Rectangular Drumhead Redux

Completed and Analyzed in class, April 15, 2025

This is our nineteenth notebook. It builds on the techniques in the eighteenth notebook. The difference between a guitar string and a rectangular drumhead is just the number of dimensions. After graduating to two dimensions, it will (I hope) be fairly apparent how to go to three dimensions.

Rectangular Drumhead — Theory

Back in the fourteenth notebook we had these acceleration formulas for drum-heads:

$$a_{j,k} = v_0{}^2\left(z_{j,k+1} + z_{j,k-1} + z_{j+1,k} + z_{j-1,k} - 4\,z_{j,k}\right)$$

The continuum version of this equation is:

$$\frac{\partial^2 z}{\partial t^2} = v_0{}^2\left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2}\right)$$

Again, let's be clear about the dependent and independent variables. The time t, is most definitely an independent variable. The drumhead is stretched along the x- and y-axes. Those are the independent variables. The dependent variable is the displacement, which we are putting in the z-direction. To summarize, the dependent variable is z, and we want to find it as a function of three independent variables, so we are looking for $z(t, x, y)$.

More Partial Derivatives

Now that we have three independent variables, we will have things like this:

In[1]:=
```
Derivative[0, 0, 2][z][t, x, y] // TraditionalForm
```

Out[1]//TraditionalForm=

$$z^{(0,0,2)}(t, x, y)$$

The Rectangular Drumhead Differential Equation

Recopying what was above, you can give Mathematica this differential equation:

$$\frac{\partial^2 z}{\partial t^2} = v_0{}^2\left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2}\right)$$

In[2]:=
```
Module[{v0 = 1}, Derivative[2, 0, 0][z][t, x, y] ==
  v0^2 (Derivative[0, 2, 0][z][t, x, y] + Derivative[0, 0, 2][z][t, x, y])]
```

Out[2]=

$$z^{(2,0,0)}[t, x, y] == z^{(0,0,2)}[t, x, y] + z^{(0,2,0)}[t, x, y]$$

Adding the Boundary Conditions

We are missing the boundary conditions at the edge of the drumhead.

In[3]:=
```
lengthX = 2;
lengthY = 3 / 2;

Module[{v0 = 1}, {Derivative[2, 0, 0][z][t, x, y] ==
    v0^2 (Derivative[0, 2, 0][z][t, x, y] + Derivative[0, 0, 2][z][t, x, y]),
   z[t, x, 0] == 0, z[t, 0, y] == 0, z[t, x, lengthY] == 0,
   z[t, lengthX, y] == 0}] // TraditionalForm
```

Out[5]//TraditionalForm=

$$\left\{z^{(2,0,0)}(t,x,y)=z^{(0,0,2)}(t,x,y)+z^{(0,2,0)}(t,x,y),\, z(t,x,0)=0,\, z(t,0,y)=0,\, z\left(t,x,\frac{3}{2}\right)=0,\, z(t,2,y)=0\right\}$$

Adding the Initial Conditions

We are also missing any specification of the initial motion of the drumhead. It isn't just going to start vibrating by itself. Here is an initial displacement function:

In[6]:=
```
amplitude = 1;
modeX = 2;
modeY = 1;

lengthX = 2;
lengthY = 3 / 2;

f[x_, y_] := amplitude Sin[modeX Pi x / lengthX] Sin[modeY Pi y / lengthY]
```

As you can see, we can specify a mode number in either axis. Add that to the equations and also have the initial velocity be zero:

In[12]:=
```
rectangularDrumheadProblem = Module[{v0 = 1}, {Derivative[2, 0, 0][z][t, x, y] ==
     v0^2 (Derivative[0, 2, 0][z][t, x, y] + Derivative[0, 0, 2][z][t, x, y]),
    z[t, x, 0] == 0, z[t, 0, y] == 0, z[t, x, lengthY] == 0, z[t, lengthX, y] == 0,
    z[0, x, y] == f[x, y], Derivative[1, 0, 0][z][0, x, y] == 0}];
```

In[13]:=
```
rectangularDrumheadProblem // TraditionalForm
```

Out[13]//TraditionalForm=

$$\left\{z^{(2,0,0)}(t,x,y)=z^{(0,0,2)}(t,x,y)+z^{(0,2,0)}(t,x,y),\, z(t,x,0)=0,\, z(t,0,y)=0,\right.$$
$$\left. z\left(t,x,\frac{3}{2}\right)=0,\, z(t,2,y)=0,\, z(0,x,y)=\sin(\pi x)\sin\left(\frac{2\pi y}{3}\right),\, z^{(1,0,0)}(0,x,y)=0\right\}$$

Making Mathematica Solve the Rectangular Problem

This problem has an exact solution. So we can use NDSolve[] rather than NDSolve[]:

In[14]:=
```
rectangularDrumheadSolutionRule = NDSolve[
   rectangularDrumheadProblem, z, {t, 0, 10}, {x, 0, lengthX}, {y, 0, lengthY}][[1]]
```

Out[14]=
```
{z → InterpolatingFunction[ Domain: {{0., 10.}, {0., 2.}, {0., 1.5}}
                            Output: scalar
                            Data not saved. Save now ]}
```

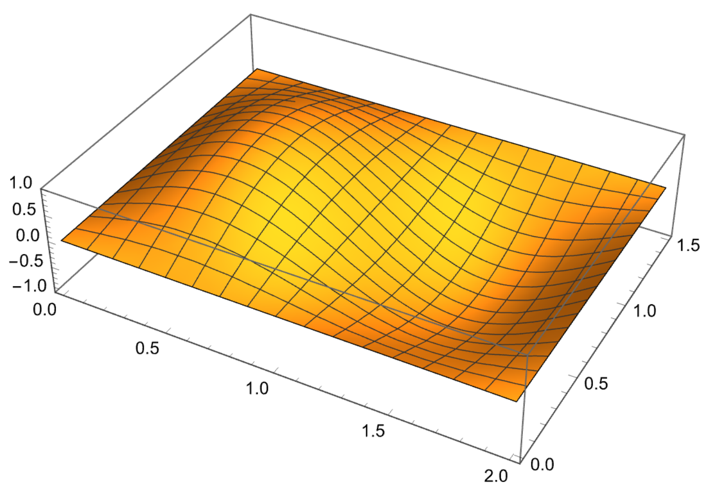
In[15]:=
```
rectangularDrumheadSolution[t_, x_, y_] =
 z[t, x, y] /. rectangularDrumheadSolutionRule
```

Out[15]=
```
InterpolatingFunction[ Domain: {{0., 10.}, {0., 2.}, {0., 1.5}}
                       Output: scalar
                       Data not saved. Save now ][t, x, y]
```

Plotting the Rectangular Solution at *t* = 1/8

In[16]:=
```
Plot3D[rectangularDrumheadSolution[1/8, x, y], {x, 0, lengthX}, {y, 0, lengthY},
 PlotRange → {Automatic, Automatic, {-1.0, 1.0}}, BoxRatios → {lengthX, lengthY, 0.5}]
```

Out[16]=



Animating the Rectangular Solution

In[17]:=
```
plots = Table[Plot3D[rectangularDrumheadSolution[t/15, x, y], {x, 0, lengthX},
    {y, 0, lengthY}, PlotRange → {Automatic, Automatic, {-1.0, 1.0}},
    BoxRatios → {lengthX, lengthY, 0.5}], {t, 0, 150}];
```

Because it is interfering with the quality of the animation to animate 3D plots, notice what I just did. I pre-made 150 (actually it is 151, because I went from 0 to 150) 3D plots. The idea was to do 15 plots per second for a 10 seconds. Now all Animate[] has to do is cycle through them instead of computing them as it cycles through them:

In[19]:=
```
Animate[plots⟦step⟧, {step, 1, 151, 1},
 DefaultDuration → 10, AnimationRunning → False, RefreshRate → 15]
```

Out[19]=