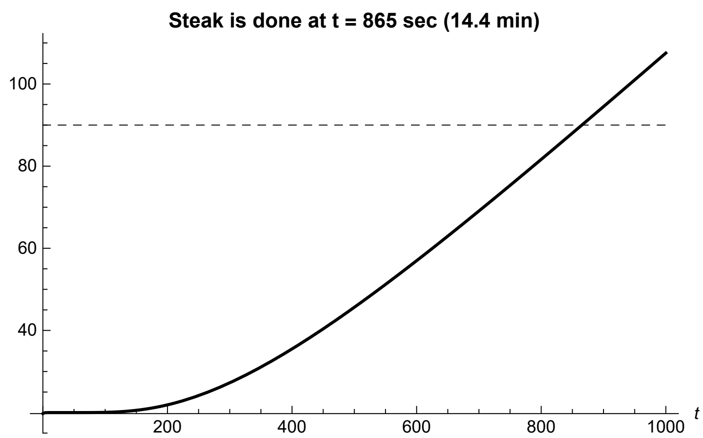
Out[52]=



Dubin’s problem calls for determining how long it takes the central temperature to reach 250 degrees

In[53]:=
```
Module[{L = 0.05 (* m *), χ = 3×10^-7 (*m^2/s*),
  Γ = 10 (*kW/m^2*) 1000 (*J/kW*), C = 3×10^6 (*Joule/(m^3K)*),
  T0 = 20, κ, nMax = 20, tMax = 1000, Ttarget = 90, refLine, root, lab},
 κ = χ C;
 refLine = {Directive[Black, Dashed], Line[{{0, Ttarget}, {tMax, Ttarget}}] };
 root =
  FindRoot[TSoln[L/2, tDone, L, χ, Γ, κ, T0, nMax] == Ttarget, {tDone, 1000}][[1, 2]] ;
 lab = Stl@StringForm["Steak is done at t = `` sec (`` min)",
    Round@root, NF1[root / 60]];
 Plot[TSoln[L/2, t, L, χ, Γ, κ, T0, nMax], {t, 0, 1000},
  PlotStyle → Black, PlotLabel → lab,
  AxesLabel → {Stl["t"], Stl["T"], PlotStyle → {Black, Blue}},
  PlotLabel → lab, Epilog → {refLine}]]
```

Out[53]=



Difficulties arise if you attempt to check this analytical separation of variables result by using NDSolve to solve the problem numerically. As the problem is specified, the initial condition T(x,0) = T_0 (constant) is inconsistent with the boundary conditions that specify constant energy flux at x = 0 and x = L and hence that $\frac{\partial T}{\partial x} \neq 0$ at the boundaries.

Appendix A: The spatial eigenvalue problem

In[54]:=
```
wA[1] = DSolve[-χ ψ''[x] == λ ψ[x], ψ[x], x][[1, 1]] // RE
```

Out[54]=

$$\psi[x] == C[1] \operatorname{Cos}\left[\frac{x\sqrt{\lambda}}{\sqrt{\chi}}\right] + C[2] \operatorname{Sin}\left[\frac{x\sqrt{\lambda}}{\sqrt{\chi}}\right]$$

Apply the homogeneous von Neumann boundary conditions

In[55]:=
```
wA[2] = MapEqn[D[#, x] &, wA[1]]
```
Out[55]=
$$\psi'[x] == \frac{\sqrt{\lambda}\, C[2] \operatorname{Cos}\left[\frac{x\sqrt{\lambda}}{\sqrt{\chi}}\right]}{\sqrt{\chi}} - \frac{\sqrt{\lambda}\, C[1] \operatorname{Sin}\left[\frac{x\sqrt{\lambda}}{\sqrt{\chi}}\right]}{\sqrt{\chi}}$$

In[56]:=
```
wA[3] = Solve[wA[2] /. x → 0 /. ψ'[0] → 0, C[2]][[1, 1]]
```
Out[56]=
$$C[2] \to 0$$

Thus

In[57]:=
```
wA[4] = wA[1] /. wA[3]
```
Out[57]=
$$\psi[x] == C[1] \operatorname{Cos}\left[\frac{x\sqrt{\lambda}}{\sqrt{\chi}}\right]$$

Apply the second boundary condition

In[58]:=
```
wA[5] = MapEqn[D[#, x] &, wA[4]] /. x → L /. ψ'[L] → 0
```
Out[58]=
$$0 == -\frac{\sqrt{\lambda}\, C[1] \operatorname{Sin}\left[\frac{L\sqrt{\lambda}}{\sqrt{\chi}}\right]}{\sqrt{\chi}}$$

The eigenvalues are determined by

In[59]:=
```
wA[6] = Solve[L √λ / √χ == n π, λ][[1, 1]] /. λ → λn // Simplify // PowerExpand
```
Out[59]=
$$\lambda_n \to \frac{n^2 \pi^2 \chi}{L^2}$$

with n >= 0, The eigenfunctions are

In[60]:=
```
wA[7] = wA[4] /. wA[6] /. C[1] → 1 /. ψ → ψn /. λ → λn // ER
```
Out[60]=
$$\psi_n[x] \to \operatorname{Cos}\left[\frac{x\sqrt{\lambda_n}}{\sqrt{\chi}}\right]$$

In[61]:=
```
wA[8] = {wA[6], wA[7]}
```
Out[61]=
$$\left\{\lambda_n \to \frac{n^2 \pi^2 \chi}{L^2},\ \psi_n[x] \to \operatorname{Cos}\left[\frac{x\sqrt{\lambda_n}}{\sqrt{\chi}}\right]\right\}$$

I consider other approaches to this problem. In the main text, I noticed that

In[62]:=
```
wA[9] = DSolve[{χ ψ''[x] == -λ ψ[x], ψ'[0] == 0, ψ'[L] == 0},
    ψ[x], x, Assumptions → {χ > 0, L > 0, λ ∈ Reals}][[1, 1]] /. λ → λn
```

Out[62]=

$$\psi[x] \to \begin{cases} C[1]\,\mathrm{Cos}\left[x\sqrt{\frac{\lambda_n}{\chi}}\right] & (L>0\ \&\&\ \lambda_n == 0\ \&\&\ \chi>0)\ ||\ \Big(\dot{n}\in \mathrm{Integers}\ \&\& \\ & \quad \big((\dot{n}\geq 0\ \&\&\ L>0\ \&\&\ \lambda_n>0\ \&\&\ \chi == (L^2\lambda_n)/(\pi^2+4\dot{n}\pi^2+4\dot{n}^2\pi^2))\ || \\ & \quad (\dot{n}\geq 1\ \&\&\ L>0\ \&\&\ \lambda_n>0\ \&\&\ \chi == \frac{L^2\lambda_n}{4\dot{n}^2\pi^2})\big)\Big) \\ 0 & \mathrm{True} \end{cases}$$

or

In[63]:=
```
wA[10] = Refine[wA[9] /. ṅ → n, Assumptions → {n ∈ Integers, χ > 0, L > 0}]
```

Out[63]=

$$\psi[x] \to \begin{cases} C[1]\,\mathrm{Cos}\left[\frac{x\sqrt{\lambda_n}}{\sqrt{\chi}}\right] & \lambda_n == 0\ ||\ (n\geq 0\ \&\&\ \lambda_n>0\ \&\&\ \chi == (L^2\lambda_n)/(\pi^2+4n\pi^2+4n^2\pi^2))\ || \\ & (n\geq 1\ \&\&\ \lambda_n>0\ \&\&\ \chi == \frac{L^2\lambda_n}{4n^2\pi^2}) \\ 0 & \mathrm{True} \end{cases}$$

Consider the implications of the eigenvalues implied by these two conditions for a numerical example

In[64]:=
```
Module[{χ = 1, L = 1, soln1, soln2},
 soln1 = Solve[χ == L^2 λ/(π^2 + 4 n π^2 + 4 n^2 π^2), λ][[1, 1]];
 soln2 = Solve[χ == L^2 λ/(4 n^2 π^2), λ][[1, 1]];
 {Table[λ /. soln1, {n, 0, 5}], Table[λ /. soln2, {n, 0, 5}] } ]
```

Out[64]=

$$\{\{\pi^2, 9\pi^2, 25\pi^2, 49\pi^2, 81\pi^2, 121\pi^2\}, \{0, 4\pi^2, 16\pi^2, 36\pi^2, 64\pi^2, 100\pi^2\}\}$$

or, combining the results of these two conditions

In[65]:=
```
Module[{χ = 1, L = 1, soln1, soln2},
 soln1 = Solve[χ == L^2 λ/(π^2 + 4 n π^2 + 4 n^2 π^2), λ][[1, 1]];
 soln2 = Solve[χ == L^2 λ/(4 n^2 π^2), λ][[1, 1]];
 Sort@Flatten@{Table[λ /. soln1, {n, 0, 5}], Table[λ /. soln2, {n, 0, 5}]} ]
```

Out[65]=

$$\{0, \pi^2, 4\pi^2, 9\pi^2, 16\pi^2, 25\pi^2, 36\pi^2, 49\pi^2, 64\pi^2, 81\pi^2, 100\pi^2, 121\pi^2\}$$

Thus, in an oblique way, DSolve yields the eigenvalues that are expected

In[66]:=
```
λn → n^2 π^2
```
Out[66]=
$$\lambda_n \to n^2 \pi^2$$

Yet another approach is to use the Mathematica function *DEigensystem* to generate a discrete sequence of eigenvalues and eigenfunctions

In[67]:=
```
resultsNeumann =
 DEigensystem[-1/χ D[u[x], {x, 2}] + NeumannValue[0, True], u[x], {x, 0, L}, 5]
```
Out[67]=
$$\left\{\left\{0, \frac{\pi^2}{L^2 \chi}, \frac{4\pi^2}{L^2 \chi}, \frac{9\pi^2}{L^2 \chi}, \frac{16\pi^2}{L^2 \chi}\right\}, \left\{1, \text{Cos}\left[\frac{\pi x}{L}\right], \text{Cos}\left[\frac{2\pi x}{L}\right], \text{Cos}\left[\frac{3\pi x}{L}\right], \text{Cos}\left[\frac{4\pi x}{L}\right]\right\}\right\}$$

This discrete results can be generalized by making use of the Mathematica function *FindSequenceFunction*

In[71]:=
```
NthEigenValueAndEigenFunction[resultsNeumann]
```
Out[71]=
$$\left\{\lambda_n \to \frac{(-1+n)^2 \pi^2}{L^2 \chi}, \psi_n \to \text{Cos}\left[\frac{n \pi x}{L}\right]\right\}$$

and nothing is lost by changing the dummy index

In[72]:=
```
NthEigenValueAndEigenFunction[resultsNeumann] /. -1 + n → n
```
Out[72]=
$$\left\{\lambda_n \to \frac{n^2 \pi^2}{L^2 \chi}, \psi_n \to \text{Cos}\left[\frac{n \pi x}{L}\right]\right\}$$

In[69]:=
```
Clear[NthEigenValueAndEigenFunction];
NthEigenValueAndEigenFunction[resultsDEigensystem_] :=
 Module[{λList, ψList, ψArguments, ψFcn},
  λList = FindSequenceFunction[resultsDEigensystem[[1]], n ];
  (* 10-27-16 The following is a kludge to handle the case where the first eigenfcn
     is Cos[0 π x] = 1. In that case I just neglect the first term *)
  ψList = Select[resultsDEigensystem[[2]], Or[(Head[#] == Cos), (Head[#] == Sin)] &];
  ψArguments = FindSequenceFunction[ψList[[All, 1]] , n];
  ψFcn = Head[ψList[[1]]];
  {λn → λList, ψn → ψFcn[ψArguments]} ]
```

Functions

In[3]:=
```
Clear[ShowPDESetup];
ShowPDESetup[A_] := Module[{top = 1.0, right = 1.0,
   boundaries, labels, textInterior, textIC, textBCL, textBCR},
  boundaries = Line /@ {{{0, 0}, {right, 0}},
     {{0, 0}, {0, top}}, {{right, 0}, {right, top}}};
  labels = Text[PhysicsForm[A[#[[1]]] ], #[[2]]] & /@
    {{"pde", {right / 2, top / 2}}, {"ic", {right / 2, 0.0}},
     {"bcL", {0.0, top / 2}}, {"bcR", {right, top / 2}}};
  Graphics[{Directive[Black, Thick], boundaries, labels}, Axes → False,
   AspectRatio → 0.25, ImageSize → 500, PlotLabel → Stl[A["description"]]]]
```

In[4]:=
```
Clear[DSolveHeatEquation];
DSolveHeatEquation[A_] :=
 Module[{soln},
  soln =
   DSolve[A["eqns"], A["depVar"], {x, t}, Assumptions → A["assumptions"]][[1, 1]];
  soln = soln //. A["substitutions"];
  soln = Simplify[soln, A["simplifications"]];
  soln]
```