

---

# Guitar String

Completed and Analyzed in class, April 8, 2025

This is our eighteenth notebook. For most of your special projects, the type of work we did in the seventeenth notebook (Harmonic Oscillator Redux) is more relevant than what I am going to show you now. Please keep referring to that notebook. It had everything you need to know about how to put ordinary differential equations into Mathematica.

In this notebook, as part of analyzing a guitar string, we going to put “partial differential equations” into Mathematica.

## Guitar String — Theory

Back in the thirteenth notebook titled “Torsion Waves,” we got our first really good visualization of waves. Torsion waves are perhaps a little more visceral to look at than the transverse waves on a guitar string, but the mathematics is completely equivalent. Here was the angular acceleration formula from the Torsion Waves notebook:

$$\alpha_j = -\omega_0^2(\theta_j - \theta_{j-1}) + \omega_0^2(\theta_{j+1} - \theta_j)$$

In a companion theory notebook titled “The Second Derivative,” I did some casual hand-waving of the kind physicists are prone to do and that mathematicians spend their lives trying to make more rigorous, and convinced you (I hope) that what we really had in the limit that the chunks and the time steps got smaller and smaller was:

$$\frac{\partial^2 \theta}{\partial t^2} = v_0^2 \frac{\partial^2 \theta}{\partial x^2}$$

For a guitar string, in which we call the direction the string is oriented in the  $x$ -axis and for which we call the displacement  $z(t, x)$ , the corresponding equation is:

$$\frac{\partial^2 z}{\partial t^2} = v_0^2 \frac{\partial^2 z}{\partial x^2}$$

## Partial Derivatives and Their Notation

Now we have two variables ( $t$  and  $x$ ), and it we have to clarify for Mathematica which variable we are taking a derivative with respect to. In other words, the obvious generalization of

```
In[25]:= Derivative[2][z][t] // TraditionalForm
Out[25]//TraditionalForm=
z''(t)
```

which would be

```
In[26]:= Derivative[2][z][t, x] // TraditionalForm
Out[26]//TraditionalForm=

$$z''(t, x)$$

```

is ambiguous. So are we taking the derivative with respect to  $t$  or  $x$ ? Mathematica specifies it this way:

```
In[27]:= Derivative[2, 0][z][t, x] // TraditionalForm
Out[27]//TraditionalForm=

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

```

## The Guitar String Differential Equation

Now that we know how Mathematica expects us to specify partial differential equations, we can give it this differential equation:

$$\frac{\partial^2 z}{\partial t^2} = v_0^2 \frac{\partial^2 z}{\partial x^2}$$

Review the previous notebook to see how that was done in for the harmonic oscillator.

```
In[28]:= (* Module[{rootin tootin }]
           "Well... Afore hoo'd bin here three days hoo'd have a
           dozen colliers whewtin' an' tootin' after her every neet." *)

In[29]:= Module[{v0 = 1}, {Derivative[2, 0][z][t, x] == v0^2 Derivative[0, 2][z][t, x]}] //
           TraditionalForm
Out[29]//TraditionalForm=

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

```

## Adding the Boundary Conditions

```
In[30]:= f[x_] :=
           Sum[(32 * Sin[(Pi * K[1]) / 4] * Sin[Pi * x * K[1]]) / (3 * Pi ^ 2 * K[1] ^ 2), {K[1], 1, 10}]

guitarStringProblem = Module[{v0 = 1, length = 1, a = 1, m = 3},
           {Derivative[2, 0][z][t, x] == v0^2 Derivative[0, 2][z][t, x], z[t, 0] == 0,
            z[t, length] == 0, z[0, x] == f[x], Derivative[1, 0][z][0, x] == 0}];
```

## Adding the Initial Conditions

## Making Mathematica Solve the Guitar String Problem

```
In[44]:= NDSolve[guitarStringProblem, z, {t, 0, 10}, {x, 0, 1}]
```

```
Out[44]=
```

```
{ {z → InterpolatingFunction[ Domain: {{0., 10.}, {0., 1.}}  
Output: scalar ] ] }
```

We have the same problems as we had with the harmonic oscillator solution: (1) It is a list of lists of rules, and (2) we haven't given it a name so don't have a convenient way of using it elsewhere. Fix these problems:

```
In[45]:= guitarStringSolutionRule = NDSolve[guitarStringProblem, z, {t, 0, 10}, {x, 0, 1}][[1]]
```

```
Out[45]=
```

```
{ z → InterpolatingFunction[ Domain: {{0., 10.}, {0., 1.}}  
Output: scalar ] }
```

It is still a rule. Turn it into a function that we can plot:

```
In[46]:= guitarStringSolution[t_, x_] = z[t, x] /. guitarStringSolutionRule
```

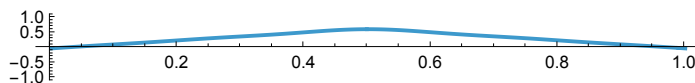
```
Out[46]=
```

```
InterpolatingFunction[ Domain: {{0., 10.}, {0., 1.}}  
Output: scalar ] [t, x]
```

## Plotting the Guitar String at $t = 1/4$

```
In[49]:= Plot[guitarStringSolution[1/4, x],  
{x, 0, 1}, PlotRange → {-1.1, 1.1}, AspectRatio → 0.1]
```

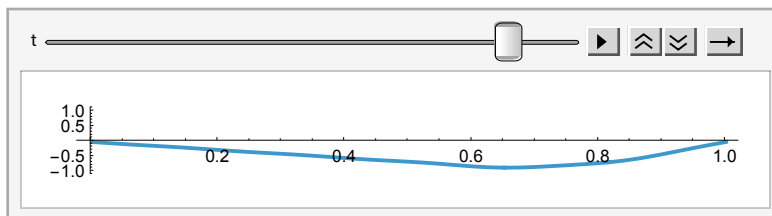
```
Out[49]=
```



## Animating the Guitar String

```
In[40]:= Animate[Plot[guitarStringSolution[t, x], {x, 0, 1},  
PlotRange → {Automatic, {-1.1, 1.1}}, AspectRatio → 0.1],  
{t, 0, 10}, DefaultDuration → 10, AnimationRunning → False]
```

```
Out[40]=
```



## Changing the Mode

Try changing the mode,  $m$ , to some other number, like 3, and see what a harmonic looks like.

## Completely Different Initial Conditions

When a guitar string is plucked with something sharp, we can imagine that instead of an initial shape looking like,

$$z(0, x) = a \sin(m\pi x)$$

instead we have, if  $0 < x < b$ ,

$$z(0, x) = a \frac{x}{b}$$

and if  $b < x < \text{length}$ ,

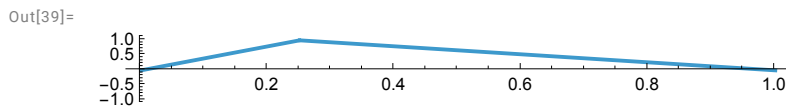
$$z(0, x) = a \frac{L-x}{L-b}$$

Using `Module[]` with  $a=1, L=1, b=1/4$ , define a function  $f[x]$  that is the right-hand side, taking into account the differing behavior when  $x < b$  vs.  $x > b$ :

```
In[38]:= f[x_] := Module[{a = 1, L = 1, b = 1/4}, If[x < b, a  $\frac{x}{b}$ , a  $\frac{L-x}{L-b}$ ]]
```

Plot your function:

```
In[39]:= Plot[f[x], {x, 0, 1}, AspectRatio -> 0.1, PlotRange -> {Automatic, {-1.1, 1.1}}]
```



Does that look like the way a pick might stretch the string (near the bridge) just before it lets it go?