Dubin 4.2.3 Microwaving Steak 11-14-16

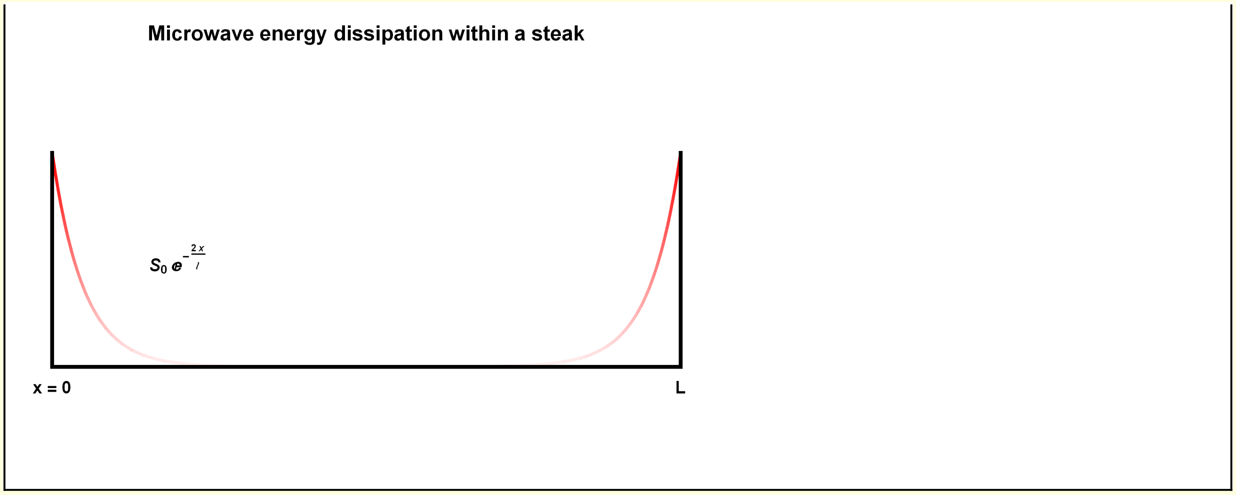
N. T. Gladd

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 *)
```

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 Problem 4.2(3) considers the cooking of a steak in a microwave oven. The problem illustrates a spatially nonuniform source of heat concentrated at the edges of the steak.



1 Part (a) of Problem 4.2(3)

The problem states that microwave power P is dissipated near the edge of radiated steak, within a skin depth l, and part (a) of the problem asked that the energy density parameter S_0 be related to P.

The microwave energy per unit time delivered to the steak is

In[10]:=
```
w1[1] = D[ℰ[t], t] == P
```
Out[10]= $\mathcal{E}'[t] == P$

The energy density dissipated within the steak per unit time is

In[11]:=
```
w1[2] = D[ℰ[t], t] == A Integrate[e^(-2x/ℓ) S0 + e^(2(-L+x)/ℓ) S0, {x, 0, L}]
```
Out[11]= $\mathcal{E}'[t] == A\left(1 - e^{-\frac{2L}{\ell}}\right)\ell S_0$

The problem states that l << L, so

In[12]:=
```
w1[3] = w1[2][[1]] == Limit[w1[2][[2]], L → ∞, Assumptions → {ℓ > 0}]
```
Out[12]= $\mathcal{E}'[t] == A\,\ell\,S_0$

Thus,

In[13]:=
```
w1[4] = Solve[w1[3] /. Solve[w1[1], ℰ'[t]][[1, 1]], S0][[1, 1]] // RE
```
Out[13]= $S_0 == \frac{P}{A\,\ell}$

In part (b) it is specified that P = 5 kWatts and that the surface area is 5000 cm^2. Thus the energy density is

In[14]:=
```
w1[5] = w1[4] /. {P → 5000 Watts, A → 5000 cm^2, ℓ → 1 cm}
```
Out[14]= $S_0 == \frac{\text{Watts}}{\text{cm}^3}$

or, in MKS units,

In[15]:=
```
w1[6] = w1[5] /. Watts → Joule/s /. cm → 0.01 m
```
Out[15]= $S_0 == \frac{1.\times 10^6\ \text{Joule}}{\text{m}^3\ \text{s}}$

Part b of the problem requires solution of the heat equation to determine how the temperature of the steak increases over time. That requires a source term in the form of S_0/C where C is the specific heat of the steak. The specific heat C for steak is not specified in the problem statement but is given in problem 4.2(3) as 3 10^6 $\frac{\text{Joule}}{m^3 K^\circ}$. Thus I will take $\mathcal{S}$ = 1/3 $\frac{K^\circ}{s}$.

In[16]:=
```
w1[7] = 𝒮 == S0/C /. (w1[6] // ER) /. C → 3 × 10^6 Joule/(m^3 K)
```

Out[16]=
```
𝒮 == 0.333333 K / s
```

2 Part (b) of Problem 4.2(3)

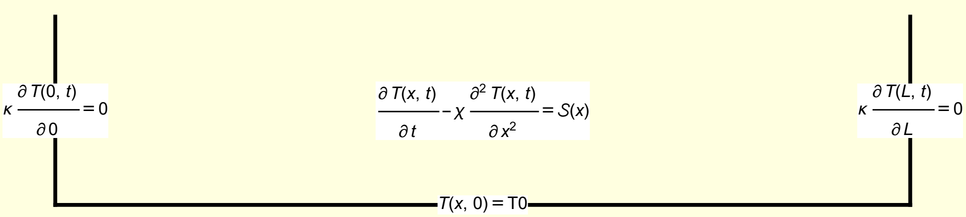
I construct an Association that encapsulates information about this problem, and then apply the function *DSolveHeatEquation* that attempts to solve this problem directly using the Mathematica function *DSolve*.

In[17]:=
```
A1 =
  Module[{description, pde, bcL, bcR, ic, eqns, depVar,
    assumptions, substitutions, simplifications, names, values},
   description = "Dubin problem 4.2(3) Microsaving a steak Steak\nHomogeneous
      heat equation, inhomogeneous Neumann boundary conditions";
   pde = D[T[x, t], t] - χ D[T[x, t], {x, 2}] == 𝒮[x];
   (* insulating boundaries *)
   bcL = κ Derivative[1, 0][T][0, t] == 0; (* homogeneous von Neumann *)
   bcR = κ Derivative[1, 0][T][L, t] == 0; (* homogeneous 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(3) Microsaving a steak Steak

Homogeneous heat equation, inhomogeneous Neumann boundary conditions



Out[18]=

$\mathcal{S}[x] + \chi\, T^{(2,0)}[x, t] == T^{(0,1)}[x, t]$

DSolve cannot immediately solve this problem — because of the inhomogeneous heat equation.

The solution technique involves representing the source term as an expansion of the eigenfunctions of the homogeneous PDE. Separation of variables leads to

In[19]:=
```
w2[1] = A1["pde"][[1]] == 0 /. T → Function[{x, t}, 𝒯[t] ψ[x]];
w2[1] = MapEqn[(#/(𝒯[t] ψ[x])) &, w2[1]] // Expand
```

Out[20]=

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

The separated equations are

In[21]:=
```
w2[2] = {w2[1][[1, 1]] == -λ, w2[1][[1, 2]] == λ }
```

Out[21]=

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

The eigenvalues and eigenfunctions are (see *Dubin 4.2.2 Broiling Steak 11-13-16* for derivation)

In[22]:=

$$\text{w2[3]} = \left\{\lambda_n \to \frac{n^2 \pi^2 \chi}{L^2}, \psi_n[x] \to \text{Cos}\left[\frac{x\sqrt{\lambda_n}}{\sqrt{\chi}}\right]\right\}$$

Out[22]=

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

The general solution can be written

In[23]:=

$$\text{w2[4]} = T_h[x, t] == \sum_{n=0}^{\infty} \mathcal{T}_n[t]\, \psi_n[x]$$

Out[23]=

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

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

In[24]:=
```
A1["pde"]
```

Out[24]=

$$T^{(0,1)}[x, t] - \chi\, T^{(2,0)}[x, t] == \mathcal{S}[x]$$

In[25]:=

$$\text{w2[5]} = \text{A1["pde"]} \;/.\; T \to \text{Function}\left[\{x, t\}, \sum_{n=0}^{\infty} \mathcal{T}_n[t]\, \psi_n[x]\right]$$

Out[25]=

$$\sum_{n=0}^{\infty} \psi_n[x]\, \mathcal{T}_n'[t] - \chi \sum_{n=0}^{\infty} \mathcal{T}_n[t]\, \psi_n''[x] == \mathcal{S}[x]$$

In[26]:=

$$\text{w2[6]} = \text{w2[5]} \;/.\; \psi_n''[x] \to -\frac{\lambda_n}{\chi} \psi_n[x]$$

Out[26]=

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

The source term is expanded in terms of eigenfunctions. To emphasize the structure of the calculation I'll defer specifying the specific form of $\mathcal{S}$[x]

In[27]:=
```
w2[7] = w2[6] /. 𝒮[x] → Sum[f_n ψ_n[x], {n, 0, ∞}]
```
Out[27]=
$$-\chi \sum_{n=0}^{\infty} -\frac{\lambda_n \mathcal{T}_n[t]\, \psi_n[x]}{\chi} + \sum_{n=0}^{\infty} \psi_n[x]\, \mathcal{T}_n'[t] == \sum_{n=0}^{\infty} f_n\, \psi_n[x]$$

Consider the case n = 0

In[28]:=
```
w2[8] = w2[7] /. ∞ → 0 /. λ0 → 0
```
Out[28]=
$$\psi_0[x]\, \mathcal{T}_0'[t] == f_0\, \psi_0[x]$$

In[29]:=
```
w2[9] = DSolve[w2[8], 𝒯0[t], t][[1, 1]] /. C[1] → A0
```
Out[29]=
$$\mathcal{T}_0[t] \to A_0 + t\, f_0$$

For n ≥ 1

In[30]:=
```
w2[10] = w2[7] /. Sum[a_, b_] → a
```
Out[30]=
$$\lambda_n \mathcal{T}_n[t]\, \psi_n[x] + \psi_n[x]\, \mathcal{T}_n'[t] == f_n\, \psi_n[x]$$

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

In[31]:=
```
w2[11] = DSolve[w2[10], 𝒯n[t], t][[1, 1]] /. C[1] → An
```
Out[31]=
$$\mathcal{T}_n[t] \to e^{-t \lambda_n} A_n + \frac{f_n}{\lambda_n}$$

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

In[32]:=
```
w2[12] = 𝒯[t] == 𝒯0[t] + Sum[𝒯n[t], {n, 1, ∞}] /. w2[9] /. w2[11]
```
Out[32]=
$$\mathcal{T}[t] == A_0 + t\, f_0 + \sum_{n=1}^{\infty} \left(e^{-t \lambda_n} A_n + \frac{f_n}{\lambda_n}\right)$$

and *T*(x,t)

In[33]:=
```
w2[13] = T[x, t] == 𝒯0[t] ψ0[x] + Sum[𝒯n[t] ψn[x], {n, 1, ∞}] /. w2[9] /. w2[11] /.
   ψ0[x] → 1
```
Out[33]=
$$T[x, t] == A_0 + t\, f_0 + \sum_{n=1}^{\infty} \left(e^{-t \lambda_n} A_n + \frac{f_n}{\lambda_n}\right) \psi_n[x]$$

The coefficients A_n are determined by the initial condition

In[34]:=
```
w2[14] = w2[13] /. t → 0
```
Out[34]=
$$T[x,0] == A_0 + \sum_{n=1}^{\infty}\left(A_n + \frac{f_n}{\lambda_n}\right)\psi_n[x]$$

The initial condition is also expanded in terms of eigenfunctions

In[35]:=
```
w2[15] = w2[14] /. T[x, 0] → g0 ψ0[x] + Sum[gn ψn[x], {n, 1, ∞}] /. ψ0[x] → 1
```
Out[35]=
$$g_0 + \sum_{n=1}^{\infty} g_n \psi_n[x] == A_0 + \sum_{n=1}^{\infty}\left(A_n + \frac{f_n}{\lambda_n}\right)\psi_n[x]$$

In[36]:=
```
w2[16] = w2[15] /. ∞ → 0;
w2[16] = Solve[w2[16], A0][[1, 1]]
```
Out[37]=
$$A_0 \to g_0$$

For n ≥ 1

In[38]:=
```
w2[17] = w2[15] /. Sum[a_, b_] → a /. w2[16]
```
Out[38]=
$$g_0 + g_n \psi_n[x] == g_0 + \left(A_n + \frac{f_n}{\lambda_n}\right)\psi_n[x]$$

In[39]:=
```
w2[18] = Solve[w2[17], An][[1, 1]]
```
Out[39]=
$$A_n \to \frac{-f_n + g_n \lambda_n}{\lambda_n}$$

With these results

In[40]:=
```
w2[19] = w2[13] /. w2[16] /. w2[18] // ExpandAll
```
Out[40]=
$$T[x,t] == t f_0 + g_0 + \sum_{n=1}^{\infty}\left(\frac{f_n}{\lambda_n} + \frac{e^{-t\lambda_n}(-f_n + g_n\lambda_n)}{\lambda_n}\right)\psi_n[x]$$

For this particular problem, the f_n and g_n are

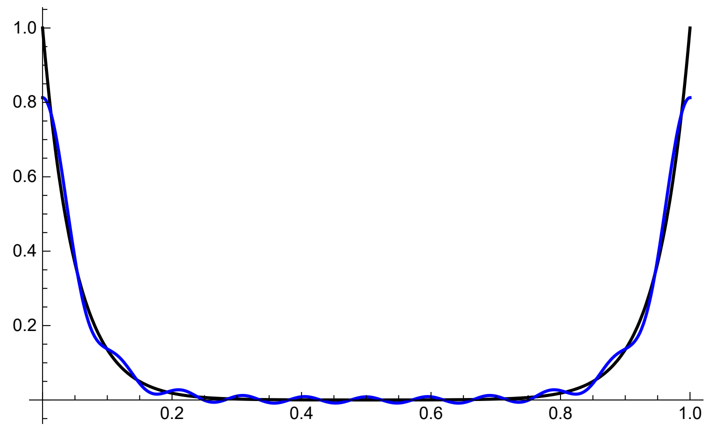
In[41]:=
```
w2[20] = {fn == (∫_0^L S (e^(-2x/ℓ) + e^(2(-L+x)/ℓ)) Cos[nπx/L] dx) / (∫_0^L Cos[nπx/L]^2 dx),
    f0 == 1/L ∫_0^L S (e^(-2x/ℓ) + e^(2(-L+x)/ℓ)) dx} // Refine[#, n ∈ Integers] & // Simplify // ER
```
Out[41]=
$$\left\{f_n \to \frac{4\left(1+(-1)^n\right) e^{-\frac{2L}{\ell}}\left(-1+e^{\frac{2L}{\ell}}\right) L \ell S}{4L^2 + n^2\pi^2\ell^2},\ f_0 \to \frac{\left(1 - e^{-\frac{2L}{\ell}}\right)\ell S}{L}\right\}$$

I check this result

In[42]:=

```
Module[{𝒮 = 1, L = 1, ℓ = 0.1, nMax = 20, fApprox},
 fApprox[x_, L_, ℓ_, 𝒮_, nMax_] := ((1 - ⅇ^(-2 L/ℓ)) ℓ 𝒮)/L +
   Sum[((4 (1 + (-1)^n) ⅇ^(-2 L/ℓ) (-1 + ⅇ^(2 L/ℓ)) L ℓ 𝒮) / (4 L^2 + n^2 π^2 ℓ^2)) Cos[n π x/L], {n, 1, nMax}];
 Plot[{𝒮 (ⅇ^(-2 x/ℓ) + ⅇ^(2 (-L+x)/ℓ)), fApprox[x, L, ℓ, 𝒮, nMax]}, {x, 0, L},
  PlotStyle → {Black, Blue}, PlotRange → All]]
```

Out[42]=



In[43]:=

$$\text{w2[22]} = \left\{g_n == \frac{\int_0^L T_0 \operatorname{Cos}\left[\frac{n \pi x}{L}\right] dx}{\int_0^L \operatorname{Cos}\left[\frac{n \pi x}{L}\right]^2 dx},\; g_0 == \frac{1}{L}\int_0^L T_0\, dx\right\} \text{ //}$$

```
    Refine[#, n ∈ Integers] & // Simplify // ER
```

Out[43]=

$$\{g_n \to 0,\; g_0 \to T_0\}$$

The explicit form for T(x,t) is

In[44]:=

```
w2[23] = w2[19] /. Sum → Inactive[Sum] /. w2[20] /. w2[22] //. w2[3] /. T₀ → T0 /.
   ∞ → nMax // PowerExpand
```

Out[44]=

$$T[x, t] == T0 + \frac{\left(1 - e^{-\frac{2L}{\ell}}\right) t \ell \mathcal{S}}{L} + \sum_{n=1}^{nMax}\left(\frac{4\left(1+(-1)^n\right) e^{-\frac{2L}{\ell}}\left(-1+e^{\frac{2L}{\ell}}\right) L^3 \ell \mathcal{S}}{n^2 \pi^2 \left(4L^2 + n^2 \pi^2 \ell^2\right) \chi} - \frac{4\left(1+(-1)^n\right) e^{-\frac{2L}{\ell} - \frac{n^2 \pi^2 t \chi}{L^2}}\left(-1+e^{\frac{2L}{\ell}}\right) L^3 \ell \mathcal{S}}{n^2 \pi^2 \left(4L^2 + n^2 \pi^2 \ell^2\right) \chi}\right) \operatorname{Cos}\left[\frac{n \pi x}{L}\right]$$

In[45]:=

```
Clear[TSoln];
```

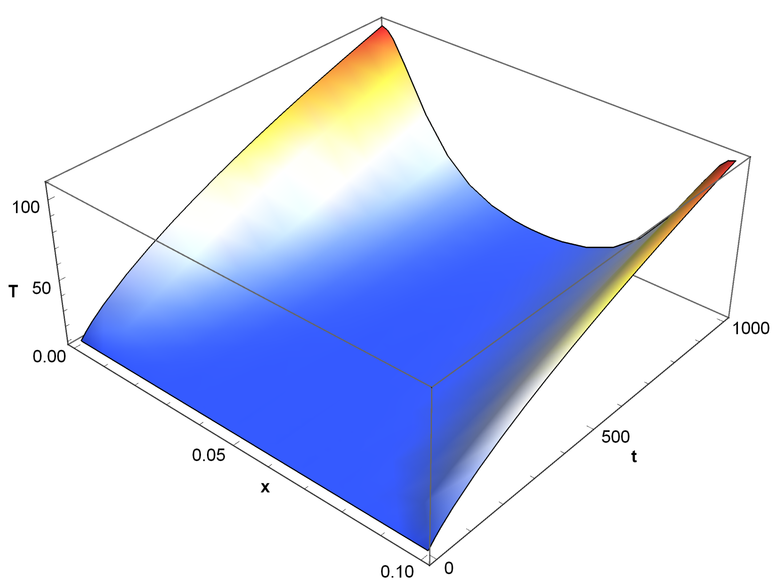
$$\text{TSoln}[x\_, t\_, L\_, \ell\_, \mathcal{S}\_, \chi\_, T0\_, \text{nMax}\_] := T0 + \frac{\left(1 - e^{-\frac{2L}{\ell}}\right) t \ell \mathcal{S}}{L} + \sum_{n=1}^{\text{nMax}} \left(\frac{4\left(1 + (-1)^n\right) e^{-\frac{2L}{\ell}} \left(-1 + e^{\frac{2L}{\ell}}\right) L^3 \ell \mathcal{S}}{n^2 \pi^2 \left(4 L^2 + n^2 \pi^2 \ell^2\right) \chi} - \frac{4\left(1 + (-1)^n\right) e^{-\frac{2L}{\ell} - \frac{n^2 \pi^2 t \chi}{L^2}} \left(-1 + e^{\frac{2L}{\ell}}\right) L^3 \ell \mathcal{S}}{n^2 \pi^2 \left(4 L^2 + n^2 \pi^2 \ell^2\right) \chi} \right) \text{Cos}\left[\frac{n \pi x}{L}\right]$$

```
    // Activate
```

In[47]:=

```
Module[{L = 0.10(* m *), ℓ = 0.01 (*m*), χ = 2×10^-7 (*m^2/s*), C = 3×10^6
   (*Joule/(m^3K°)*), S = 1/3 (*K°/s*), T0 = 10, nMax = 20, tMax = 1000},
 Plot3D[TSoln[x, t, L, ℓ, S, χ, T0, nMax], {x, 0, L},
  {t, 0, tMax}, ColorFunction → "TemperatureMap",
  AxesLabel → {Stl["x"], Stl["t"], Stl["T"]}, Mesh → False, PlotRange → All]]
```

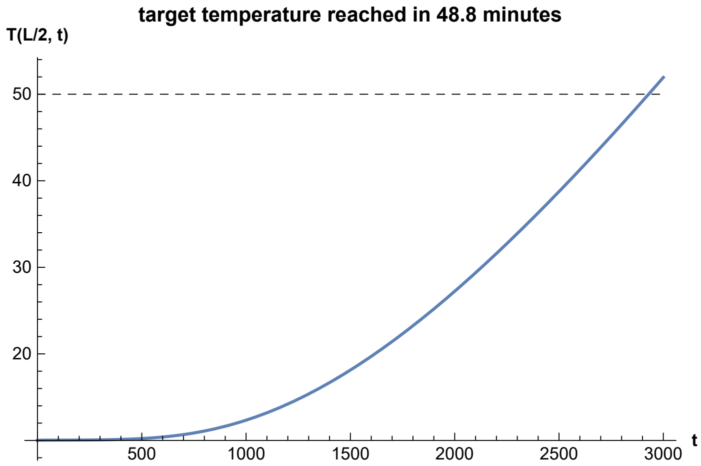
Out[47]=



The problem calls for finding the time for the central temperature in the steak to rise to 50 degrees.

In[48]:=
```
Module[{L = 0.10(* m *), ℓ = 0.01 (*m*), χ = 2×10^-7
  (*m^2/s*), C = 3×10^6 (*Joule/(m^3K°)*), S = 1/3 (*K°/s*),
  T0 = 10, nMax = 20, tMax = 3000, Ttarget = 50, root, lab},
 root = FindRoot[TSoln[L/2, t, L, ℓ, S, χ, T0, nMax] == Ttarget, {t, tMax}][[1, 2]];
 lab = Stl@StringForm["target temperature reached in `` minutes", NF1[root/60]];
 Plot[TSoln[L/2, t, L, ℓ, S, χ, T0, nMax], {t, 0, tMax},
  Epilog → {Directive[Black, Dashed], Line[{{0, Ttarget}, {tMax, Ttarget}}] },
  PlotLabel → lab, AxesLabel → {Stl["t"], Stl["T(L/2, t)"]}]]
```

Out[48]=



This takes longer to cook than I expected. I expect that my choice of specific C heat (3×10^6 (*Joule/(m^3K°)) was larger than what Dubin intended.

I check the separation of variable solution by solving this problem numerically

In[49]:=
```
SOLNNUMERICAL = Module[{L = 0.10(* m *), ℓ = 0.01 (*m*), χ = 2×10^-7 (*m^2/s*),
   C = 3×10^6 (*Joule/(m^3K°)*), S = 1/3 (*K°/s*), T0 = 10, tMax = 3000, eqns},
  f[x_] := S (e^(-2x/ℓ) + e^(2(-L+x)/ℓ));
  eqns = {D[T[x, t], t] == χ D[T[x, t], {x, 2}] + f[x],
    Derivative[1, 0][T][0, t] == 0, Derivative[1, 0][T][L, t] == 0, T[x, 0] == T0};
  NDSolve[eqns, T, {x, 0, L}, {t, 0, tMax}][[1, 1]] ]
```

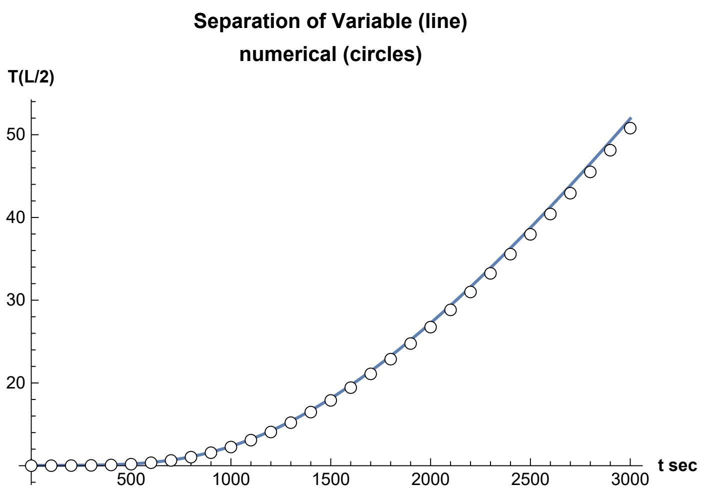
Out[49]=
```
T → InterpolatingFunction[ Domain: {{0., 0.1}, {0., 3.00×10^3}}
                           Output: scalar ]
```

I compare the analytical and numerical solutions

In[50]:=

```
Module[{L = 0.10(* m *), ℓ = 0.01 (*m*),
  χ = 2×10^-7 (*m^2/s*), C = 3×10^6 (*Joule/(m^3K°)*), S = 1/3
  (*K°/s*), T0 = 10, nMax = 20, tMax = 3000, numerical},
 numerical = Table[{t, T[L/2, t] /. SOLNNUMERICAL}, {t, 0, 3000, 100}];
 Plot[TSoln[L/2, t, L, ℓ, S, χ, T0, nMax],
  {t, 0, tMax}, Epilog → {OC[#, Black] & /@ numerical},
  AxesLabel → {Stl["t sec"], Stl["T(L/2)"]},
  PlotLabel → Stl["Separation of Variable (line)\nnumerical (circles)"]]]
```

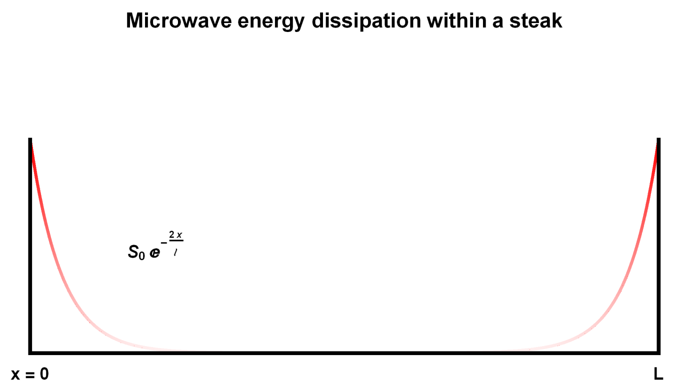
Out[50]=



Graphics

In[51]:=
```
Module[{L = 10, S0 = 1, ls = 1, lineBase, lineLeft, lineRight, textS},
 lineBase = {Directive[Black, Thick], Line[{{0, 0}, {L, 0}}] };
 lineLeft = {Directive[Black, Thick],
   Line[{{0, 0}, {0, 1}}], Stl@Text["x = 0", {0.0, -0.1}] };
 lineRight = {Directive[Black, Thick], Line[{{L, 0}, {L, 1}}],
   Stl@Text["L", {L, -0.1}] };
 textS = Text[Stl@TraditionalForm[S0 Exp[-2 x/ℓ]], {2, 0.5}];
 Plot[{S0 Exp[-2 x / ls], S0 Exp[2 (x - L) / ls]},
  {x, 0, L}, PlotRange → All, PlotStyle → Black,
  ColorFunction → Function[{x, y}, Blend[{Lighter[Red, 0.95], Red}, y]],
  PlotLabel → Stl["Microwave energy dissipation within a steak"],
  AxesLabel → {Stl["x"], Stl["S(J/(m^3 s))"]}, Axes → None,
  Epilog → {lineBase, lineLeft, lineRight, textS}, PlotRangePadding → 0.5]]
```

Out[51]=



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]
```