Heat Equation Problem -Dirichlet BCs - 11-07-16

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 - *Heat Equation Problem -Dirichlet BCs - 10-13-16.* Much earlier work circa 2008.

Purpose

I solve a heat equation problem from Chapter 3 of *Numerical and Analytical Methods for Scientists and Engineers, Using Mathematica*, Daniel Dubin. The specific example (p 214) considers a static heat source with static Dirichlet boundary conditions. Dubin, consistent with the title of the book, solved this problem utilizing Mathematica to facilitate the calculation of eigenfunctions that arise in the method of separation of variables. I will use Mathematica for additional aspects of the solution process. Also, I mention that the Mathematica technology for solving such problems has evolved in recent years.

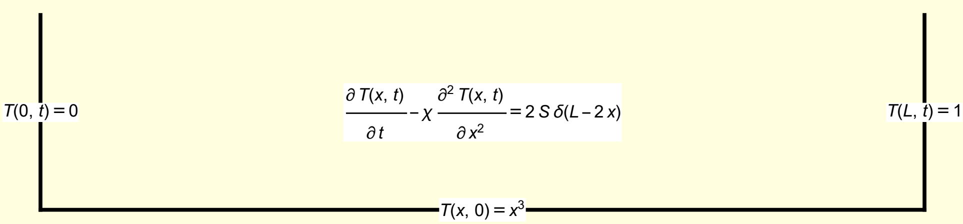
I construct an Association that encapsulates information about this problem. I then apply the function *DSolveHeatEquation* that attempts to solve this problem using the Mathematica function *DSolve*.

In[12]:=

```
A1 =
  Module[{description, pde, bcL, bcR, ic, eqns,
    assumptions, substitutions, simplifications, names, values},
   description = "Dubin example, p214 Static point heat source";
   pde = D[T[x, t], t] - χ D[T[x, t], {x, 2}] == S DiracDelta[x - L/2];
   bcL = T[0, t] == 0;
   bcR = T[L, t] == 1;
   ic = T[x, 0] == x^3;
   eqns = {pde, bcL, bcR, ic};
   assumptions = {L > 0, χ > 0};
   substitutions = {K[1] → n};
   simplifications = {n ∈ Integers};
   values = {description, pde, bcL, bcR,
     ic, eqns, assumptions, substitutions, simplifications};
   names = {"description", "pde", "bcL", "bcR", "ic", "eqns",
     "assumptions", "substitutions", "simplifications"};
   AssociationThread[names, values]];

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

Dubin example, p214 Static point heat source



Out[13]=

```
T^(0,1)[x, t] - χ T^(2,0)[x, t] == 2 S DiracDelta[L - 2 x]
```

DSolve cannot directly solve this problem. This is not surprising, given that both the PDE and the boundary conditions are inhomogeneous. A solution method in such cases is to write T(x, t) = T_h(x, t) + T_{bc}(x), where T_h(x, t) satisfies a PDE with homogeneous boundary conditions and T_{bc}(x) is a time-independent function that satisfies the boundary conditions. Since T_{bc}(x) does not depend on time, it is often called the equilibrium solution.

I perform a sequence of operations that generate the governing equations for T_h(x,t) and T_{bc}(x).

In[14]:=

```
w[1] = A1["pde"]
```

Out[14]=

```
T^(0,1)[x, t] - χ T^(2,0)[x, t] == 2 S DiracDelta[L - 2 x]
```

In[15]:=
```
w[2] = w[1] /. T → Function[{x, t}, Th[x, t] + Tbc[x]] // Expand
```
Out[15]=
$$-\chi\, T_{bc}''[x] + T_h^{(0,1)}[x, t] - \chi\, T_h^{(2,0)}[x, t] == 2\, S\, \text{DiracDelta}[L - 2\, x]$$

In[16]:=
```
w[3] = MapEqn[(# + χ Tbc''[x]) &, w[2]]
```
Out[16]=
$$T_h^{(0,1)}[x, t] - \chi\, T_h^{(2,0)}[x, t] == 2\, S\, \text{DiracDelta}[L - 2\, x] + \chi\, T_{bc}''[x]$$

The equations for T_h(x,t) and T_{bc}(x) are

In[17]:=
```
w[4] = {w[3]〚1〛 == 0, w[3]〚2〛 == 0}
```
Out[17]=
$$\{T_h^{(0,1)}[x, t] - \chi\, T_h^{(2,0)}[x, t] == 0,\ 2\, S\, \text{DiracDelta}[L - 2\, x] + \chi\, T_{bc}''[x] == 0\}$$

In[18]:=
```
w[4] // ColumnForm // PhysicsForm
```
Out[18]//TraditionalForm=
$$\frac{\partial T_h(x,t)}{\partial t} - \chi\, \frac{\partial^2 T_h(x,t)}{\partial x^2} = 0$$
$$\chi\, \frac{\partial^2 T_{bc}(x)}{\partial x^2} + 2\, S\, \delta(L - 2\, x) = 0$$

Solve for $T_{bc}(x)$

In[19]:=
```
wbc[1] = w[4]〚2〛
```
Out[19]=
$$2\, S\, \text{DiracDelta}[L - 2\, x] + \chi\, T_{bc}''[x] == 0$$

Find a solution of this equation that satisfies both of the Dirichlet boundary conditions on T(x,t)

In[20]:=
```
wbc[2] = DSolve[{wbc[1], Tbc[0] == 0, Tbc[L] == 1}, Tbc[x], x]〚1, 1〛
```
Out[20]=
$$T_{bc}[x] \to \frac{1}{2\, L\, \chi}\left(2\, x\, \chi - L^2\, S\, \text{HeavisideTheta}[-L] + L\, S\, x\, \text{HeavisideTheta}[-L] + L\, S\, x\, \text{HeavisideTheta}[L] + L^2\, S\, \text{HeavisideTheta}[-L + 2\, x] - 2\, L\, S\, x\, \text{HeavisideTheta}[-L + 2\, x]\right)$$

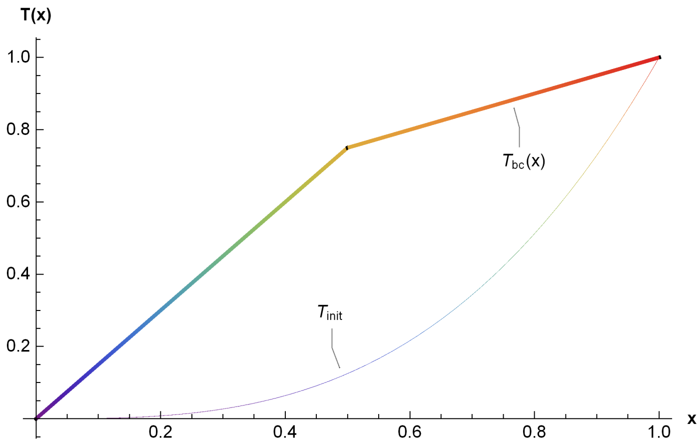
For future convenience, construct functions for $T_{bc}(x)$ and the original initial condition

In[21]:=
```
Clear[Tbc, Tinit];
Tbc[x_, L_, χ_, S_] :=
  1/(2 L χ) (2 x χ - L^2 S HeavisideTheta[-L] + L S x HeavisideTheta[-L] + L S x HeavisideTheta[L] +
     L^2 S HeavisideTheta[-L + 2 x] - 2 L S x HeavisideTheta[-L + 2 x]);
Tinit[
   x_] :=
  x^3;
```

Visualize the equilibrium solution and initial profiles

In[24]:=
```
Module[{L = 1, χ = 1, S = 1},
 Plot[{Callout[Tbc[x, L, χ, S], "T_bc(x)", 0.75], Callout[Tinit[x], "T_init", 0.5]},
  {x, 0, 1}, AxesLabel → {Stl["x"], Stl["T(x)"]}, PlotStyle → {Thick, Thin},
  ColorFunction → Function[{x, y}, ColorData["Rainbow"][y]]]]
```

Out[24]=



T_h(x,t) satisfies

In[25]:=
```
wh[1] = w[4][[1]]
```

Out[25]= $T_h^{(0,1)}[x, t] - \chi T_h^{(2,0)}[x, t] == 0$

Because the boundary conditions for T(x,t) were incorporated in $T_{bc}(x)$, the boundary conditions for T_h(x,t) are homogeneous

In[26]:=
```
wh[2] = {Th[0, t] == 0, Th[L, t] == 0}
```

Out[26]= $\{T_h[0, t] == 0, T_h[L, t] == 0\}$

The initial condition for T_h(x,t) is

In[27]:=
```
wh[3] = Solve[T[x, t] == Th[x, t] + Tbc[x], Th[x, t]][[1, 1]] /. t → 0
```

Out[27]= $T_h[x, 0] \to T[x, 0] - T_{bc}[x]$

In[28]:=

$$\text{wh}[4] = \text{wh}[3] \text{ /. } T[x, 0] \to x^3 \text{ /. } T_{bc} \to \text{Function}\Big[\{x\}, \frac{1}{2 L \chi} \big(2 x \chi - L^2 S \,\text{HeavisideTheta}[-L] + L S x \,\text{HeavisideTheta}[-L] + L S x \,\text{HeavisideTheta}[L] + L^2 S \,\text{HeavisideTheta}[-L + 2 x] - 2 L S x \,\text{HeavisideTheta}[-L + 2 x]\big)\Big]$$

Out[28]=

$$T_h[x, 0] \to x^3 - \frac{1}{2 L \chi} \big(2 x \chi - L^2 S \,\text{HeavisideTheta}[-L] + L S x \,\text{HeavisideTheta}[-L] + L S x \,\text{HeavisideTheta}[L] + L^2 S \,\text{HeavisideTheta}[-L + 2 x] - 2 L S x \,\text{HeavisideTheta}[-L + 2 x]\big)$$

I use DSolve to solve the PDE for T_h(x,t)

In[29]:=
```
wh[4] = Module[{pde, bcL, bcR, ic},
    pde = D[Th[x, t], t] - χ D[Th[x, t], {x, 2}] == 0;
    bcL = Th[0, t] == 0;
    bcR = Th[L, t] == 0;
    ic = Th[x, 0] == x^3 - 1/(2 L χ) (2 x χ - L^2 S HeavisideTheta[-L] +
          L S x HeavisideTheta[-L] + L S x HeavisideTheta[L] +
          L^2 S HeavisideTheta[-L + 2 x] - 2 L S x HeavisideTheta[-L + 2 x]);
    DSolve[{pde, bcL, bcR, ic}, Th[x, t], {x, t}, Assumptions → {L > 0, χ > 0}]][[
   1, 1]] /. {K[1] → n, Th → T_h}
```

Out[29]=

$$T_h[x, t] \to \sum_{n=1}^{\infty} -\frac{2\, e^{-\frac{n^2 \pi^2 t \chi}{L^2}} \left(\left((-1)^n \left(-n^2 \pi^2 + L^3 \left(-6 + n^2 \pi^2\right)\right) \chi + L\, n\, \pi\, S \,\mathrm{Sin}\left[\frac{n \pi}{2}\right]\right) \mathrm{Sin}\left[\frac{n \pi x}{L}\right]}{n^3 \pi^3 \chi}$$

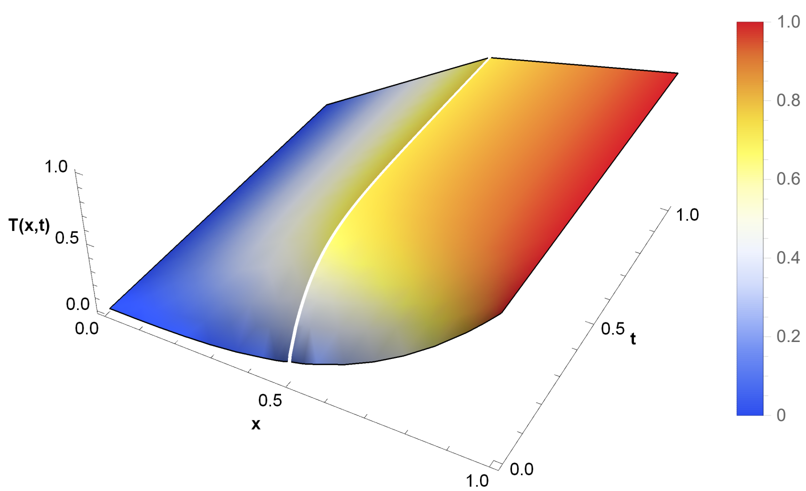
Define functions for T_h(x,t) and T(x,t)

In[30]:=
```
Clear[Th, TSoln];
Th[x_, t_, L_, χ_, S_, nMax_] :=
  Activate[Sum[-2 e^(-n^2 π^2 t χ/L^2) (((-1)^n (-n^2 π^2 + L^3 (-6 + n^2 π^2)) χ + L n π S Sin[n π/2]) Sin[n π x/L])/(n^3 π^3 χ), {n, 1, nMax}]];
TSoln[x_, t_, L_, χ_, S_, nMax_] := Tbc[x, L, χ, S] + Th[x, t, L, χ, S, nMax];
```

Visualize the solution

In[33]:=
```
Module[{L = 1, χ = 1, S = 1, nMax = 20},
 Plot3D[TSoln[x, t, L, χ, S, nMax], {x, 0, L}, {t, 0, 1},
  ColorFunction → "TemperatureMap", AxesLabel → {Stl["x"], Stl["t"], Stl["T(x,t)"]},
  Mesh → False, Boxed → False, PlotLegends → Automatic]]
```

Out[33]=

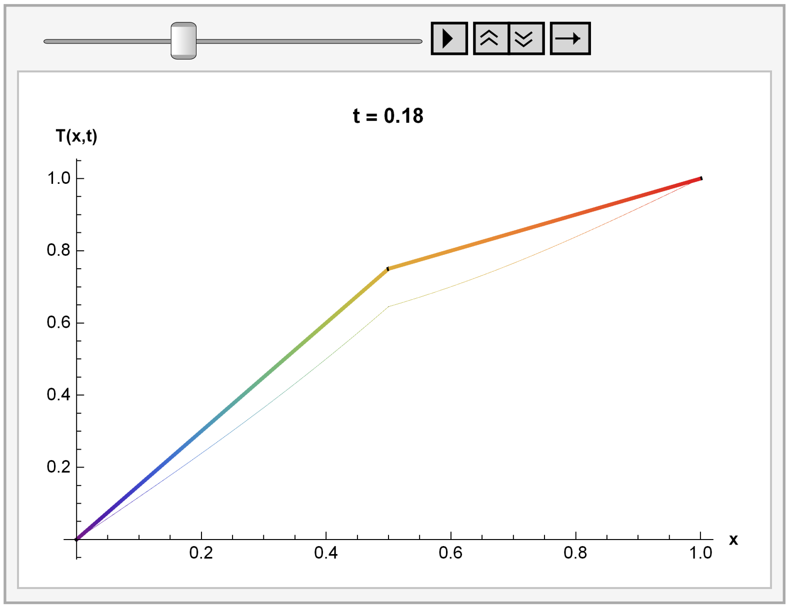


An animation of the relaxation to equilibrium is also interesting

In[34]:=

```
Module[{L = 1, χ = 1, S = 1, nMax = 20, frames},
 frames = MakeFrame[#, L, χ, S, nMax] & /@ Range[0, 0.5, 0.025];
 ListAnimate@frames]
```

Out[34]=



In[3]:=

```
Clear[MakeFrame];
MakeFrame[t_, L_, χ_, S_, nMax_] :=
 Module[{lab},
  lab = Stl@StringForm["t = ``", NF2@t];
  Plot[{TSoln[x, t, L, χ, S, nMax], Tbc[x, L, χ, S]}, {x, 0, L}, PlotLabel → lab,
   AxesLabel → {Stl["x"], Stl["T(x,t)"]}, PlotStyle → {Thin, Thick},
   ColorFunction → Function[{x, y}, ColorData["Rainbow"][y]]]]
```

Functions

In[5]:=

```
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[6]:=
```
Clear[DSolveHeatEquation];
DSolveHeatEquation[A_] :=
 Module[{soln},
  soln = DSolve[A["eqns"], T[x, t], {x, t}, Assumptions → A["assumptions"]][[1, 1]];
  soln = soln //. A["substitutions"];
  soln = Simplify[soln, A["simplifications"]];
  soln]
```