Differential Equations&*Mathematica*

©1999 Bill Davis and Jerry Uhl
Produced by Bruce Carpenter Published by Math Everywhere, Inc.
www.matheverywhere.com

DE.09 The Heat Equation and the Wave Equation
Basics

B.1) Rigging f[t] on [0, 2 L] to get a pure sine fit of f[t] on [0, L]

□B.1.a)

When you go for a fast Fourier fit of a function on an interval [0,L] you usually get a mixture of both Sines and Cosines:

```
Clear[f, t]
f[t_] = t^2 E^(-0.5 t);
L = 2 π;
Chop[ComplexExpand[FastFourierfit[f, L, 4, t]]]
```

```
1.43827 - 0.78213 Cos[t] - 0.29451 Cos[2 t] - 0.243714 Cos[3 t] -
 0.593786 Sin[t] - 0.265601 Sin[2 t] - 0.103904 Sin[3 t]
```

But sometimes you get pure Cosines:

```
Clear[f, t]
f[t_] = t (t - 1)^2 (2 - t);
L = 2;
Chop[ComplexExpand[FastFourierfit[f, L, 6, t]]]
```

```
0.128729 + 0.0781497 Cos[π t] - 0.0802469 Cos[2 π t] - 0.0493827 Cos[3 π t] -
 0.0349794 Cos[4 π t] - 0.028767 Cos[5 π t]
```

Other times you get pure Sines:

```
Clear[f, t]
f[t_] = t (1 - t) (2 - t);
L = 2;
Chop[ComplexExpand[FastFourierfit[f, L, 6, t]]]
```

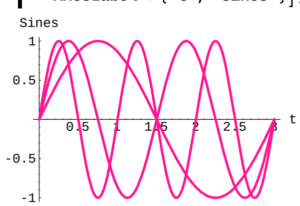
```
0.386895 Sin[π t] + 0.0481125 Sin[2 π t] + 0.0138889 Sin[3 π t] +
 0.00534584 Sin[4 π t] + 0.00199435 Sin[5 π t]
```

The question here is:
How do you recognize when you are going to get a pure Sine fit?

□Answer:

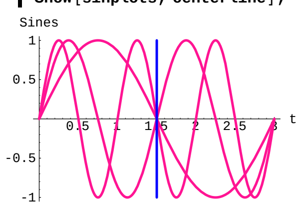
Look at plots of Sin[k 2 Pi/L t] on [0,L] like this:

```
L = 3;
sinplots = Plot[{Sin[(2 π) t / L], Sin[2 (2 π) t / L], Sin[3 (2 π) t / L]},
  {t, 0, L}, PlotStyle → {{Thickness[0.01], DeepPink}},
  AxesLabel → {"t", "Sines"}];
```



Embellish the plot a bit:

```
centerline =
  Graphics[{Blue, Thickness[0.01], Line[{{L/2, -1}, {L/2, 1}}]}];
Show[sinplots, centerline];
```



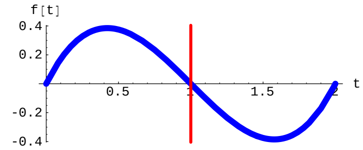
To get all the message, grab both plots and animate slowly.

To the right of the line, each sine plot is the negative mirror image of its plot to the left of the line.

Functions that plot out the same way on [0,L] have fast Fourier fits on [0,L] composed of pure sines.

Look at the plot of this function f[t] on [0,L] for L = 2:

```
Clear[f, t]
f[t_] = t (1 - t) (2 - t);
L = 2;
fplot = Plot[f[t], {t, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}}, AxesLabel → {"t", "f[t]"},
  Epilog → {Red, Thickness[0.01], Line[{{L/2, -1}, {L/2, 1}}]}];
```



To the right of the center line, the plot of f[t] is the negative mirror image of its plot to the left of the center line. This is why you can count on the fast Fourier fit of f[t] on [0,L] to come out as a pure Sine fit.

Try it:

```
Chop[ComplexExpand[FastFourierfit[f, L, 5, t]]]
```

```
0.386759 Sin[π t] + 0.047806 Sin[2 π t] + 0.0133207 Sin[3 π t] +
 0.00431067 Sin[4 π t]
```

```
Chop[ComplexExpand[FastFourierfit[f, L, 10, t]]]
```

```
0.387003 Sin[π t] + 0.0483449 Sin[2 π t] + 0.0142834 Sin[3 π t] +
 0.00597576 Sin[4 π t] + 0.003 Sin[5 π t] + 0.00166509 Sin[6 π t] +
 0.00096271 Sin[7 π t] + 0.000538834 Sin[8 π t] + 0.000243536 Sin[9 π t]
```

Haw!

If you go with a different L, you'll probably lose the pure Sine fit:

```
L = Random[Real, {3, 4}]
```

```
3.86303
```

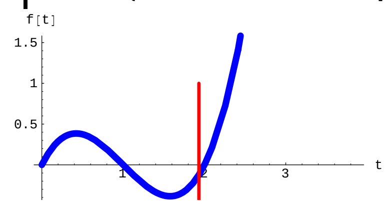
```
Chop[ComplexExpand[FastFourierfit[f, L, 3, t]]]
```

```
1.82802 + 1.19949 Cos[1.62649 t] - 1.88963 Cos[3.25298 t] -
 3.17459 Sin[1.62649 t] - 1.67455 Sin[3.25298 t]
```

For this L, the pure Sine fit on [0,L] was lost!

If you look at the plot of f[t] on [0,L], then you'll see why:

```
Plot[f[t], {t, 0, L},
 PlotStyle → {{Thickness[0.02], Blue}}, AxesLabel → {"t", "f[t]"},
 Epilog → {Red, Thickness[0.01], Line[{{L/2, -1}, {L/2, 1}}]}];
```



For this new L, the plot of f[t] to the right of the center line is NOT the negative mirror image of its plot to the left of the center line.

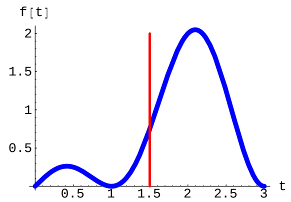
This is why the pure Sine fit was lost on [0,L].

□B.1.b.i) Rigging f[t] on [0, 2 L] to get a pure Sine fit of f[t] on [0,L]

When Fourier first said he could do this, the skeptics laughed at him, but you and Fourier get the last laugh here.

Look at this:

```
Clear[f, t]
f[t_] = t Cos[π t / 2]^2;
L = 3;
fplot = Plot[f[t], {t, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}}, AxesLabel → {"t", "f[t]"},
  Epilog → {Red, Thickness[0.01], Line[{{L/2, 0}, {L/2, 2}}]}];
```



The plot tells you that that this particular function f[t] has no chance of a pure Sine fit on this particular interval [0,L].
Stay with the same L, the same f[t] on [0,L] but rig f[t] on [L, 2 L] as follows:

riggedf[t] = f[t] for 0≤ t ≤ L and
riggedf[t] = f[2 L - t] for L < t ≤ 2 L

and look at some fast Fourier fits of riggedf[t] on [0, 2L]

```
Clear[riggedf, t]
riggedf[t_] := f[t] /; 0 ≤ t ≤ L;
riggedf[t_] := -f[2 L - t] /; L < t ≤ 2 L;
Chop[ComplexExpand[FastFourierfit[riggedf, 2 L, 3, t]]]
```

$1.1547\ \mathrm{Sin}\left[\frac{\pi t}{3}\right] - 1.1547\ \mathrm{Sin}\left[\frac{2 \pi t}{3}\right]$

```
Chop[ComplexExpand[FastFourierfit[riggedf, 2 L, 5, t]]]
```

$1.08102\ \mathrm{Sin}\left[\frac{\pi t}{3}\right] - 0.874567\ \mathrm{Sin}\left[\frac{2 \pi t}{3}\right] + 0.266701\ \mathrm{Sin}[\pi t] + 0.255195\ \mathrm{Sin}\left[\frac{4 \pi t}{3}\right]$

```
Chop[ComplexExpand[FastFourierfit[riggedf, 2 L, 10, t]]]
```

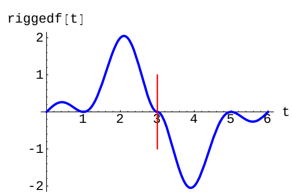
$1.07466\ \mathrm{Sin}\left[\frac{\pi t}{3}\right] - 0.860184\ \mathrm{Sin}\left[\frac{2 \pi t}{3}\right] + 0.239901\ \mathrm{Sin}[\pi t] + 0.305288\ \mathrm{Sin}\left[\frac{4 \pi t}{3}\right] - 0.105195\ \mathrm{Sin}\left[\frac{5 \pi t}{3}\right] + 0.0500932\ \mathrm{Sin}[2 \pi t] - 0.0267998\ \mathrm{Sin}\left[\frac{7 \pi t}{3}\right] + 0.0143832\ \mathrm{Sin}\left[\frac{8 \pi t}{3}\right] - 0.00636405\ \mathrm{Sin}[3 \pi t]$

The fast Fourier fits of riggedf[t] on [0,2 L] are pure Sine fits!
Why did this happen?

□Answer:

Stay with the same L and take a gander at the plot of riggedf[t] on [0, 2 L]:

```
riggedfplot =
 Plot[riggedf[t], {t, 0, 2 L}, PlotStyle → {{Thickness[0.01], Blue}},
  PlotRange → All, AxesLabel → {"t", "riggedf[t]"},
  Epilog → {Red, Line[{{2 L/2, -1}, {2 L/2, 1}}]}];
```



To the right of the center line, the plot of riggedf[t] is the negative mirror image of its plot to the left of the center line.
This is why you can count on the fast Fourier fit of riggedf[t] on [0,2 L] to come out as a pure Sine fit.
The purpose of rigging f[t] in this way is to guarantee the pure sine fit on [0, 2 L]
This will work for any L with L > 0 and any function f[t] with f[0] = f[L] = 0:
Try it:

□Function 1:

```
Clear[f, t]
L = 4;
f[t_] = 6 t (4 - t) E^-t;
{f[0], f[L]}
```

{0, 0}

```
Clear[riggedf, t]
riggedf[t_] := f[t] /; 0 ≤ t ≤ L;
riggedf[t_] := -f[2 L - t] /; L < t ≤ 2 L;
Chop[ComplexExpand[FastFourierfit[riggedf, 2 L, 3, t]]]
```

$4.10249\ \mathrm{Sin}\left[\frac{\pi t}{4}\right] + 2.39086\ \mathrm{Sin}\left[\frac{\pi t}{2}\right]$

□Function 2:

```
Clear[f, t]
L = 1;
f[t_] = t^2 Sin[π t] Log[1 + t];
{f[0], f[L]}
```

{0, 0}

```
Clear[riggedf, t]
riggedf[t_] := f[t] /; 0 ≤ t ≤ L;
riggedf[t_] := -f[2 L - t] /; L < t ≤ 2 L;
Chop[ComplexExpand[FastFourierfit[riggedf, 2 L, 8, t]]]
```

$0.134146\ \mathrm{Sin}[\pi t] - 0.109275\ \mathrm{Sin}[2 \pi t] + 0.0377306\ \mathrm{Sin}[3 \pi t] - 0.0124079\ \mathrm{Sin}[4 \pi t] + 0.00623112\ \mathrm{Sin}[5 \pi t] - 0.00291276\ \mathrm{Sin}[6 \pi t] + 0.00128041\ \mathrm{Sin}[7 \pi t]$

```
Clear[f, t]
L = 3;
f[t_] = (E^-0.3 t - 1) (t - 3) Cos[t];
{f[0], f[L]}
```

{0, 0}

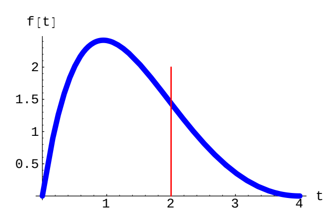
```
Clear[riggedf, t]
riggedf[t_] := f[t] /; 0 ≤ t ≤ L;
riggedf[t_] := -f[2 L - t] /; L < t ≤ 2 L;
Chop[ComplexExpand[FastFourierfit[riggedf, 2 L, 7, t]]]
```

$0.055216\ \mathrm{Sin}\left[\frac{\pi t}{3}\right] + 0.284544\ \mathrm{Sin}\left[\frac{2 \pi t}{3}\right] + 0.0187675\ \mathrm{Sin}[\pi t] + 0.0149468\ \mathrm{Sin}\left[\frac{4 \pi t}{3}\right] + 0.00236841\ \mathrm{Sin}\left[\frac{5 \pi t}{3}\right] + 0.00239598\ \mathrm{Sin}[2 \pi t]$

□B.1.b.ii)

Here is the plot of a function f[t] on an interval [0,L]:

```
Clear[f, t]
f[t_] = 0.4 t (t - 4)^2 E^-0.4 t;
L = 4;
fplot = Plot[f[t], {t, 0, L}, PlotStyle → {{Thickness[0.02], Blue}},
  AxesLabel → {"t", "f[t]"}, DisplayFunction → Identity];
centerline = Graphics[{Red, Line[{{L/2, 0}, {L/2, 2}}]}];
Show[fplot, centerline, DisplayFunction → $DisplayFunction];
```



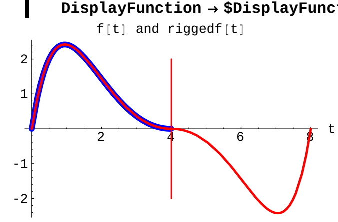
Good, f[0] = f[L] = 0:

```
{f[0], f[L]}
```

{0, 0}

Here's the plot of riggedf[t] on [0,2 L] shown together with the plot of f[t] on [0,L]:

```
Clear[riggedf, t]
riggedf[t_] := f[t] /; 0 ≤ t ≤ L;
riggedf[t_] := -f[2 L - t] /; L < t ≤ 2 L;
riggedfplot = Plot[riggedf[t], {t, 0, 2 L},
  PlotStyle → {{Thickness[0.008], Red}}, PlotRange → All,
  AxesLabel → {"t", "riggedf[t]"}, DisplayFunction → Identity];
newcenterline = Graphics[{Red, Line[{{2 L/2, -2}, {2 L/2, 2}}]}];
all = Show[fplot, riggedfplot, newcenterline,
  AxesLabel → {"t", ""}, PlotLabel → "f[t] and riggedf[t]",
  DisplayFunction → $DisplayFunction];
```

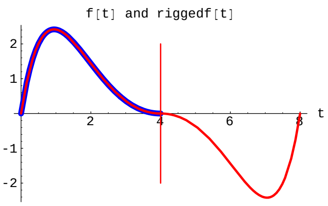


How does this plot signal a way of rigging a pure sine fit of f[t] on [0,L]

□Answer

Take another look:

```
Show[all];
```



The plot tells you that:

-> f[t] and riggedf[t] run together for 0 ≤ t ≤ L and

-> riggedf[t] has a pure Sine fit on [0,2 L].

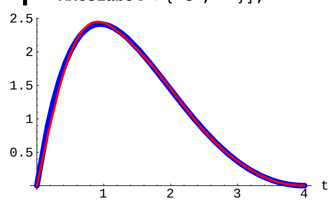
To rig a pure sine fit of f[t] on [0,L], all you gotta do is to use the pure sine fit of riggedf[t] on [0,2 L] to fit f[t] on [0,L];

Go for it:

```
Clear[riggedsinefit]
riggedsinefit[t_] =
 Chop[ComplexExpand[FastFourierfit[riggedf, 2 L, 7, t]]]
```

$$1.71585\,\mathrm{Sin}\left[\frac{\pi t}{4}\right] + 1.06788\,\mathrm{Sin}\left[\frac{\pi t}{2}\right] + 0.335371\,\mathrm{Sin}\left[\frac{3\pi t}{4}\right] + 0.167432\,\mathrm{Sin}[\pi t] + 0.0694637\,\mathrm{Sin}\left[\frac{5\pi t}{4}\right] + 0.0322314\,\mathrm{Sin}\left[\frac{3\pi t}{2}\right]$$

```
fitplot = Plot[{f[t], riggedsinefit[t]}, {t, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
  AxesLabel → {"t", ""}];
```



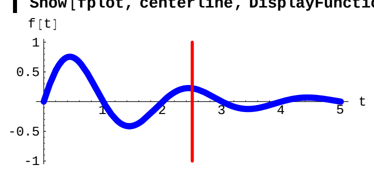
Just as expected, a beautiful Sine fit of f[t] on [0,L].

Math happens again.

□B.1.b.iii)

Here's an interval [0,L] and a function f[t] resistant to a direct Sine on [0,L]:

```
Clear[f, t]
f[t_] = E^(-0.6 t) Sin[π t];
L = 5;
fplot = Plot[f[t], {t, 0, L}, PlotStyle → {{Thickness[0.02], Blue}},
  AxesLabel → {"t", "f[t]"}, DisplayFunction → Identity];
centerline =
 Graphics[{Red, Thickness[0.01], Line[{{L/2, -1}, {L/2, 1}}]}];
Show[fplot, centerline, DisplayFunction → $DisplayFunction];
```



Notice that f[0] = f[L] = 0.
In spite of the lack of symmetry, rig a pure sine fit of f[t] on [0,L].

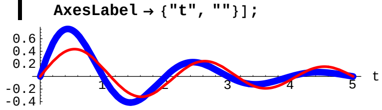
□Answer:

Just do it:

```
Clear[riggedf, t]
riggedf[t_] := f[t] /; 0 ≤ t ≤ L;
riggedf[t_] := -f[2 L - t] /; L < t ≤ 2 L;
Clear[riggedsinefit]
riggedsinefit[t_] =
 Chop[ComplexExpand[FastFourierfit[riggedf, 2 L, 6, t]]]
```

$$0.00595768\,\mathrm{Sin}\left[\frac{\pi t}{5}\right] + 0.0175612\,\mathrm{Sin}\left[\frac{2\pi t}{5}\right] + 0.0403926\,\mathrm{Sin}\left[\frac{3\pi t}{5}\right] + 0.133833\,\mathrm{Sin}\left[\frac{4\pi t}{5}\right] + 0.257565\,\mathrm{Sin}[\pi t]$$

```
fitplot = Plot[{f[t], riggedsinefit[t]}, {t, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
  AxesLabel → {"t", ""}];
```



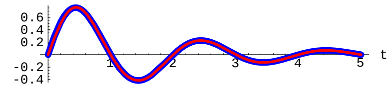
Not bad., but not great.

You can increase the quality of the fit:

```
Clear[riggedsinefit]
riggedsinefit[t_] =
 Chop[ComplexExpand[FastFourierfit[riggedf, 2 L, 13, t]]]
```

$$0.00916125\,\mathrm{Sin}\left[\frac{\pi t}{5}\right] + 0.0255927\,\mathrm{Sin}\left[\frac{2\pi t}{5}\right] + 0.0540633\,\mathrm{Sin}\left[\frac{3\pi t}{5}\right] + 0.162563\,\mathrm{Sin}\left[\frac{4\pi t}{5}\right] + 0.313349\,\mathrm{Sin}[\pi t] + 0.163541\,\mathrm{Sin}\left[\frac{6\pi t}{5}\right] + 0.0559637\,\mathrm{Sin}\left[\frac{7\pi t}{5}\right] + 0.0291482\,\mathrm{Sin}\left[\frac{8\pi t}{5}\right] + 0.0142914\,\mathrm{Sin}\left[\frac{9\pi t}{5}\right] + 0.00906991\,\mathrm{Sin}[2\pi t] + 0.0045965\,\mathrm{Sin}\left[\frac{11\pi t}{5}\right] + 0.00229929\,\mathrm{Sin}\left[\frac{12\pi t}{5}\right]$$

```
Plot[{f[t], riggedsinefit[t]}, {t, 0, L},
 PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
 AxesLabel → {"t", ""}];
```



It doesn't get much better than this.

□B.1.b.iv)

Why is it a good idea to check whether
f[0] = f[L] = 0
before you go for a rigged Sine fit of f[t] on [0,L]?

□Answer:

The rigged sine fit of f[t] on [0,L] comes from from a fast Fourier fit of riggedf[t] on [0,2L]. This means you are fitting with complex exponentials

$$E^{\frac{I 2 k \pi t}{2L}} = E^{\frac{I k \pi t}{L}}$$

Consequently the sines involved in the rigged Sine fit are

$$\mathrm{Sin}\left[\frac{k \pi t}{L}\right].$$

These functions are all zeroed out at t = 0 and t = L.

This forces all the rigged sine fitters of f[t] on [0,L] to be zeroed out at t = 0 and at t = L.

The upshot:

If you expect a good rigged sine fit of f[t] on [0,L], you'll want
f[0] = f[L] = 0.

B.2) Fast Fourier Sine fit and the heat equation

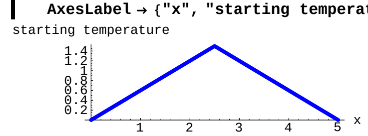
$$\partial_{\{x,2\}}\,\mathrm{temp}[x, t] = \partial_t\,\mathrm{temp}[x, t]$$

Fourier invented Fourier fit for
the purpose of working on this very problem.

□B.2.a.i)

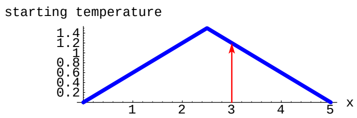
Start with a heated wire L units long with the temperature allowed to vary from position to position on the wire. Think of the wire as the interval 0 ≤ x ≤ L. At the start of the experiment, you instantly cool the ends at x = 0 and x = L and maintain these ends at temperature 0, and you take pains to guarantee that the rest of the wire is perfectly insulated.
At the start of this particular experiment, the temperature of the wire at position x (for 0 ≤ x ≤ L = 5) is given by the following function startertemp[x]:

```
L = 5;
Clear[startertemp, x]
startertemp[x_] = 3 Abs[0.2 x - Round[0.2 x]];
starterplot = Plot[startertemp[x],
  {x, 0, L}, PlotStyle → {{Thickness[0.015], Blue}},
  AxesLabel → {"x", "starting temperature"}];
```



To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] :=
 Arrow[{0, startertemp[x]}, Tail → {x, 0}, VectorColor → Red];
Show[starterplot, pointer[3]];
```



Think of the interval [0, L] = [0, 5] as the wire.
The tip of the pointer tells you the starting temperature (at time t = 0) at the tail of the pointer.
Your problem here is to come up with a function temp[x, t] that estimates the temperature of the wire at position x at time t after the experiment begins.
Do it.

Answer:

The function startertemp[x] is ripe for a rigged Fourier Sine fit on [0, L] because

startertemp[x] = 0 for x = 0 and x = L:

```
{startertemp[0], startertemp[L]}
```

{0, 0.}

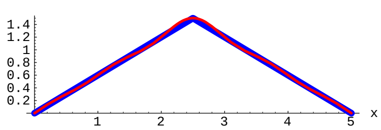
Rig startertemp[x] for a pure sine fit on [0, L] and get a good sine fit of startertemp[x] on [0, L]:

If none of this makes sense to you, then look at B.1) immediately above.

```
Clear[rigged]
rigged[x_] := startertemp[x] /; 0 ≤ x ≤ L;
rigged[x_] := -startertemp[2 L - x] /; L < x ≤ 2 L;
n = 10;
Clear[riggedsinefit]
riggedsinefit[x_] =
 Chop[ComplexExpand[FastFourierfit[rigged, 2 L, n, x]]]
```

$$1.2259\,\mathrm{Sin}\left[\frac{\pi x}{5}\right]-0.145555\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right]+0.06\,\mathrm{Sin}[\pi x]-0.0377885\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right]+0.0307526\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right]$$

```
fitplot = Plot[{startertemp[x], riggedsinefit[x]}, {x, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
  AxesLabel → {"x", ""}];
```



That's a decent fit.

Now look at the rigged sine fit of startertemp[x]:

```
riggedsinefit[x]
```

$$1.2259\,\mathrm{Sin}\left[\frac{\pi x}{5}\right]-0.145555\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right]+0.06\,\mathrm{Sin}[\pi x]-0.0377885\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right]+0.0307526\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right]$$

Pick off the coefficients of the Sin[$\frac{(k\pi)x}{L}$] terms:

```
Clear[A, k]
A[k_] := Coefficient[riggedsinefit[x], Sin[(k Pi / L) x]];

Table[A[k], {k, 1, n}]
```

{1.2259, 0, -0.145555, 0, 0.06, 0, -0.0377885, 0, 0.0307526, 0}

The reason you run k from 1 to n is to pick up all the coefficients.

Now you're done because you can write down temp[x, t].

It's just:

```
Clear[temp, x, t]
temp[x_, t_] = Sum[A[k] E^(-((k π)/L)^2 t) Sin[((k π) x)/L], {k, 1, n}]
```

$$1.2259\,E^{-\frac{\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{\pi x}{5}\right]-0.145555\,E^{-\frac{9\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right]+0.06\,E^{-\pi^2 t}\,\mathrm{Sin}[\pi x]-0.0377885\,E^{-\frac{49\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right]+0.0307526\,E^{-\frac{81\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right]$$

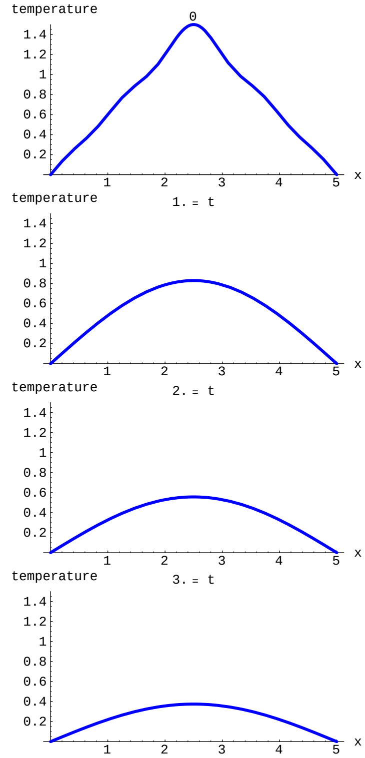
Quite a slap in the face, but that's all there is to it.

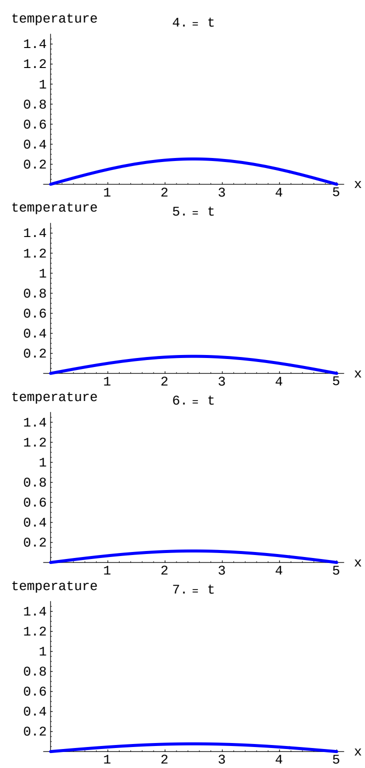
If you want to understand why this works, then continue with the rest of this Basic problem.

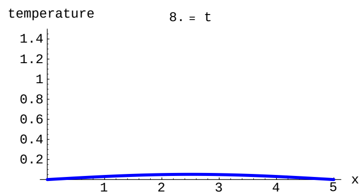
B.2.a.ii) A Movie

Stay with the same situation as in part ii) immediately above and look at this movie:

```
Clear[tempplotter]
tempplotter[t_] :=
 Plot[temp[x, t], {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  PlotRange → {0, 1.5}, AxesLabel → {"x", " temperature"},
  PlotLabel → N[t] "= t", AspectRatio → 1/2];
timejump = 1;
Table[tempplotter[t], {t, 0, 8, timejump}];
```







Grab all the plots and animate.

What does this movie depict?

Answer:

It depicts the the wire cooling down as time goes on.

This comes from the fact that the temperature at the ends of the wire are maintained at 0 throughout the experiment.

Another way of seeing what's happening is to take another look at the formula for temp[x, t]:

```
temp[x, t]
```

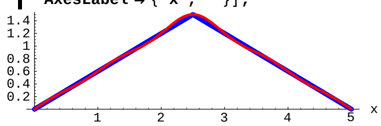
$$1.2259\,E^{-\frac{\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.145555\,E^{-\frac{9\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.06\,E^{-\pi^2 t}\,\mathrm{Sin}[\pi x] - 0.0377885\,E^{-\frac{49\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right] + 0.0307526\,E^{-\frac{81\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right]$$

Those exponentials in the denominators are driving temp[x, t] to 0 as t gets large.

B.2.a.iii)

Look at this:

```
Plot[{startertemp[x], temp[x, 0]}, {x, 0, L},
 PlotStyle → {{Thickness[0.015], Blue}, {Thickness[0.008], Red}},
 AxesLabel → {"x", ""}];
```



This is a plot of startertemp[x] and temp[x, 0] for $0 \le x \le L$.
Why is it totally natural to expect the this plot to come out they way it did?

Physical Answer:

To begin with,

temp[x, t]

estimates the temperature of the wire at position x at t time units after the experiment starts.

But

startertemp[x]

measures the the temperature of the wire at position x at the start (at time t = 0). So it is natural that

temp[x, 0] should estimate starter[x]

pretty well.

Math Answer:

Look at:

```
temp[x, t]
```

$$1.2259\,E^{-\frac{\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.145555\,E^{-\frac{9\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.06\,E^{-\pi^2 t}\,\mathrm{Sin}[\pi x] - 0.0377885\,E^{-\frac{49\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right] + 0.0307526\,E^{-\frac{81\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right]$$

When you put t = 0, all the exponentials become 1:

```
temp[x, 0]
```

$$1.2259\,\mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.145555\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.06\,\mathrm{Sin}[\pi x] - 0.0377885\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right] + 0.0307526\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right]$$

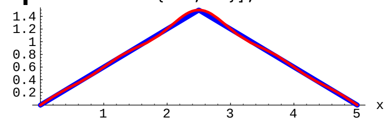
This is the same as the rigged sine fit of starter[x]:

```
riggedsinefit[x]
```

$$1.2259\,\mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.145555\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.06\,\mathrm{Sin}[\pi x] - 0.0377885\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right] + 0.0307526\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right]$$

This is why the plots of temp[x, 0] and startertemp[x] share a lot of ink:

```
Plot[{startertemp[x], temp[x, 0]}, {x, 0, L},
 PlotStyle → {{Thickness[0.015], Blue}, {Thickness[0.008], Red}},
 AxesLabel → {"x", ""}];
```



B.2.a.iv) The heat equation explains where the exponentials come from

In the parts above, you get temp[x, t] by taking a rigged sine fit of startertemp[x]:

```
riggedsinefit[x]
```

$$1.2259\,\mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.145555\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.06\,\mathrm{Sin}[\pi x] - 0.0377885\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right] + 0.0307526\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right]$$

And you get temp[x, t] by inserting exponentials into each term of the rigged sine fit:

```
temp[x, t]
```

$$1.2259\,E^{-\frac{\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.145555\,E^{-\frac{9\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.06\,E^{-\pi^2 t}\,\mathrm{Sin}[\pi x] - 0.0377885\,E^{-\frac{49\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right] + 0.0307526\,E^{-\frac{81\pi^2 t}{25}}\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right]$$

Why do you insert those exponentials?

Answer

This is the central question.

Engineering studies have shown that after the appropriate unit adjustments are made, the function temp[x,t] satisfies the partial differential equation known as the heat equation

$$\partial_{\{x,2\}}\,\mathrm{temp}[x, t] = \partial_t\,\mathrm{temp}[x, t]$$

This is the same as D[temp[x,t], {x,2}] = D[temp[x,t], t];
in other words

the second derivative of temp[x,t[with respect to x equals the first derivative of temp[x,t] with respect to t.

with

→ temp[x, 0] = startertemp[x]

and

→ temp[0, t] = 0 and temp[L, t] = 0 for all t's.

Lots of folks call this a by the name "Boundary Value Problem."

The key is the boundary conditions

temp[0, t] = 0 and temp[L, t] = 0 for all t's.

These match up well with the fact that

$\mathrm{Sin}[\frac{k\pi x}{L}] = 0$ for x = 0 and x = L

for all positive integers k. This suggests that for each fixed time t, you can fit temp[x,t] with a rigged Sine fit like this:

```
Clear[approxtemp, t, x, u, L]
n = 6;
approxtemp[x_, t_] = Sum[u[t, k] Sin[(k π) x / L], {k, 1, n}]
```

$$\mathrm{Sin}\left[\frac{\pi x}{L}\right] u[t, 1] + \mathrm{Sin}\left[\frac{2\pi x}{L}\right] u[t, 2] + \mathrm{Sin}\left[\frac{3\pi x}{L}\right] u[t, 3] + \mathrm{Sin}\left[\frac{4\pi x}{L}\right] u[t, 4] + \mathrm{Sin}\left[\frac{5\pi x}{L}\right] u[t, 5] + \mathrm{Sin}\left[\frac{6\pi x}{L}\right] u[t, 6]$$

There is nothing magic taking n = 6.
You can go with a bigger n or a smaller n.

The Fourier fit coefficients u[t, k] depend on t as well as k because you expect a different rigged sine fit at different times t.

The heat equation says

D[temp[x,t],{x,2}] = D[temp[x,t],t].

Plug

$$\text{approxtemp}[x, t] = \sum_{k=1}^{n} u[t, k]\, \text{Sin}[\frac{(k\pi)x}{L}]$$

into the heat equation and see that

$$\sum_{k=1}^{n} u[t, k]\,(\frac{k\pi}{L})^2\,(-\text{Sin}[\frac{(k\pi)x}{L}]) = \sum_{k=1}^{6} \partial_t u[t, k]\, \text{Sin}[\frac{(k\pi)x}{L}].$$

You can make this happen by setting

$$D[u[t, k], t] = -(\frac{k\pi}{L})^2\, u[t, k]\ .$$

This is a big break in your favor because this is the exponential differential equation and you know that this makes

$$u[t, k] = A[k]\, E^{-(\frac{k\pi}{L})^2 t}$$

where the real constant A[k] has yet to be determined.

You get a different constant for each k.

Substitute these u[t,k]'s into approxtemp[x,t] to get

$$\text{approxtemp}[x, t] = \sum_{k=1}^{n} A[k]\, E^{-(\frac{k\pi}{L})^2 t}\, \text{Sin}[\frac{(k\pi)x}{L}]$$

Look at what happens for t = 0:

$$\text{approxtemp}[x, 0] = \sum_{k=1}^{n} A[k]\ \text{Sin}[\frac{(k\pi)x}{L}]$$

This is to fit the starting temperature, so you lift the A[k]'s from the rigged sine fit of startertemp[x].

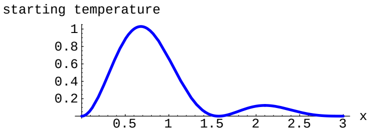
And now you see why you want to insert the exponentials.

□B.2.b) A new one

Start with a heated wire L units long with the temperature allowed to vary from position to position on the wire. Think of the wire as the interval 0 ≤ x ≤ L. At the start of the experiment, you instantly cool the ends at x = 0 and x = L and maintain these ends at temperature 0, and you take pains to guarantee that the rest of the wire is perfectly insulated.

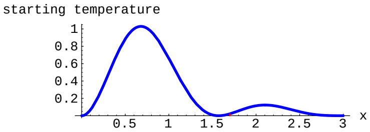
At the start of this particular experiment, the temperature of the wire at position x (for 0 ≤ x ≤ L = 3) is given by the following function startertemp[x]:

```
L = 3;
Clear[startertemp, x]
startertemp[x_] = 0.2 Sin[2 x]^2 (x - 3)^2;
starterplot = Plot[startertemp[x],
  {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  AxesLabel → {"x", "starting temperature"}];
```



To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] :=
 Arrow[{0, startertemp[x]}, Tail → {x, 0}, VectorColor → Red];
Show[starterplot, pointer[1.7]];
```



Think of the interval

[0,L] = [0,3]

as the wire.

The tip of the pointer tells you the starting temperature (at time t = 0) at the tail of the pointer.

You now know how and understand how come up with a function temp[x,t] that estimates the temperature of the wire at position x at time t after the experiment begins.

Do it.

Make a good movie.

□Answer:

This is a copy, paste and edit job on B.2.a.i and a.ii)

The function

startertemp[x]

is ripe for a rigged Fourier Sine fit on [0, L]because

startertemp[x] = 0 for x = 0 and x = L:

```
{startertemp[0], startertemp[L]}
{0, 0}
```

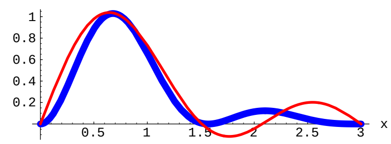
Rig startertemp[x] for a pure sine fit on [0,L] and get a good sine fit of startertemp[t] on [0,L]:

If none of this makes sense to you, then look at B.1) immediately above.

```
Clear[rigged, x, t]
rigged[x_] := startertemp[x] /; 0 ≤ x ≤ L;
rigged[x_] := -startertemp[2 L - x] /; L < x ≤ 2 L;
Clear[riggedsinefit]
n = 4;
riggedsinefit[x_] =
 Chop[ComplexExpand[FastFourierfit[rigged, 2 L, n, x]]]
```

$$0.39867\, \text{Sin}[\frac{\pi x}{3}] + 0.449966\, \text{Sin}[\frac{2\pi x}{3}] + 0.389708\, \text{Sin}[\pi x]$$

```
fitplot = Plot[{startertemp[x], riggedsinefit[x]}, {x, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
  AxesLabel → {"x", ""}];
```

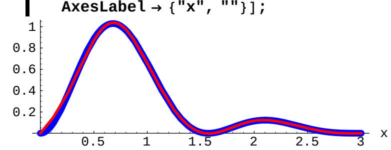


That's not a great fit. To improve the fit, raise n:

```
n = 10;
Clear[riggedsinefit]
riggedsinefit[x_] =
 Chop[ComplexExpand[FastFourierfit[rigged, 2 L, n, x]]]
```

$$0.379448\, \text{Sin}[\frac{\pi x}{3}] + 0.393967\, \text{Sin}[\frac{2\pi x}{3}] + 0.297019\, \text{Sin}[\pi x] + 0.0662433\, \text{Sin}[\frac{4\pi x}{3}] - 0.0949962\, \text{Sin}[\frac{5\pi x}{3}] - 0.0612287\, \text{Sin}[2\pi x] - 0.0280943\, \text{Sin}[\frac{7\pi x}{3}] - 0.0158073\, \text{Sin}[\frac{8\pi x}{3}] - 0.00656556\, \text{Sin}[3\pi x]$$

```
fitplot = Plot[{startertemp[x], riggedsinefit[x]}, {x, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
  AxesLabel → {"x", ""}];
```



That's a decent fit.

Now look at the rigged sine fit of startertep[x]:

```
riggedsinefit[x]
```

$$0.379448\, \text{Sin}[\frac{\pi x}{3}] + 0.393967\, \text{Sin}[\frac{2\pi x}{3}] + 0.297019\, \text{Sin}[\pi x] + 0.0662433\, \text{Sin}[\frac{4\pi x}{3}] - 0.0949962\, \text{Sin}[\frac{5\pi x}{3}] - 0.0612287\, \text{Sin}[2\pi x] - 0.0280943\, \text{Sin}[\frac{7\pi x}{3}] - 0.0158073\, \text{Sin}[\frac{8\pi x}{3}] - 0.00656556\, \text{Sin}[3\pi x]$$

Pick off the coefficients of the Sin[(k Pi/L) x] terms:

```
Clear[A, k]
A[k_] := Coefficient[riggedsinefit[x], Sin[(k π) x / L]];
coeffs = Table[A[k], {k, 1, n}]
{0.379448, 0.393967, 0.297019, 0.0662433, -0.0949962, -0.0612287,
 -0.0280943, -0.0158073, -0.00656556, 0}
```

The reason you run k from 1 to n is to pick up all the coefficients.

Now you're done because you can write down temp[x,t]:

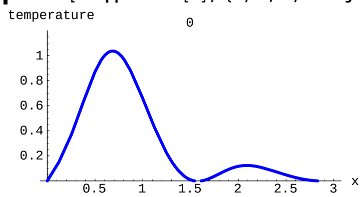
```
Clear[temp, x, t]
```

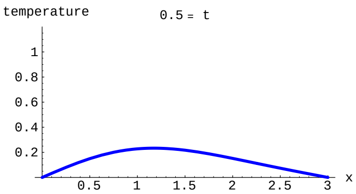
$$\text{temp}[x\_, t\_] = \sum_{k=1}^{\text{Length}[\text{coeffs}]} A[k]\, E^{-(\frac{k\pi}{L})^2 t}\, \text{Sin}[\frac{(k\pi)x}{L}]$$

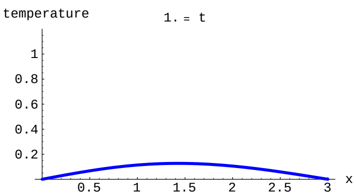
$$0.379448\, E^{-\frac{\pi^2 t}{9}} \text{Sin}\left[\frac{\pi x}{3}\right] + 0.393967\, E^{-\frac{4\pi^2 t}{9}} \text{Sin}\left[\frac{2\pi x}{3}\right] + 0.297019\, E^{-\pi^2 t} \text{Sin}[\pi x] + 0.0662433\, E^{-\frac{16\pi^2 t}{9}} \text{Sin}\left[\frac{4\pi x}{3}\right] - 0.0949962\, E^{-\frac{25\pi^2 t}{9}} \text{Sin}\left[\frac{5\pi x}{3}\right] - 0.0612287\, E^{-4\pi^2 t} \text{Sin}[2\pi x] - 0.0280943\, E^{-\frac{49\pi^2 t}{9}} \text{Sin}\left[\frac{7\pi x}{3}\right] - 0.0158073\, E^{-\frac{64\pi^2 t}{9}} \text{Sin}\left[\frac{8\pi x}{3}\right] - 0.00656556\, E^{-9\pi^2 t} \text{Sin}[3\pi x]$$

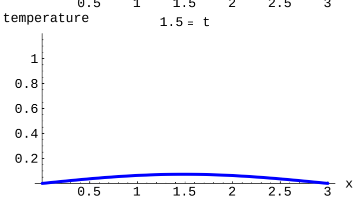
Here's the movie:

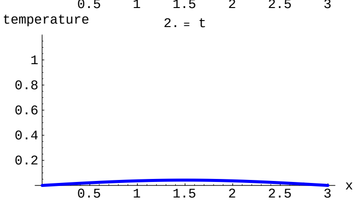
```
Clear[tempplotter]
tempplotter[t_] :=
 Plot[temp[x, t], {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  PlotRange → {0, 1.2}, AxesLabel → {"x", " temperature"},
  PlotLabel → N[t] "= t", AspectRatio → 1/2];
timejump = 0.5;
Table[tempplotter[t], {t, 0, 3, timejump}];
```

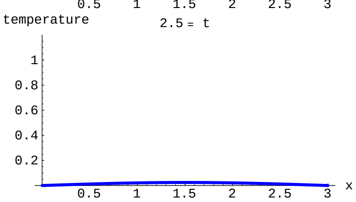


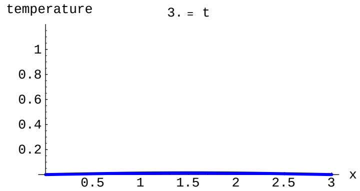












Grab all the plots and animate.

After 3 time units, the temperature of the wire is approximately 0 from end to end.

B.3) Fast Fourier Sine fit and the wave equation

$$\partial_{\{x,2\}} \text{position}[x, t] = \partial_{\{t,2\}} \text{position}[x, t]$$

□B.3.a.i)

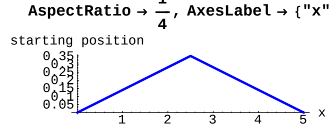
Run this code before you start:

```
Clear[FastFourierfit, Fourierfitters, F, Fvalues, n, k,
   jump, num, numtab, coeffs, t, L]
jump[n_] := jump[n] = N[1 / (2 n)];
Fvalues[F_, L_, n_] :=
 N[Table[F[L t], {t, 0, 1 - jump[n], jump[n]}]];


numtab[n_] := numtab[n] = Table[k, {k, 1, n}];

Fourierfitters[L_, n_, t_] := Table[E^(2 Pi I k t / L),
    {k, -n + 1, n - 1}];
coeffs[n_, list_] := Join[Reverse[Part[Fourier[list], numtab[n]]],
 Part[InverseFourier[list], Drop[numtab[n], 1]]] /
  N[Sqrt[Length[list]]]

FastFourierfit[F_, L_, n_, t_] :=
  Chop[Fourierfitters[L, n, t] . coeffs[n, Fvalues[F, L, n]]];
L := Expand[a D[#1, {#2, 2}] + b D[#1, #2] + c #1]&
```

The ends of a guitar string are anchored at 0 and L on the x-axis and the string is pulled to an initial position and then allowed to vibrate on its own starting with initial velocity 0.

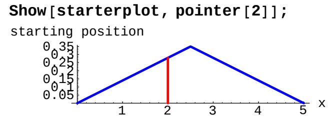
At the start of this particular experiment, the position of the wire at position x (for 0 ≤ x ≤ L = 5) is given by the following function starterposition[x]:

```
L = 5;
Clear[starterposition, x]
starterposition[x_] = 0.7 Abs[0.2 x - Round[0.2 x]];
starterplot = Plot[starterposition[x], {x, 0, L},
  PlotStyle → {{Thickness[0.01], Blue}}, PlotRange → All,
  AspectRatio → 1/4, AxesLabel → {"x", "starting position"}];
```



Think of the curve as the the starting position of the guitar string.
To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] := Graphics[
  {Red, Thickness[0.01], Line[{{x, starterposition[x]}, {x, 0}}]}];
Show[starterplot, pointer[2]];
```



The tip of the pointer tells you the starting position (at time t = 0) at the tail of the pointer.
Your problem here is to come up with a function position[x,t] that estimates the position of the guitar string at position x on the x-axis at time t after the experiment begins.
Do it.

□Answer:

```
R
```

R

The function starterposition[x] is ripe for a rigged Fourier Sine fit on [0,L] because starterposition[x] = 0 for x = 0 and x = L:

```
{starterposition[0], starterposition[L]}
```

{0, 0.}

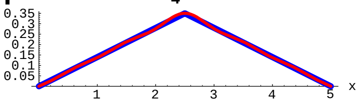
Rig starterposition[x] for a pure sine fit on [0,L] and get a good sine fit of starterposition[x] on [0,L]:

If none of this makes sense to you, then look at B.1) immediately above.

```
Clear[rigged]
rigged[x_] := starterposition[x] /; 0 ≤ x ≤ L;
rigged[x_] := -starterposition[2 L - x] /; L < x ≤ 2 L;
Clear[riggedsinefit]
n = 12;
riggedsinefit[x_] =
 Chop[ComplexExpand[FastFourierfit[rigged, 2 L, n, x]]]
```

$$0.285325\,\mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.0331937\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.0131172\,\mathrm{Sin}[\pi x] - 0.00772329\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right] + 0.00569515\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right] - 0.00494537\,\mathrm{Sin}\left[\frac{11\pi x}{5}\right]$$

```
fitplot = Plot[{starterposition[x], riggedsinefit[x]}, {x, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.01], Red}},
  AspectRatio → 1/4, AxesLabel → {"x", ""}];
```



That's a great fit.

Now look at the rigged sine fit of starterposition[x]:

```
riggedsinefit[x]
```

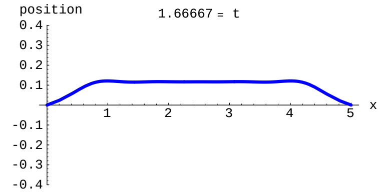
$$0.285325\,\mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.0331937\,\mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.0131172\,\mathrm{Sin}[\pi x] - 0.00772329\,\mathrm{Sin}\left[\frac{7\pi x}{5}\right] + 0.00569515\,\mathrm{Sin}\left[\frac{9\pi x}{5}\right] - 0.00494537\,\mathrm{Sin}\left[\frac{11\pi x}{5}\right]$$

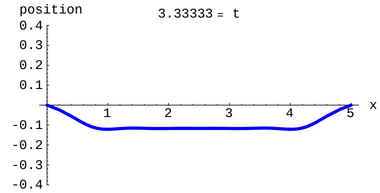
Pick off the coefficients of the Sin[(k Pi/L) x] terms:

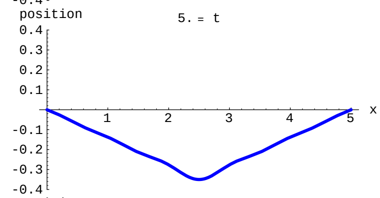
```
Clear[A, k]
A[k_] := Coefficient[riggedsinefit[x], Sin[(k π) x / L]];
coeffs = Table[A[k], {k, 1, n}]
```

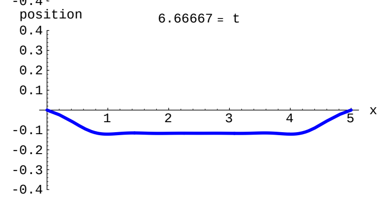
{0.285325, 0, -0.0331937, 0, 0.0131172, 0, -0.00772329, 0, 0.00569515, 0, -0.00494537, 0}

The reason you run k from 1 to n is to pick up all the coefficients.









Now you're done because you can write down approxposition[x,t]:

```
Clear[approxposition, x, t]
approxposition[x_, t_] = Sum[A[k] Cos[(k π) t / L] Sin[(k π) x / L], {k, 1, Length[coeffs]}]
```

$$0.285325\,\mathrm{Cos}\left[\frac{\pi t}{5}\right]\mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.0331937\,\mathrm{Cos}\left[\frac{3\pi t}{5}\right]\mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.0131172\,\mathrm{Cos}[\pi t]\,\mathrm{Sin}[\pi x] - 0.00772329\,\mathrm{Cos}\left[\frac{7\pi t}{5}\right]\mathrm{Sin}\left[\frac{7\pi x}{5}\right] + 0.00569515\,\mathrm{Cos}\left[\frac{9\pi t}{5}\right]\mathrm{Sin}\left[\frac{9\pi x}{5}\right] - 0.00494537\,\mathrm{Cos}\left[\frac{11\pi t}{5}\right]\mathrm{Sin}\left[\frac{11\pi x}{5}\right]$$

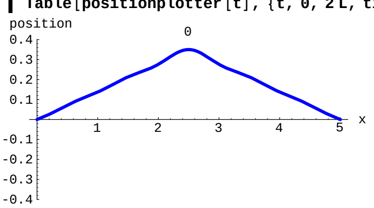
Quite a slap in the face, but that's all there is to it.

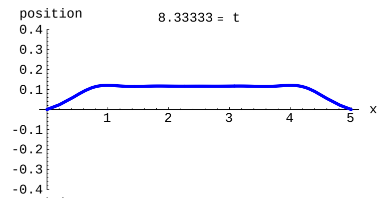
If you want to understand why this works, then continue with the rest of this Basic problem.

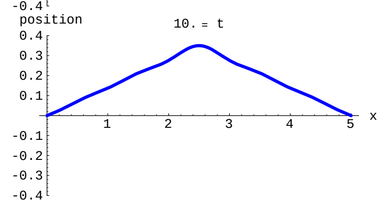
□B.3.a.ii) A Movie

Stay with the same situation as in part a.i) immediately above and look at this movie:

```
Clear[positionplotter]
positionplotter[t_] := Plot[approxposition[x, t],
  {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  PlotRange → {-0.4, 0.4}, AxesLabel → {"x", " position"},
  PlotLabel → N[t] "= t", AspectRatio → 1/2];
timejump = L/3;
Table[positionplotter[t], {t, 0, 2 L, timejump}]
```







{- Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -}

Grab all the plots and animate.

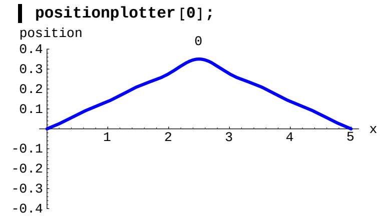
What does this movie depict?

□**Answer:**

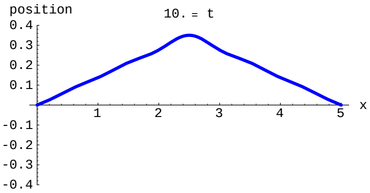
It depicts the vibration of the guitar string.

Notice that the string returns to its starting position after 2 L time units:

```
positionplotter[0];
```




```
positionplotter[2 L];
```



This oscillatory (periodic) behavior as time goes on comes from the Cosines in position[x,t]:

```
approxposition[x, t]
```

$$0.285325\,\text{Cos}[\tfrac{\pi t}{5}]\,\text{Sin}[\tfrac{\pi x}{5}] - 0.0331937\,\text{Cos}[\tfrac{3\pi t}{5}]\,\text{Sin}[\tfrac{3\pi x}{5}] + 0.0131172\,\text{Cos}[\pi t]\,\text{Sin}[\pi x] - 0.00772329\,\text{Cos}[\tfrac{7\pi t}{5}]\,\text{Sin}[\tfrac{7\pi x}{5}] + 0.00569515\,\text{Cos}[\tfrac{9\pi t}{5}]\,\text{Sin}[\tfrac{9\pi x}{5}] - 0.00494537\,\text{Cos}[\tfrac{11\pi t}{5}]\,\text{Sin}[\tfrac{11\pi x}{5}]$$

□B.3.a.iii) The wave equation explains where the Cosines come from

You get position[x,t] by taking a rigged sine fit of starterposition[x]:

```
riggedsinefit[x]
```

$$0.285325\,\text{Sin}[\tfrac{\pi x}{5}] - 0.0331937\,\text{Sin}[\tfrac{3\pi x}{5}] + 0.0131172\,\text{Sin}[\pi x] - 0.00772329\,\text{Sin}[\tfrac{7\pi x}{5}] + 0.00569515\,\text{Sin}[\tfrac{9\pi x}{5}] - 0.00494537\,\text{Sin}[\tfrac{11\pi x}{5}]$$

And you get position[x,t] by inserting Cosines into each term of the rigged sine fit:

```
approxposition[x, t]
```

$$0.285325\,\text{Cos}[\tfrac{\pi t}{5}]\,\text{Sin}[\tfrac{\pi x}{5}] - 0.0331937\,\text{Cos}[\tfrac{3\pi t}{5}]\,\text{Sin}[\tfrac{3\pi x}{5}] + 0.0131172\,\text{Cos}[\pi t]\,\text{Sin}[\pi x] - 0.00772329\,\text{Cos}[\tfrac{7\pi t}{5}]\,\text{Sin}[\tfrac{7\pi x}{5}] + 0.00569515\,\text{Cos}[\tfrac{9\pi t}{5}]\,\text{Sin}[\tfrac{9\pi x}{5}] - 0.00494537\,\text{Cos}[\tfrac{11\pi t}{5}]\,\text{Sin}[\tfrac{11\pi x}{5}]$$

Why do you insert those Cosines?

□Answer

This is the central question.

Engineering studies have shown that after the appropriate unit adjustments are made, the function position[x,t] satisfies the wave equation

$$\partial_{\{x,2\}}\,\text{position}[x, t] = \partial_{\{t,2\}}\,\text{position}[x, t].$$

This is the same as D[position[x,t], {x,2}] = D[position[x,t], {t,2}].

with

-> position[x, 0] = starterposition[x]

-> position[0, t] = 0 and position[L, t] = 0 for all t's

Reason: The ends of the guitar string are attached at the ends.

and with

-> ∂_t position[x, t] = 0 for t = 0 .

Reason: you let the guitar string vibrate giving it initial velocity 0.

Lots of folks call this a partial differential equation.

The key is the boundary conditions

position[0, t] = 0 and position[L, t] = 0.

These match up well with the fact that

$$\text{Sin}[\tfrac{k\pi x}{L}] = 0 \text{ for } x = 0 \text{ and } x = L$$

for all positive integers k. This suggests that for a fixed time t, you can fit position[x,t] with a rigged Sine fit like this:

```
Clear[approxposition, t, x, u, L]
n = 4;
approxposition[x_, t_] = Sum[u[t, k] Sin[((k π) x)/L], {k, 1, n}]
```

$$\text{Sin}[\tfrac{\pi x}{L}]\,u[t, 1] + \text{Sin}[\tfrac{2\pi x}{L}]\,u[t, 2] + \text{Sin}[\tfrac{3\pi x}{L}]\,u[t, 3] + \text{Sin}[\tfrac{4\pi x}{L}]\,u[t, 4]$$

There is nothing magic about setting n = 4. You can use higher or lower n's

The Fourier fit coefficients u[t, k] depend on t as well as k because you expect a different rigged sine fit at different times t.

The wave equation says

$$\partial_{\{x,2\}}\,\text{position}[x, t] = \partial_{\{t,2\}}\,\text{position}[x, t]$$

Plug

$$\text{approxposition}[x, t] = \sum_{k=1}^{n} u[t, k]\,\text{Sin}[\tfrac{(k\pi)\,x}{L}]$$

into both sides of thw wave equation and see that you want:

$$\sum_{k=1}^{n} u[t, k]\,(\tfrac{k\pi}{L})^2\,(-\text{Sin}[\tfrac{(k\pi)\,x}{L}]) = \sum_{k=1}^{n} \partial_{\{t,2\}}\,u[t, k]\,\text{Sin}[\tfrac{(k\pi)\,x}{L}].$$

You can make this happen by setting

$$\partial_{\{t,2\}}\,u[t, k] = -(\tfrac{k\pi}{L})^2\,u[t, k]\ .$$

This is the same as

$$\partial_{\{t,2\}}\,u[t, k] + (\tfrac{k\pi}{L})^2\,u[t, k] = 0$$

This is a big break in your favor because this is the undamped unforced oscillator:

```
Clear[u, t, A, B, k]
u[t_, k_] = A[k] Cos[((k π) t)/L] + B[k] Sin[((k π) t)/L]
```

$$A[k]\,\text{Cos}[\tfrac{k\pi t}{L}] + B[k]\,\text{Sin}[\tfrac{k\pi t}{L}]$$

Here the real constants A[k] and B[k] have yet to be determined.

You get a different constant for each k.

Now look at the condition

∂_t position[x, t] = 0 for t = 0.

Because

$$\text{approxposition}[x, t] = \sum_{k=1}^{n} u[t, k]\,\text{Sin}[\tfrac{(k\pi)\,x}{L}],$$

you can achieve

∂_t position[x, t] = 0 for t = 0

by insisting that:

```
(∂t u[t, k] /. t → 0) == 0
```

$$\frac{k\pi B[k]}{L} == 0$$

This tells you to set B[k] = 0 so that u[t,k] becomes:

```
u[t, k] /. B[k] → 0
```

$$A[k]\,\text{Cos}[\tfrac{k\pi t}{L}]$$

Substitute

$$u[t, k] = A[k]\,\text{Cos}[\tfrac{(k\pi)\,t}{L}]$$

into approxposition[x,t] to get

$$\text{approxposition}[x, t] = \sum_{k=1}^{n} A[k]\,\text{Cos}[\tfrac{(k\pi)\,x}{L}]\,\text{Sin}[\tfrac{(k\pi)\,x}{L}]:$$

Look at what happens for t = 0:

$$\text{approxposition}[x, 0] = \sum_{k=1}^{n} A[k]\,\text{Cos}[\tfrac{(k\pi)\,0}{L}]\,\text{Sin}[\tfrac{(k\pi)\,x}{L}] = \sum_{k=1}^{n} A[k]\,\text{Sin}[\tfrac{(k\pi)\,x}{L}]$$

This is to fit the starting position, so you lift the A[k]'s from the rigged sine fit of starterposition[x].

And now you see why you want to insert the Cosines.

☐B.3.b) A new one

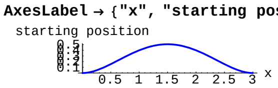
Activate this code:

```
Clear[FastFourierfit, Fourierfitters, F, Fvalues, n, k,
   jump, num, numtab, coeffs, t, L]
jump[n_] := jump[n] = N[1 / (2 n)];
Fvalues[F_, L_, n_] :=
 N[Table[F[L t], {t, 0, 1 - jump[n], jump[n]}]];


numtab[n_] := numtab[n] = Table[k, {k, 1, n}];

Fourierfitters[L_, n_, t_] := Table[E^(2 Pi I k t / L),
     {k, -n + 1, n - 1}];
coeffs[n_, list_] := Join[Reverse[Part[Fourier[list], numtab[n]]],
 Part[InverseFourier[list], Drop[numtab[n], 1]]] /
  N[Sqrt[Length[list]]]

FastFourierfit[F_, L_, n_, t_] :=
  Chop[Fourierfitters[L, n, t] . coeffs[n, Fvalues[F, L, n]]];
L := Expand[a D[#1, {#2, 2}] + b D[#1, #2] + c #1]&
```

The ends of a guitar string are anchored at 0 and L on the x-axis and the string is pulled to an initial position and then allowed to vibrate on its own starting with initial velocity 0.

At the start of this partricular experiment, the position of the wire at position x (for $0 \leq x \leq L = 3$) is given by the following function starterposition[x]:

```
L = 3;
Clear[starterposition, x]
starterposition[x_] = 0.1 x^2 (x - 3)^2;
starterplot = Plot[starterposition[x],
  {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  AxesLabel → {"x", "starting position"}];
```



.Your problem here is to come up with a function position[x, t] that estimates the position of the guitar string at position x on the x-axis at time t after the experiment begins.

Do it.
Make a good movie.

☐**Answer:**

The function starterposition[x] is ripe for a rigged Fourier Sine fit on [0,L] because starterposition[x] = 0 for x = 0 and x = L:

```
{starterposition[0], starterposition[L]}
```

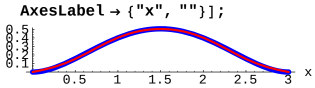
{0, 0}

Rig starterposition[x] for a pure sine fit on [0,L] and get a good sine fit of starterposition[t] on [0,L]:

```
Clear[rigged]
rigged[x_] := starterposition[x] /; 0 ≤ x ≤ L;
rigged[x_] := -starterposition[2 L - x] /; L < x ≤ 2 L;
Clear[riggedsinefit]
n = 7;
riggedsinefit[x_] =
 Chop[ComplexExpand[FastFourierfit[rigged, 2 L, n, x]]]
```

$0.451469 \sin\left[\frac{\pi x}{3}\right] - 0.065733 \sin[\pi x] - 0.0132391 \sin\left[\frac{5 \pi x}{3}\right]$

```
fitplot = Plot[{starterposition[x], riggedsinefit[x]}, {x, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
  AxesLabel → {"x", ""}];
```



That's a great fit.

Now look at the rigged sine fit of starterposition[x]:

```
riggedsinefit[x]
```

$0.451469 \sin\left[\frac{\pi x}{3}\right] - 0.065733 \sin[\pi x] - 0.0132391 \sin\left[\frac{5 \pi x}{3}\right]$

Pick off the coefficients of the Sin[$\frac{(k\pi)x}{L}$] terms:

```
Clear[A, k]
A[k_] := Coefficient[riggedsinefit[x], Sin[(k π) x / L]];
coeffs = Table[A[k], {k, 1, n}]
```

{0.451469, 0, -0.065733, 0, -0.0132391, 0, 0}

The reason you run k from 1 to n is to pick up all the coefficients.

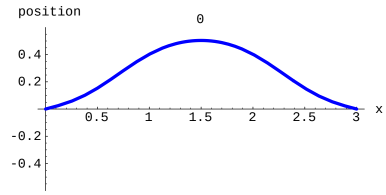
Now you're done because you can write down approxposition[x,t]:

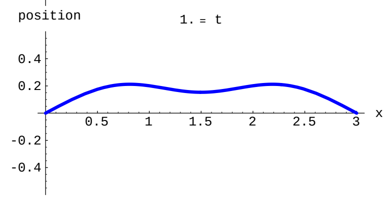
```
Clear[approxposition, x, t]
approxposition[x_, t_] = Sum[A[k] Cos[(k π) t / L] Sin[(k π) x / L], {k, 1, Length[coeffs]}]
```

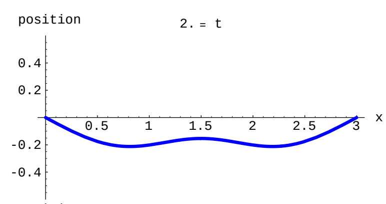
$0.451469 \cos\left[\frac{\pi t}{3}\right] \sin\left[\frac{\pi x}{3}\right] - 0.065733 \cos[\pi t] \sin[\pi x] - 0.0132391 \cos\left[\frac{5 \pi t}{3}\right] \sin\left[\frac{5 \pi x}{3}\right]$

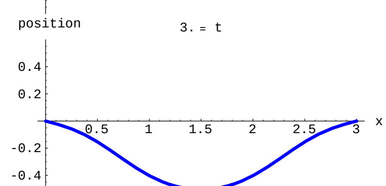
Here comes the movie:

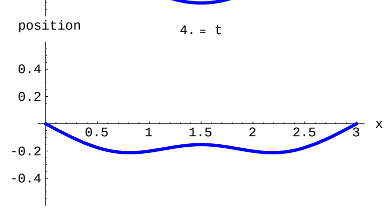
```
Clear[positionplotter]
positionplotter[t_] := Plot[approxposition[x, t],
  {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  PlotRange → {-0.6, 0.6}, AxesLabel → {"x", " position"},
  PlotLabel → N[t] "= t", AspectRatio → 1/2];
timejump = L/3;
Table[positionplotter[t], {t, 0, 2 L, timejump}]
```

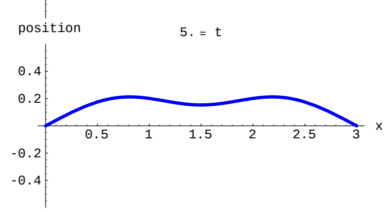


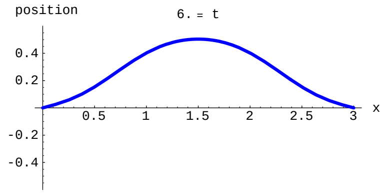












{- Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -,
- Graphics -, - Graphics -}

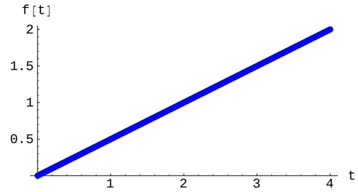
Hey Mr. Disney, look at that.

B.4) Fourier integral fit

Here's how you can use integrals to go after a Fourier fit on an interval [0,L]:

Start with the function and its plot on [0,L]:

```
Clear[f, t]
f[t_] = t/2;
L = 4;
Plot[f[t], {t, 0, L}, PlotStyle → {{Thickness[0.02], Blue}},
 AxesLabel → {"t", "f[t]"}];
```



To fit f[t] on [0,L] with Fourier Integral fit, you fit with combinations

$E^{\frac{Ik2\pi t}{L}}$

this way:

```
Clear[A, t]
A[k_] := A[k] = N[(∫_0^L f[t] E^(-I k (2 π) t / L) ⅆt) / L];
```

```
Clear[k, complexfitter]
m = 5;
complexfitter[t_] = Chop[Sum[A[k] E^(I k (2 π) t / L), {k, -m, m}]]
```

$1. - 0.31831\, I\, E^{-\frac{1}{2} I\pi t} + 0.31831\, I\, E^{\frac{I\pi t}{2}} - 0.159155\, I\, E^{-I\pi t} + 0.159155\, I\, E^{I\pi t} - 0.106103\, I\, E^{-\frac{3}{2} I\pi t} + 0.106103\, I\, E^{\frac{3I\pi t}{2}} - 0.0795775\, I\, E^{-2I\pi t} + 0.0795775\, I\, E^{2I\pi t} - 0.063662\, I\, E^{-\frac{5}{2} I\pi t} + 0.063662\, I\, E^{\frac{5I\pi t}{2}}$

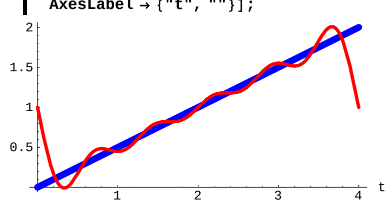
Covert to combinations of Sine and Cosine waves.

```
Clear[realfitter, t]
realfitter[t_] = Chop[ComplexExpand[complexfitter[t]]]
```

$1. - 0.63662 \,\mathrm{Sin}\left[\frac{\pi t}{2}\right] - 0.31831 \,\mathrm{Sin}[\pi t] - 0.212207 \,\mathrm{Sin}\left[\frac{3\pi t}{2}\right] - 0.159155 \,\mathrm{Sin}[2\pi t] - 0.127324 \,\mathrm{Sin}\left[\frac{5\pi t}{2}\right]$

Check out the fit with a plot:

```
Plot[{f[t], realfitter[t]}, {t, 0, L},
 PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.01], Red}},
 AxesLabel → {"t", ""}];
```



The periodic nature of the Fourier fit ruins the fit at the ends, but inside [0,L], the fit is not bad.

□B.4.a)

How do you try to go after a better Fourier Integral fit?

□Answer:

The same way you go after a better fast Fourier fit:

You increase n.

Try it:

```
Clear[A, t]
A[k_] := A[k] = N[(∫_0^L f[t] E^(-I k (2 π) t / L) ⅆt) / L];
Clear[k, complexfitter]
n = 13;
complexfitter[t_] = Chop[Sum[A[k] E^(I k (2 π) t / L), {k, -n, n}]]
```

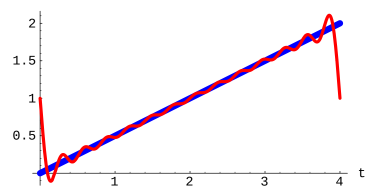
$1. - 0.31831\, I\, E^{-\frac{1}{2} I\pi t} + 0.31831\, I\, E^{\frac{I\pi t}{2}} - 0.159155\, I\, E^{-I\pi t} + 0.159155\, I\, E^{I\pi t} - 0.106103\, I\, E^{-\frac{3}{2} I\pi t} + 0.106103\, I\, E^{\frac{3I\pi t}{2}} - 0.0795775\, I\, E^{-2I\pi t} + 0.0795775\, I\, E^{2I\pi t} - 0.063662\, I\, E^{-\frac{5}{2} I\pi t} + 0.063662\, I\, E^{\frac{5I\pi t}{2}} - 0.0530516\, I\, E^{-3I\pi t} + 0.0530516\, I\, E^{3I\pi t} - 0.0454728\, I\, E^{-\frac{7}{2} I\pi t} + 0.0454728\, I\, E^{\frac{7I\pi t}{2}} - 0.0397887\, I\, E^{-4I\pi t} + 0.0397887\, I\, E^{4I\pi t} - 0.0353678\, I\, E^{-\frac{9}{2} I\pi t} + 0.0353678\, I\, E^{\frac{9I\pi t}{2}} - 0.031831\, I\, E^{-5I\pi t} + 0.031831\, I\, E^{5I\pi t} - 0.0289373\, I\, E^{-\frac{11}{2} I\pi t} + 0.0289373\, I\, E^{\frac{11I\pi t}{2}} - 0.0265258\, I\, E^{-6I\pi t} + 0.0265258\, I\, E^{6I\pi t} - 0.0244854\, I\, E^{-\frac{13}{2} I\pi t} + 0.0244854\, I\, E^{\frac{13I\pi t}{2}}$

```
Clear[realfitter, t]
realfitter[t_] = Chop[ComplexExpand[complexfitter[t]]]
```

$1. - 0.63662 \,\mathrm{Sin}\left[\frac{\pi t}{2}\right] - 0.31831 \,\mathrm{Sin}[\pi t] - 0.212207 \,\mathrm{Sin}\left[\frac{3\pi t}{2}\right] - 0.159155 \,\mathrm{Sin}[2\pi t] - 0.127324 \,\mathrm{Sin}\left[\frac{5\pi t}{2}\right] - 0.106103 \,\mathrm{Sin}[3\pi t] - 0.0909457 \,\mathrm{Sin}\left[\frac{7\pi t}{2}\right] - 0.0795775 \,\mathrm{Sin}[4\pi t] - 0.0707355 \,\mathrm{Sin}\left[\frac{9\pi t}{2}\right] - 0.063662 \,\mathrm{Sin}[5\pi t] - 0.0578745 \,\mathrm{Sin}\left[\frac{11\pi t}{2}\right] - 0.0530516 \,\mathrm{Sin}[6\pi t] - 0.0489708 \,\mathrm{Sin}\left[\frac{13\pi t}{2}\right]$

Check out the fit with a plot:

```
Plot[{f[t], realfitter[t]}, {t, 0, L},
 PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.01], Red}},
 AxesLabel → {"t", ""}];
```



That's about as good as it can get. If f[0] had been the same as f[L], then the trouble near the endpoints would have disappeared.

□B.4.b.i) The formula for the Fourier Integral coefficients

$$\mathbf{A[k]} = \frac{1}{L}\int_0^L \mathbf{f[t]}\, E^{-\frac{Ik(2\pi)t}{L}}\, dt$$

What's the idea behind the formula

$$A[k] = \frac{1}{L}\int_0^L f[t]\, E^{-\frac{Ik(2\pi)t}{L}}\, dt$$

as used above?

□Answer:

The idea:

You go for the whole ball of wax in one gulp.

Given a function f[t] with period L, you go ahead and assume that f[t] can be written in a form like this:

```
m = 3;
Clear[k, f, complexfitter, A, L]
complexfitter[t_] = Sum[A[k] E^(I k 2 π t / L), {k, -m, m}]
```

$E^{-\frac{6I\pi t}{L}} A[-3] + E^{-\frac{4I\pi t}{L}} A[-2] + E^{-\frac{2I\pi t}{L}} A[-1] + A[0] + E^{\frac{2I\pi t}{L}} A[1] + E^{\frac{4I\pi t}{L}} A[2] + E^{\frac{6I\pi t}{L}} A[3]$

Once you go with the assumption that

f[t] = complexfitter[t],

you gotta agree that

$$\int_0^L \text{complexfitter}[t]\, E^{-\frac{Ip(2\pi)t}{L}}\, dt = \int_0^L f[t]\, E^{-\frac{Ip(2\pi)t}{L}}\, dt$$

no matter what p you go with.

Peek at what this tells you:

This might take a while.

```
Clear[p, f]
Table[{Expand[∫_0^L complexfitter[t] E^(-(I p (2 π) t)/L) dt],
  " must be", ∫_0^L f[t] E^(-(I p (2 π) t)/L) dt},
 {p, -m, m}]
```

{{L A[-3], must be, $\int_0^L E^{\frac{6 I \pi t}{L}}$ f[t] dt}, {L A[-2], must be, $\int_0^L E^{\frac{4 I \pi t}{L}}$ f[t] dt}, {L A[-1], must be, $\int_0^L E^{\frac{2 I \pi t}{L}}$ f[t] dt}, {L A[0], must be, $\int_0^L$ f[t] dt}, {L A[1], must be, $\int_0^L E^{-\frac{2 I \pi t}{L}}$ f[t] dt}, {L A[2], must be, $\int_0^L E^{-\frac{4 I \pi t}{L}}$ f[t] dt}, {L A[3], must be, $\int_0^L E^{-\frac{6 I \pi t}{L}}$ f[t] dt}}

Your eyes lead you to the formula

$$A[k] = \frac{1}{L}\int_0^L f[t]\, E^{-\frac{I k (2\pi) t}{L}}\, dt$$

This is the formula that was used in part i) so successfully.

Explanation over.

□B.4.b.ii)

Not so fast.
Wasn't this supposed to be a math course?
You know very well that a false assumption like
f[t] = complexfitter[t]
can be used to explain anything.
How do you justify this false assumption?

□Answer:

Good question.

The reason it works is that it is almost true.

Reason: For most functions f[t] with period L, only a bureaucratic bean-counter armed with a good magnifying glass could ever tell the difference between f[t] and the complexfitter you get with a very high m.

Try it on

$$f[t] = \sqrt{1 - \mathrm{Cos}[\pi t]}$$

Because the period of Cos[π t] is 2, the period of f[t] is

L = 2.

```
Clear[f, t, x]
f[t_] = √(1 - Cos[π t]);
L = 2;
m = 6;
Clear[k, complexfitter, A]
A[k_] := N[1/L] NIntegrate[f[x] E^(-(I k (2 π) x)/L), {x, 0, L}, AccuracyGoal → 3];
complexfitter[t_] = Chop[Sum_{k=-m}^{m} A[k] E^((I k (2 π) t)/L)]
```

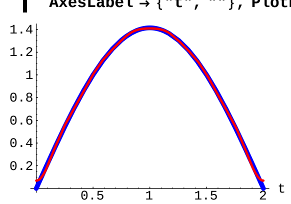
$0.900316 - 0.300105\, E^{-I \pi t} - 0.300105\, E^{I \pi t} - 0.0600211\, E^{-2 I \pi t} - 0.0600211\, E^{2 I \pi t} - 0.0257233\, E^{-3 I \pi t} - 0.0257233\, E^{3 I \pi t} - 0.0142907\, E^{-4 I \pi t} - 0.0142907\, E^{4 I \pi t} - 0.0090941\, E^{-5 I \pi t} - 0.0090941\, E^{5 I \pi t} - 0.00629592\, E^{-6 I \pi t} - 0.00629592\, E^{6 I \pi t}$

This time the A[k]'s are calculated with NIntegrate.

```
Clear[realfitter, t]
realfitter[t_] = Chop[ComplexExpand[complexfitter[t]]]
```

0.900316 − 0.600211 Cos[π t] − 0.120042 Cos[2 π t] − 0.0514466 Cos[3 π t] − 0.0285815 Cos[4 π t] − 0.0181882 Cos[5 π t] − 0.0125918 Cos[6 π t]

```
Plot[{f[t], realfitter[t]}, {t, 0, L},
 PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.01], Red}},
 AxesLabel → {"t", ""}, PlotRange → All];
```



Tell that miserable bean counter to get out of your life.

□B.4.c.i) The connection between fast Fourier Fit and Fourier integral fit

Activate this code:

```
Clear[FastFourierfit, Fourierfitters,
 F, Fvalues, n, k, jump, num, numtab, coeffs, t, L]
jump[n_] := jump[n] = N[1/(2 n)]; Fvalues[F_, L_, n_] :=
 N[Table[F[L t], {t, 0, 1 - jump[n], jump[n]}]]; numtab[n_] :=
 numtab[n] = Table[k, {k, 1, n}]; Fourierfitters[L_, n_, t_] :=
 Table[E^((2 π I k t)/L), {k, -n + 1, n - 1}]; coeffs[n_, list_] :=
 1/N[√Length[list]] (Join[Reverse[Fourier[list][[numtab[n]]]],
   InverseFourier[list][[Drop[numtab[n], 1]]]])
FastFourierfit[F_, L_, n_, t_] :=
 Chop[Fourierfitters[L, n, t] . coeffs[n, Fvalues[F, L, n]]];
L := Expand[a ∂_{#2,2} #1 + b ∂_#2 #1 + c #1] &
```

Go with f[t] = t (3 - t) on [0, l} with L = 3 and compare what you get

```
Clear[f, t]
f[t_] = t (3 - t);
L = 3;
Clear[A, t]
A[k_] := A[k] = N[(∫_0^L f[t] E^(-(I k (2 π) t)/L) dt)/L];
Clear[k, complexintegralfitter]
m = 5;
complexintegralfitter[t_] = Chop[Sum_{k=-m}^{m} A[k] E^((I k (2 π) t)/L)]
```

$1.5 - 0.455945\, E^{-\frac{2}{3} I \pi t} - 0.455945\, E^{\frac{2 I \pi t}{3}} - 0.113986\, E^{-\frac{4}{3} I \pi t} - 0.113986\, E^{\frac{4 I \pi t}{3}} - 0.0506606\, E^{-2 I \pi t} - 0.0506606\, E^{2 I \pi t} - 0.0284966\, E^{-\frac{8}{3} I \pi t} - 0.0284966\, E^{\frac{8 I \pi t}{3}} - 0.0182378\, E^{-\frac{10}{3} I \pi t} - 0.0182378\, E^{\frac{10 I \pi t}{3}}$

```
n = m + 1;
FastFourierfit[f, L, n, t]
```

$1.48958 - 0.466506\, E^{-\frac{2}{3} I \pi t} - 0.466506\, E^{\frac{2 I \pi t}{3}} - 0.125\, E^{-\frac{4}{3} I \pi t} - 0.125\, E^{\frac{4 I \pi t}{3}} - 0.0625\, E^{-2 I \pi t} - 0.0625\, E^{2 I \pi t} - 0.0416667\, E^{-\frac{8}{3} I \pi t} - 0.0416667\, E^{\frac{8 I \pi t}{3}} - 0.0334936\, E^{-\frac{10}{3} I \pi t} - 0.0334936\, E^{\frac{10 I \pi t}{3}}$

They are almost the same.
Subtract them:

```
Expand[complexintegralfitter[t] - FastFourierfit[f, L, n, t]]
```

$0.0104167 + 0.010561\, E^{-\frac{2}{3} I \pi t} + 0.010561\, E^{\frac{2 I \pi t}{3}} + 0.0110137\, E^{-\frac{4}{3} I \pi t} + 0.0110137\, E^{\frac{4 I \pi t}{3}} + 0.0118394\, E^{-2 I \pi t} + 0.0118394\, E^{2 I \pi t} + 0.0131701\, E^{-\frac{8}{3} I \pi t} + 0.0131701\, E^{\frac{8 I \pi t}{3}} + 0.0152558\, E^{-\frac{10}{3} I \pi t} + 0.0152558\, E^{\frac{10 I \pi t}{3}}$

Puny coefficients reflecting how close the fast Fourier fit is to the Fourier integral fit.

Try it again with a bigger m:

```
Clear[k, complexintegralfitter]
m = 10;
complexintegralfitter[t_] = Chop[Sum_{k=-m}^{m} A[k] E^((I k (2 π) t)/L)];
n = m + 1;
Expand[complexintegralfitter[t] - FastFourierfit[f, L, n, t]]
```

$0.00309917 + 0.00311185\, E^{-\frac{2}{3} I \pi t} + 0.00311185\, E^{\frac{2 I \pi t}{3}} + 0.00315039\, E^{-\frac{4}{3} I \pi t} + 0.00315039\, E^{\frac{4 I \pi t}{3}} + 0.00321633\, E^{-2 I \pi t} + 0.00321633\, E^{2 I \pi t} + 0.00331238\, E^{-\frac{8}{3} I \pi t} + 0.00331238\, E^{\frac{8 I \pi t}{3}} + 0.00344268\, E^{-\frac{10}{3} I \pi t} + 0.00344268\, E^{\frac{10 I \pi t}{3}} + 0.00361324\, E^{-4 I \pi t} + 0.00361324\, E^{4 I \pi t} + 0.00383251\, E^{-\frac{14}{3} I \pi t} + 0.00383251\, E^{\frac{14 I \pi t}{3}} + 0.00411247\, E^{-\frac{16}{3} I \pi t} + 0.00411247\, E^{\frac{16 I \pi t}{3}} + 0.00447017\, E^{-6 I \pi t} + 0.00447017\, E^{6 I \pi t} + 0.00493027\, E^{-\frac{20}{3} I \pi t} + 0.00493027\, E^{\frac{20 I \pi t}{3}}$

How do you explain this striking similarity of the two Fourier fits?

□Answer:

The fast Fourier fit tries to fit at equally spaced points on the plot of f[t] on [0,L]. When you go with a high n in FastFourierfit[f,L,n,t], you are fitting so many densely packed points that FastFourierfit[f,L,n,t] is almost fitting all the points on the plot of f[t] on [0,L].

This nearly replicates the Fourier integral fit of f[t] on [0,L] because the Fourier integral fit tries to fit the plot of f[t] all the points on the plot of f[t] on [0,L].

◻B.4.c.ii) Practicalities

Explain the following statement:
"Fast Fourier fit is almost always fast and practical; Fourier integral fit is almost sometimes low and slow and sometimes impractical and even impossible."

◻Answer:

The Achilles heel of the Fourier integral fit is the integral

$$A[k] = \frac{1}{L}\int_0^L f[t]\, E^{-\frac{I k (2\pi) t}{L}}\, dt$$

that most be done to calculate the coefficients. It is possible to try to calculate them with NIntegrate, but the highly oscillatory nature of

$$f[t]\, E^{-\frac{I k (2\pi) t}{L}}$$

sometimes makes the integrals impractical for NIntegrate.

Fast Fourier fit carries none of this disappointing baggage.

It runs quickly and accurately - provided no impulses are around.

In their book, "Numerical Methods and Software" (Prentice-Hall, 1989),

David Kahner, Cleve Moler and Stephen Nash say

"Many knowledgeable people feel that [Tukey's fast Fourier fit] is the single most important contribution to computing since the advent of the stored programming concept."

◻B.4.c.iii)

Why do folks want to do Fourier integral fit?

◻Calculational Answer:

If the professors demand that the students work by hand, then the student has little choice between Fourier integral fit and fast Fourier fit.

Reason:

Doing fast Fourier fit by hand is completely out of the question.

Doing integrals by hand is sometimes possible but hardly ever a good use of your time.

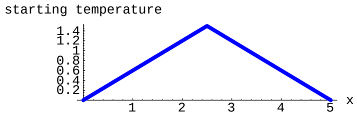
◻Theoretical Answer:

In theoretical situations, most good folks want to use the Fourier integral fit because it gives them a specific formula to work with. But after the theory is developed and calculations begin, then the same folks go with fast Fourier fit.

B.5) Fourier integral fit and the heat equation

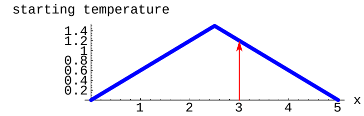
Start with a heated wire L units long with the temperature allowed to vary from position to position on the wire. Think of the wire as the interval $0 \le x \le L$. At the start of the experiment, you instantly cool the ends at x = 0 and x = L and maintain these ends at temperature 0, and you take pains to guarantee that the rest of the wire is perfectly insulated.
At the start of this particular experiment, the temperature of the wire at position x (for $0 \le x \le L = 5$) is given by the following function startertemp[x]:

```
L = 5;
Clear[startertemp, x]
startertemp[x_] = 3 Abs[0.2 x - Round[0.2 x]];
starterplot = Plot[startertemp[x],
  {x, 0, L}, PlotStyle → {{Thickness[0.015], Blue}},
  AxesLabel → {"x", "starting temperature"}];
```



To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] :=
 Arrow[{0, startertemp[x]}, Tail → {x, 0}, VectorColor → Red];
Show[starterplot, pointer[3]];
```



Think of the interval [0,L] = [0,5] as the wire.
The tip of the pointer tells you the starting temperature (at time t = 0) at the tail of the pointer.
Your problem here is to use Fourier Integral fit to come up with a function temp[x,t] that estimates the temperature of the wire at position x at time t after the experiment begins.
Do it.

◻Answer:

The function startertemp[x] is ripe for a rigged Fourier Sine fit on [0,L] because

startertemp[x] = 0 for x = 0 and x = L:

```
{startertemp[0], startertemp[L]}
{0, 0.}
```

Rig startertemp[x] for a pure sine fit on [0,L] and get a goodFourier integral sine fit of startertemp[x] on [0,L]:

If none of this makes sense to you, then look at B.1) immediately above.

```
Clear[rigged]
rigged[x_] := startertemp[x] /; 0 ≤ x ≤ L;
rigged[x_] := -startertemp[2 L - x] /; L < x ≤ 2 L;
Clear[A, t]
A[k_] := A[k] = (1 / (2 L)) NIntegrate[rigged[x] E^(-I k x 2π/(2 L)), {x, 0, 2 L}];
Clear[k, complexfitter]
m = 5;
complexfitter[x_] = Chop[Sum[A[k] E^(I k (2π) x/(2 L)), {k, -m, m}]]
```

$0.607927\, I\, E^{-\frac{1}{5} I \pi x} - 0.607927\, I\, E^{\frac{I \pi x}{5}} - 0.0675475\, I\, E^{-\frac{3}{5} I \pi x} + 0.0675475\, I\, E^{\frac{3 I \pi x}{5}} + 0.0243171\, I\, E^{-I \pi x} - 0.0243171\, I\, E^{I \pi x}$

Covert to combinations of Sine and Cosine waves.

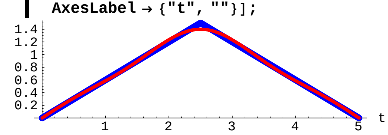
```
Clear[approxstartertemp, x]
approxstartertemp[x_] = Chop[ComplexExpand[complexfitter[x]]]
```

$1.21585 \operatorname{Sin}\left[\frac{\pi x}{5}\right] - 0.135095 \operatorname{Sin}\left[\frac{3 \pi x}{5}\right] + 0.0486342 \operatorname{Sin}[\pi x]$

Check out the fit with a plot:

The periodic nature of the Fourier fit ruins the fit at the ends, but inside [0,L], the fit is not bad.

```
Plot[{startertemp[x], approxstartertemp[x]}, {x, 0, L},
 PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.01], Red}},
 AxesLabel → {"t", ""}];
```



That's a decent fit.

Now look at the rigged sine fit of startertemp[x]:

```
approxstartertemp[x]
```

$1.21585 \operatorname{Sin}\left[\frac{\pi x}{5}\right] - 0.135095 \operatorname{Sin}\left[\frac{3 \pi x}{5}\right] + 0.0486342 \operatorname{Sin}[\pi x]$

Pick off the coefficients of the Sin[$\frac{(k\pi)x}{L}$] terms:

```
Clear[A, k]
A[k_] := Coefficient[approxstartertemp[x], Sin[(k π) x / L]];
Table[A[k], {k, 1, m}]
{1.21585, 0, -0.135095, 0, 0.0486342}
```

The reason you run k from 1 to n is to pick up all the coefficients.

Now you're done because you can write downapprox temp[x,t].

It's just:

```
Clear[temp, x, t]
approxtemp[x_, t_] = Sum[A[k] (E^(-(k π/L)^2 t) Sin[(k π) x / L]), {k, 1, m}]
```

$$1.21585\, E^{-\frac{\pi^2 t}{25}}\, \mathrm{Sin}\left[\frac{\pi x}{5}\right] - 0.135095\, E^{-\frac{9\pi^2 t}{25}}\, \mathrm{Sin}\left[\frac{3\pi x}{5}\right] + 0.0486342\, E^{-\pi^2 t}\, \mathrm{Sin}[\pi x]$$

Tthat's all there is to it.

DE.09 The Heat Equation

$\partial_{\{x,2\}}$ temp[x, t] = ∂_t temp[x, t]

and the Wave Equation

$\partial_{\{x,2\}}$ position[x, t] = $\partial_{\{t,2\}}$ position[x, t]

Tutorials

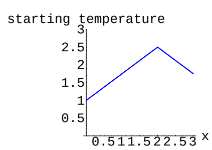
T.1) Different starter temperatures on the left and right: The general case of the heat equation

This problem is likely to leave you clueless unless you have some experience with the Basics.

□T.1.a.i)

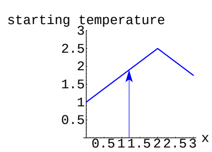
Start with a heated wire L units long with the temperature allowed to vary from position to position on the wire. Think of the wire as the interval 0 ≤ x ≤ L. At the start of the experiment, you apply apparatus that maintains the temperatures at the end points of then wire. At the start of this particular experiment, the temperature of the wire at position x (for 0 ≤ x ≤ L= 3) is given by the following function startertemp[t]:

```
L = 3;
Clear[startertemp, x]
startertemp[x_] = 3 Abs[0.25 x - Round[0.25 x]] + 1;
starterplot = Plot[startertemp[x], {x, 0, L},
  PlotStyle → {{Thickness[0.01], Blue}}, PlotRange → {0, 3},
  AxesLabel → {"x", "starting temperature"}];
```



To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] := Arrow[{0, startertemp[x]}, Tail → {x, 0}];
Show[starterplot, pointer[1.2]];
```



Think of the interval [0,L] = [0, 3] as the wire.
The tip of the pointer tells you the starting temperature (at time t = 0) at the tail of the pointer.
Your problem here is to come up with the function temp[x,t] that estimates the temperature of the wire at position x at time t after the experiment begins. This means that you are looking for a function temp[x,t] satisfying the heat equation

$\partial_{\{x,2\}}$ temp[x, t] = ∂_t temp[x, t]

with
-> temp[x, 0] a good approximation of startertemp[x] and
-> temp[0, t] = startertemp[0] and temp[L ,t] = startertemp[L] for all t's.
Do it.
Make a movie.

□**Answer:**

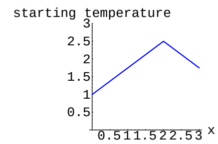
The function startertemp[x] is NOT ripe for a rigged Fourier Sine fit on [0,L] because

startertemp[0]≠ 0 and startertemp[L] ≠ 0 .

```
{startertemp[0], startertemp[L]}
{1, 1.75}
```

To see how to get around this little obstacle, look at the plot of startertemp[x]:

```
Show[starterplot];
```

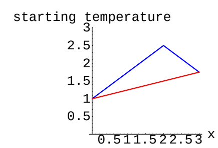


Run a line through the endpoints of the plot.

```
Clear[line]
line[x_] = (startertemp[L] - startertemp[0]) x / L + startertemp[0]
1 + 0.25 x
```

Take a look:

```
Plot[{startertemp[x], line[x]}, {x, 0, L},
 PlotStyle → {{Thickness[0.01], Blue}, {Thickness[0.01], Red}},
 PlotRange → {0, 3}, AxesLabel → {"x", "starting temperature"}];
```

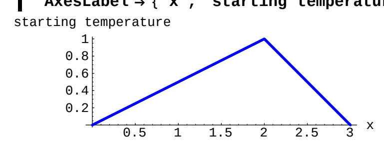


Put

adjustedstartertemp[x] = startertemp[x] - line[x]

and plot:

```
Clear[adjustedstartertemp]
adjustedstartertemp[x_] = startertemp[x] - line[x];
Plot[adjustedstartertemp[x],
 {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
 AxesLabel → {"x", "starting temperature"}];
```



The function adjustedstartertemp[x] IS ripe for a rigged Fourier Sine fit on [0,L] because

adjustedstartertemp[0] = 0 and adjustedstartertemp[L] = 0.

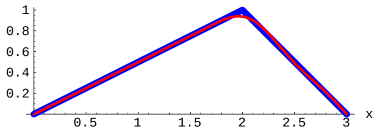
```
{adjustedstartertemp[0], adjustedstartertemp[L]}
{0, 0.}
```

Rig adjustedstartertemp[x] for a pure sine fit on [0,L] and get a good sine fit of adjustedstartertemp[x] on [0,L]:

```
Clear[rigged]
rigged[x_] := adjustedstartertemp[x] /; 0 ≤ x ≤ L;
rigged[x_] := -adjustedstartertemp[2 L - x] /; L < x ≤ 2 L;
n = 10;
Clear[riggedsinefit]
riggedsinefit[x_] =
 Chop[ComplexExpand[FastFourierfit[rigged, 2 L, n, x]]]
```

$$0.787364\, \mathrm{Sin}\left[\frac{\pi x}{3}\right] - 0.195559\, \mathrm{Sin}\left[\frac{2\pi x}{3}\right] + 0.00110072\, \mathrm{Sin}[\pi x] + 0.0461653\, \mathrm{Sin}\left[\frac{4\pi x}{3}\right] - 0.03\, \mathrm{Sin}\left[\frac{5\pi x}{3}\right] + 0.0025727\, \mathrm{Sin}[2\pi x] + 0.0113915\, \mathrm{Sin}\left[\frac{7\pi x}{3}\right] - 0.0108981\, \mathrm{Sin}\left[\frac{8\pi x}{3}\right] + 0.00512782\, \mathrm{Sin}[3\pi x]$$

```
fitplot = Plot[{adjustedstartertemp[x], riggedsinefit[x]}, {x, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
  AxesLabel → {"x", ""}];
```



That's a decent fit.

Now look at the rigged sine fit of startertep[x]:

```
riggedsinefit[x]
```

$$0.787364\ \mathrm{Sin}\left[\frac{\pi x}{3}\right] - 0.195559\ \mathrm{Sin}\left[\frac{2\pi x}{3}\right] + 0.00110072\ \mathrm{Sin}[\pi x] + 0.0461653\ \mathrm{Sin}\left[\frac{4\pi x}{3}\right] - 0.03\ \mathrm{Sin}\left[\frac{5\pi x}{3}\right] + 0.0025727\ \mathrm{Sin}[2\pi x] + 0.0113915\ \mathrm{Sin}\left[\frac{7\pi x}{3}\right] - 0.0108981\ \mathrm{Sin}\left[\frac{8\pi x}{3}\right] + 0.00512782\ \mathrm{Sin}[3\pi x]$$

Pick off the coefficients of the Sin[(k Pi/L) x] terms:

```
Clear[A, k]
A[k_] := Coefficient[riggedsinefit[x], Sin[(k π) x / L]];
coeffs = Table[A[k], {k, 1, n}]
```

{0.787364, -0.195559, 0.00110072, 0.0461653, -0.03, 0.0025727, 0.0113915, -0.0108981, 0.00512782, 0}

The reason you run k from 1 to n is to pick up all the coefficients.

Now you're done because you can write down temp[x,t]:

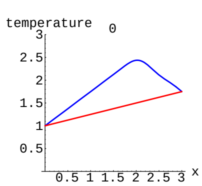
```
Clear[temp, x, t]
temp[x_, t_] = line[x] + Sum[A[k] E^(-(k π / L)^2 t) Sin[(k π) x / L], {k, 1, n}]
```

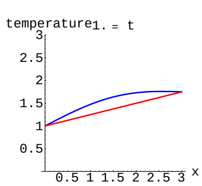
$$1 + 0.25 x + 0.787364\, E^{-\frac{\pi^2 t}{9}} \mathrm{Sin}\left[\frac{\pi x}{3}\right] - 0.195559\, E^{-\frac{4\pi^2 t}{9}} \mathrm{Sin}\left[\frac{2\pi x}{3}\right] + 0.00110072\, E^{-\pi^2 t} \mathrm{Sin}[\pi x] + 0.0461653\, E^{-\frac{16\pi^2 t}{9}} \mathrm{Sin}\left[\frac{4\pi x}{3}\right] - 0.03\, E^{-\frac{25\pi^2 t}{9}} \mathrm{Sin}\left[\frac{5\pi x}{3}\right] + 0.0025727\, E^{-4\pi^2 t} \mathrm{Sin}[2\pi x] + 0.0113915\, E^{-\frac{49\pi^2 t}{9}} \mathrm{Sin}\left[\frac{7\pi x}{3}\right] - 0.0108981\, E^{-\frac{64\pi^2 t}{9}} \mathrm{Sin}\left[\frac{8\pi x}{3}\right] + 0.00512782\, E^{-9\pi^2 t} \mathrm{Sin}[3\pi x]$$

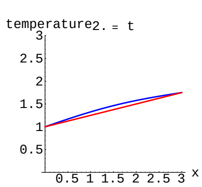
Notice the line term at the beginning.

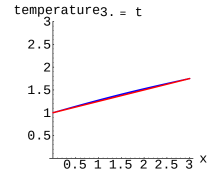
Here comes a movie showing the approximate temperature along with the line function.

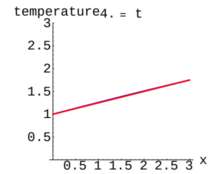
```
Clear[tempplotter]
tempplotter[t_] := Plot[{temp[x, t], line[x]}, {x, 0, L},
  PlotStyle → {{Thickness[0.01], Blue}, {Thickness[0.01], Red}},
  PlotRange → {0, 3},
  AxesLabel → {"x", " temperature"}, PlotLabel → N[t] "= t"];
timejump = 1;
Table[tempplotter[t], {t, 0, 6, timejump}]
```

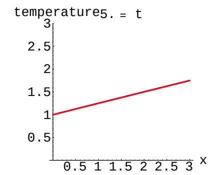


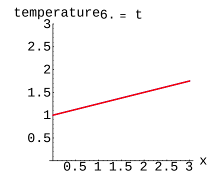












{- Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -}

Grab all the plot and animate.

You are seeing the wire settle into its steady state temparature which at point x with $0 \le x \le L$ is given by:

```
line[x]
```

$1 + 0.25 x$

□T.1.a.ii)

How does the heat equation explain why the little stunt with the line works?

□**Answer:**

Engineering studies have shown that after the appropriate unit adjustments are made, the function temp[x,t] satisfies the heat equation

$$\partial_{\{x,2\}}\,\mathrm{temp}[x, t] = \partial_t\,\mathrm{temp}[x, t]$$

with

-> temp[x, 0] = startertemp[x] and

-> temp[0, t] = startertemp[0] and temp[L, t] = startertemp[L] for all t's.

□Check out the heat equation D[temp[x,t], {x,2}] = D[temp[x,t], t]:

To do this, look at:

```
temp[x, t]
```

$$1 + 0.25 x + 0.787364\, E^{-\frac{\pi^2 t}{9}} \mathrm{Sin}\left[\frac{\pi x}{3}\right] - 0.195559\, E^{-\frac{4\pi^2 t}{9}} \mathrm{Sin}\left[\frac{2\pi x}{3}\right] + 0.00110072\, E^{-\pi^2 t} \mathrm{Sin}[\pi x] + 0.0461653\, E^{-\frac{16\pi^2 t}{9}} \mathrm{Sin}\left[\frac{4\pi x}{3}\right] - 0.03\, E^{-\frac{25\pi^2 t}{9}} \mathrm{Sin}\left[\frac{5\pi x}{3}\right] + 0.0025727\, E^{-4\pi^2 t} \mathrm{Sin}[2\pi x] + 0.0113915\, E^{-\frac{49\pi^2 t}{9}} \mathrm{Sin}\left[\frac{7\pi x}{3}\right] - 0.0108981\, E^{-\frac{64\pi^2 t}{9}} \mathrm{Sin}\left[\frac{8\pi x}{3}\right] + 0.00512782\, E^{-9\pi^2 t} \mathrm{Sin}[3\pi x]$$

Note the Sine terms. In the Basics, it was seen that the Sine terms were specially selected to solve the heat equation. So checking that temp[x,t] satisies the heat equation reduces to checking that the first two terms of temp[x,t] solve the heat equation. The first two terms are:

```
line[x]
```

```
1 + 0.25 x
```

Check whether line[x] solves the heat equation

$\partial_{\{x,2\}}$ line[x] = ∂_t line[x]:

```
{∂{x,2} line[x], ∂t line[x]}
{0, 0}
```

This tells you that

$\partial_{\{x,2\}}$ line[x] = ∂_t line[x].

and explains why temp[x,t] solves the heat equation.

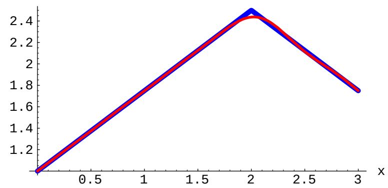
□Check out temp[0,0] = startertemp[0] and temp[L,0] = startertemp[L]

```
{startertemp[0], temp[0, 0]}
{1, 1}
{startertemp[L], temp[L, 0]}
{1.75, 1.75}
```

Perfecto.

□Check out temp[x,0] ≈ startertemp[x]

```
Plot[{startertemp[x], temp[x, 0]}, {x, 0, L},
 PlotStyle → {{Thickness[0.015], Blue}, {Thickness[0.008], Red}},
 AxesLabel → {"x", ""}];
```



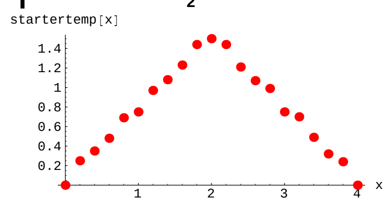
Good enough. The slight discrepanies indicate that temp[x,t] does not solve the original problem but does solve a problem virtually indistinguishable from the original problem.

```
coeffs[n_, list_] := Join[Reverse[Fourier[list][[numtab[n]]]],
   InverseFourier[list][[Drop[numtab[n], 1]]]] / N[√Length[list]]
FastFourierfit[F_, L_, n_, t_] :=
 Chop[Fourierfitters[L, n, t] . coeffs[n, Fvalues[F, L, n]]];
L := Expand[a ∂{#2,2} #1 + b ∂#2 #1 + c #1] &
```

□T.2.a) Heat equation

Start with a heated wire L units long with the temperature allowed to vary from position to position on the wire. Think of the wire as the interval $0 \le x \le L$. At the start of the experiment, you instantly cool the ends at x = 0 and x = L and maintain these ends at temperature 0, and you take pains to guarantee that the rest of the wire is perfectly insulated.
At the start of this partricular experiment, the temperature of the wire at position x (for $0 \le x \le L = 4$) is given by the following startertemp data:

```
startertempdata = {{0.0, 0.0}, {0.2, 0.25},
   {0.4, 0.35}, {0.6, 0.48}, {0.8, 0.69}, {1.0, 0.75},
   {1.2, 0.97}, {1.4, 1.08}, {1.6, 1.23}, {1.8, 1.44}, {2.0, 1.5},
   {2.2, 1.44}, {2.4, 1.21}, {2.6, 1.07}, {2.8, 0.99}, {3., 0.75},
   {3.2, 0.70}, {3.4, 0.49}, {3.6, 0.32}, {3.8, 0.24}, {4., 0.0}};
starterdataplot =
 ListPlot[startertempdata, PlotStyle → {PointSize[0.03], Red},
   AspectRatio → 1/2, AxesLabel → {"x", "startertemp[x]"}];
```



Your problem here is to come up with a function temp[x,t] that estimates the temperature of the wire at position x at time t after the experiment begins.
Do it.
Throw in a good movie.

□Answer:

□T.1.a.iii)

That idea of running a line connecting
{0, startertemp[0]} to {L, startertemp[L]}
looks like a really cheap trick.
Could you have run another curve through those points and gotten good results?

□Answer:

If you make the adjustment with any other function f[x] with f[0] = startertemp[0] and f[L] = startertemp[L], the issue is whether the heat equation is solved. This boils down to checking

Check whether f[x] solves the heat equation

$\partial_{\{x,2\}}$ f[x] = ∂_t f[x]:

```
Clear[f]
{∂{x,2} f[x], ∂t f[x]}
{f''[x], 0}
```

This tells you that to solve the heat equation you need f''[x] = 0. This forces you to go with a line:

```
DSolve[f''[x] == 0, f[x], x]
{{f[x] → C[1] + x C[2]}}
```

Not only was the line a cheap trick, it was the only possible adjustment that could have been made.

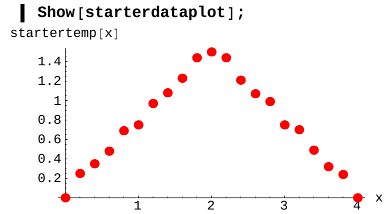
T.2) When the starter conditions are given by a data list

Activate this code:

```
Clear[FastFourierfit, Fourierfitters,
 F, Fvalues, n, k, jump, num, numtab, coeffs, t, L]
jump[n_] := jump[n] = N[1/(2 n)];
Fvalues[F_, L_, n_] := N[Table[F[L t], {t, 0, 1 - jump[n], jump[n]}]];
numtab[n_] := numtab[n] = Table[k, {k, 1, n}];
Fourierfitters[L_, n_, t_] := Table[E^(2 π I k t / L), {k, -n + 1, n - 1}];
```

Take another look at the data:

```
Show[starterdataplot];
```

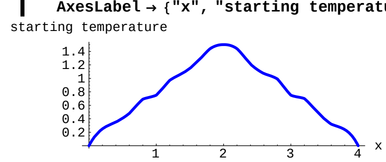


Now put an interpolating function through these data:

```
Clear[startertemp]
startertemp[t_] = Interpolation[startertempdata][t]
InterpolatingFunction[{{0., 4.}}, <>][t]
```

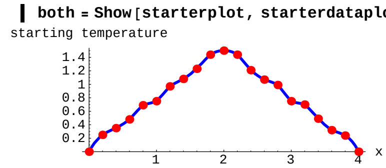
Take a look:

```
L = 4;
starterplot = Plot[startertemp[x],
   {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
   AxesLabel → {"x", "starting temperature"}];
```



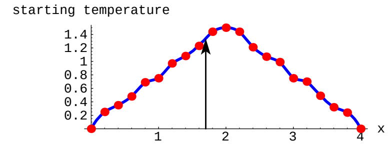
Check this plot against the data:

```
both = Show[starterplot, starterdataplot];
```



To fully understand the curve plot, look at this:

```
Clear[pointer]
pointer[x_] :=
  Arrow[{0, startertemp[x]}, Tail → {x, 0}, VectorColor → Black];
Show[both, pointer[1.7]];
```



Think of the interval [0,L] = [0,4] as the wire.

The tip of the pointer tells you the starting temperature (at time t = 0) at the tail of the pointer.

The function startertemp[x] is ripe for a rigged Fourier Sine fit on [0,L] because startertemp[x] = 0 for x = 0 and x = L:

```
{startertemp[0], startertemp[L]}
```

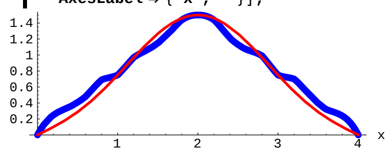
{0., 0.}

Rig startertemp[x] for a pure sine fit on [0,L] and get a good sine fit of startertemp[t] on [0,L]:

```
Clear[rigged]
rigged[x_] := startertemp[x] /; 0 ≤ x ≤ L;
rigged[x_] := -startertemp[2 L - x] /; L < x ≤ 2 L;
n = 4;
Clear[riggedsinefit]
riggedsinefit[x_] =
  Chop[ComplexExpand[FastFourierfit[rigged, 2 L, n, x]]]
```

$$1.28033\,\mathrm{Sin}\left[\frac{\pi x}{4}\right] - 0.21967\,\mathrm{Sin}\left[\frac{3\pi x}{4}\right]$$

```
fitplot = Plot[{startertemp[x], riggedsinefit[x]}, {x, 0, L},
   PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
   AxesLabel → {"x", ""}];
```

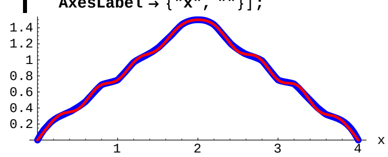


That's not a great fit. Go for a better fit by increasing n:

```
Clear[rigged]
rigged[x_] := startertemp[x] /; 0 ≤ x ≤ L;
rigged[x_] := -startertemp[2 L - x] /; L < x ≤ 2 L;
n = 20;
Clear[riggedsinefit]
riggedsinefit[x_] =
  Chop[ComplexExpand[FastFourierfit[rigged, 2 L, n, x]]]
```

$$1.27682\,\mathrm{Sin}\left[\frac{\pi x}{4}\right] + 0.000394774\,\mathrm{Sin}\left[\frac{\pi x}{2}\right] - 0.112577\,\mathrm{Sin}\left[\frac{3\pi x}{4}\right] + 0.000224514\,\mathrm{Sin}[\pi x] + 0.0697904\,\mathrm{Sin}\left[\frac{5\pi x}{4}\right] + 0.00732766\,\mathrm{Sin}\left[\frac{3\pi x}{2}\right] - 0.0199829\,\mathrm{Sin}\left[\frac{7\pi x}{4}\right] + 0.00176336\,\mathrm{Sin}[2\pi x] + 0.0309918\,\mathrm{Sin}\left[\frac{9\pi x}{4}\right] + 0.001\,\mathrm{Sin}\left[\frac{5\pi x}{2}\right] - 0.000031248\,\mathrm{Sin}\left[\frac{11\pi x}{4}\right] + 0.0024899\,\mathrm{Sin}[3\pi x] + 0.0280287\,\mathrm{Sin}\left[\frac{13\pi x}{4}\right] - 0.00570962\,\mathrm{Sin}\left[\frac{7\pi x}{2}\right] + 0.0277904\,\mathrm{Sin}\left[\frac{15\pi x}{4}\right] - 0.00285317\,\mathrm{Sin}[4\pi x] - 0.0196747\,\mathrm{Sin}\left[\frac{17\pi x}{4}\right] + 0.00022326\,\mathrm{Sin}\left[\frac{9\pi x}{2}\right] - 0.00924238\,\mathrm{Sin}\left[\frac{19\pi x}{4}\right]$$

```
fitplot = Plot[{startertemp[x], riggedsinefit[x]}, {x, 0, L},
   PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
   AxesLabel → {"x", ""}];
```



That's better.

Now look at the rigged sine fit of startertep[x]:

```
riggedsinefit[x]
```

$$1.27682\,\mathrm{Sin}\left[\frac{\pi x}{4}\right] + 0.000394774\,\mathrm{Sin}\left[\frac{\pi x}{2}\right] - 0.112577\,\mathrm{Sin}\left[\frac{3\pi x}{4}\right] + 0.000224514\,\mathrm{Sin}[\pi x] + 0.0697904\,\mathrm{Sin}\left[\frac{5\pi x}{4}\right] + 0.00732766\,\mathrm{Sin}\left[\frac{3\pi x}{2}\right] - 0.0199829\,\mathrm{Sin}\left[\frac{7\pi x}{4}\right] + 0.00176336\,\mathrm{Sin}[2\pi x] + 0.0309918\,\mathrm{Sin}\left[\frac{9\pi x}{4}\right] + 0.001\,\mathrm{Sin}\left[\frac{5\pi x}{2}\right] - 0.000031248\,\mathrm{Sin}\left[\frac{11\pi x}{4}\right] + 0.0024899\,\mathrm{Sin}[3\pi x] + 0.0280287\,\mathrm{Sin}\left[\frac{13\pi x}{4}\right] - 0.00570962\,\mathrm{Sin}\left[\frac{7\pi x}{2}\right] + 0.0277904\,\mathrm{Sin}\left[\frac{15\pi x}{4}\right] - 0.00285317\,\mathrm{Sin}[4\pi x] - 0.0196747\,\mathrm{Sin}\left[\frac{17\pi x}{4}\right] + 0.00022326\,\mathrm{Sin}\left[\frac{9\pi x}{2}\right] - 0.00924238\,\mathrm{Sin}\left[\frac{19\pi x}{4}\right]$$

Pick off the coefficients of the Sin[(k Pi/L) x] terms:

```
Clear[A, k]
A[k_] := Coefficient[riggedsinefit[x], Sin[(k π) x / L]];
coeffs = Table[A[k], {k, 1, n}]
```

{1.27682, 0.000394774, -0.112577, 0.000224514, 0.0697904, 0.00732766, -0.0199829, 0.00176336, 0.0309918, 0.001, -0.000031248, 0.0024899, 0.0280287, -0.00570962, 0.0277904, -0.00285317, -0.0196747, 0.00022326, -0.00924238, 0}

The reason you run k from 1 to n is to pick up all the coefficients.

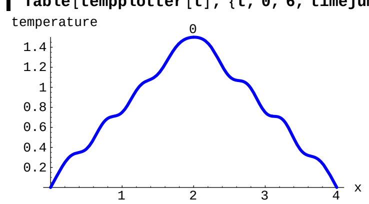
Now you're done because you can write down temp[x,t]:

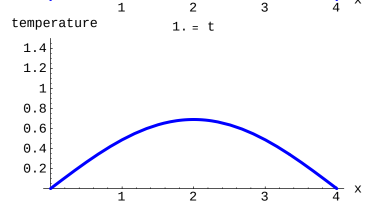
```
Clear[temp, x, t]
temp[x_, t_] = Sum[A[k] E^(-(k π / L)^2 t) Sin[(k π) x / L], {k, 1, n}]
```

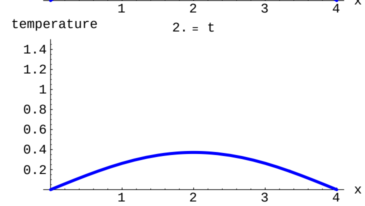
$$1.27682\,E^{-\frac{\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{\pi x}{4}\right] + 0.000394774\,E^{-\frac{\pi^2 t}{4}}\,\mathrm{Sin}\left[\frac{\pi x}{2}\right] - 0.112577\,E^{-\frac{9\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{3\pi x}{4}\right] + 0.000224514\,E^{-\pi^2 t}\,\mathrm{Sin}[\pi x] + 0.0697904\,E^{-\frac{25\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{5\pi x}{4}\right] + 0.00732766\,E^{-\frac{9\pi^2 t}{4}}\,\mathrm{Sin}\left[\frac{3\pi x}{2}\right] - 0.0199829\,E^{-\frac{49\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{7\pi x}{4}\right] + 0.00176336\,E^{-4\pi^2 t}\,\mathrm{Sin}[2\pi x] + 0.0309918\,E^{-\frac{81\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{9\pi x}{4}\right] + 0.001\,E^{-\frac{25\pi^2 t}{4}}\,\mathrm{Sin}\left[\frac{5\pi x}{2}\right] - 0.000031248\,E^{-\frac{121\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{11\pi x}{4}\right] + 0.0024899\,E^{-9\pi^2 t}\,\mathrm{Sin}[3\pi x] + 0.0280287\,E^{-\frac{169\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{13\pi x}{4}\right] - 0.00570962\,E^{-\frac{49\pi^2 t}{4}}\,\mathrm{Sin}\left[\frac{7\pi x}{2}\right] + 0.0277904\,E^{-\frac{225\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{15\pi x}{4}\right] - 0.00285317\,E^{-16\pi^2 t}\,\mathrm{Sin}[4\pi x] - 0.0196747\,E^{-\frac{289\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{17\pi x}{4}\right] + 0.00022326\,E^{-\frac{81\pi^2 t}{4}}\,\mathrm{Sin}\left[\frac{9\pi x}{2}\right] - 0.00924238\,E^{-\frac{361\pi^2 t}{16}}\,\mathrm{Sin}\left[\frac{19\pi x}{4}\right]$$

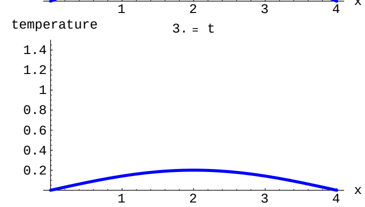
Here comes a movie:

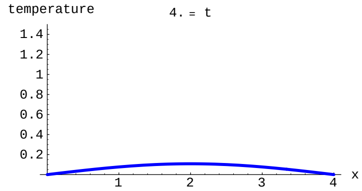
```
Clear[tempplotter]
tempplotter[t_] :=
  Plot[temp[x, t], {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
   PlotRange → {0, 1.5}, AxesLabel → {"x", " temperature"},
   PlotLabel → N[t] "= t", AspectRatio → 1/2];
timejump = 1;
Table[tempplotter[t], {t, 0, 6, timejump}]
```

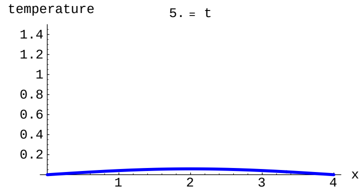


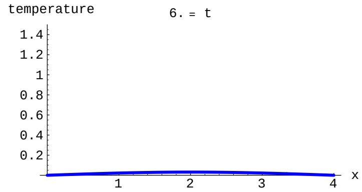













```
{- Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -,
 - Graphics -, - Graphics -}
```

Those exponentials are cooling this wire fast.

□T.2.b) Wave equation

Activate this code:

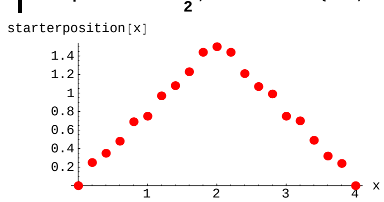
```
Clear[FastFourierfit, Fourierfitters,
 F, Fvalues, n, k, jump, num, numtab, coeffs, t, L]
jump[n_] := jump[n] = N[1/(2 n)];
Fvalues[F_, L_, n_] := N[Table[F[L t], {t, 0, 1 - jump[n], jump[n]}]];
numtab[n_] := numtab[n] = Table[k, {k, 1, n}];
Fourierfitters[L_, n_, t_] := Table[E^(2 π I k t/L), {k, -n + 1, n - 1}];
coeffs[n_, list_] := Join[Reverse[Fourier[list][[numtab[n]]]],
   InverseFourier[list][[Drop[numtab[n], 1]]]] / N[Sqrt[Length[list]]]
```

```
FastFourierfit[F_, L_, n_, t_] :=
 Chop[Fourierfitters[L, n, t] . coeffs[n, Fvalues[F, L, n]]];
L := Expand[a ∂{#2,2} #1 + b ∂#2 #1 + c #1] &
```

The ends of a guitar string are anchored at 0 and L on the x-axis and the string is pulled to an initial position and then allowed to vibrate on its own starting with initial velocity 0.

At the start of this partricular experiment, the position of the wire at position x (for $0 \le x \le L = 4$) is given by the following starterposition data:

```
starterpositiondata = {{0.0, 0.0}, {0.2, 0.25},
   {0.4, 0.35}, {0.6, 0.48}, {0.8, 0.69}, {1.0, 0.75},
   {1.2, 0.97}, {1.4, 1.08}, {1.6, 1.23}, {1.8, 1.44}, {2.0, 1.5},
   {2.2, 1.44}, {2.4, 1.21}, {2.6, 1.07}, {2.8, 0.99}, {3., 0.75},
   {3.2, 0.70}, {3.4, 0.49}, {3.6, 0.32}, {3.8, 0.24}, {4., 0.0}};
starterdataplot =
 ListPlot[starterpositiondata, PlotStyle → {PointSize[0.03], Red},
  AspectRatio → 1/2, AxesLabel → {"x", "starterposition[x]"}];
```

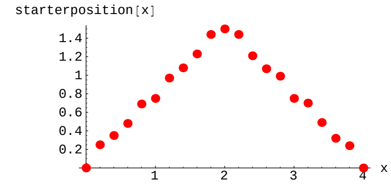


Your problem here is to come up with a function position[x,t] that estimates the position of the guitar string at position x on the x-axis at time t after the experiment begins.
Do it.
Throw in a good movie.

□Answer:

Take another look at the data:

```
Show[starterdataplot];
```

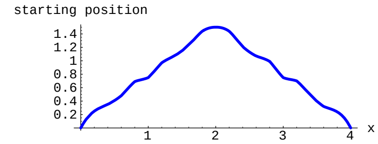


Now put an interpolating function through these data:

```
Clear[starterposition]
starterposition[t_] = Interpolation[starterpositiondata][t]
InterpolatingFunction[{{0., 4.}}, <>][t]
```

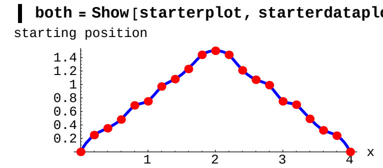
Take a look:

```
L = 4;
starterplot = Plot[starterposition[x],
  {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  AxesLabel → {"x", "starting position"}];
```



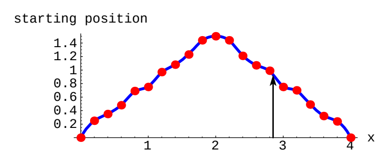
Check this plot against the data:

```
both = Show[starterplot, starterdataplot];
```



To fully understand the curve plot, look at this:

```
Clear[pointer]
pointer[x_] :=
 Arrow[{0, starttemp[x]}, Tail → {x, 0}, VectorColor → Black];
Show[both, pointer[2.85]];
```



The function starterposition[x] is ripe for a rigged Fourier Sine fit on [0,L] because starterposition[x] = 0 for x = 0 and x = L:

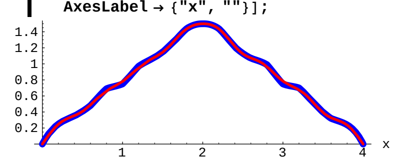
```
{starterposition[0], starterposition[L]}
{0., 0.}
```

Rig starterposition[x] for a pure sine fit on [0,L] and get a good sine fit of starterposition[t] on [0,L]:

```
Clear[rigged]
rigged[x_] := starterposition[x] /; 0 ≤ x ≤ L;
rigged[x_] := -starterposition[2 L - x] /; L < x ≤ 2 L;
Clear[riggedsinefit]
n = 18;
riggedsinefit[x_] =
 Chop[ComplexExpand[FastFourierfit[rigged, 2 L, n, x]]]
```

$$1.2771\,\mathrm{Sin}\left[\frac{\pi x}{4}\right] + 0.0004125\,\mathrm{Sin}\left[\frac{\pi x}{2}\right] - 0.112591\,\mathrm{Sin}\left[\frac{3\pi x}{4}\right] + 0.000257171\,\mathrm{Sin}[\pi x] + 0.0703317\,\mathrm{Sin}\left[\frac{5\pi x}{4}\right] + 0.00720654\,\mathrm{Sin}\left[\frac{3\pi x}{2}\right] - 0.0191593\,\mathrm{Sin}\left[\frac{7\pi x}{4}\right] + 0.00184198\,\mathrm{Sin}[2\pi x] + 0.0319138\,\mathrm{Sin}\left[\frac{9\pi x}{4}\right] + 0.000188565\,\mathrm{Sin}\left[\frac{5\pi x}{2}\right] + 0.00454808\,\mathrm{Sin}\left[\frac{11\pi x}{4}\right] + 0.00156063\,\mathrm{Sin}[3\pi x] + 0.0182007\,\mathrm{Sin}\left[\frac{13\pi x}{4}\right] - 0.00432241\,\mathrm{Sin}\left[\frac{7\pi x}{2}\right] + 0.0171384\,\mathrm{Sin}\left[\frac{15\pi x}{4}\right] - 0.00180089\,\mathrm{Sin}[4\pi x] - 0.00760997\,\mathrm{Sin}\left[\frac{17\pi x}{4}\right]$$

```
fitplot = Plot[{starterposition[x], riggedsinefit[x]}, {x, 0, L},
  PlotStyle → {{Thickness[0.02], Blue}, {Thickness[0.008], Red}},
  AxesLabel → {"x", ""}];
```



That's a pretty good fit.

Now look at the rigged sine fit of starterposition[x]:

```
riggedsinefit[x]
```

$$1.2771\, \text{Sin}\left[\frac{\pi x}{4}\right] + 0.0004125\, \text{Sin}\left[\frac{\pi x}{2}\right] - 0.112591\, \text{Sin}\left[\frac{3 \pi x}{4}\right] + 0.000257171\, \text{Sin}[\pi x] + 0.0703317\, \text{Sin}\left[\frac{5 \pi x}{4}\right] + 0.00720654\, \text{Sin}\left[\frac{3 \pi x}{2}\right] - 0.0191593\, \text{Sin}\left[\frac{7 \pi x}{4}\right] + 0.00184198\, \text{Sin}[2 \pi x] + 0.0319138\, \text{Sin}\left[\frac{9 \pi x}{4}\right] + 0.000188565\, \text{Sin}\left[\frac{5 \pi x}{2}\right] + 0.00454808\, \text{Sin}\left[\frac{11 \pi x}{4}\right] + 0.00156063\, \text{Sin}[3 \pi x] + 0.0182007\, \text{Sin}\left[\frac{13 \pi x}{4}\right] - 0.00432241\, \text{Sin}\left[\frac{7 \pi x}{2}\right] + 0.0171384\, \text{Sin}\left[\frac{15 \pi x}{4}\right] - 0.00180089\, \text{Sin}[4 \pi x] - 0.00760997\, \text{Sin}\left[\frac{17 \pi x}{4}\right]$$

Pick off the coefficients of the Sin[$\frac{(k\pi)x}{L}$] terms:

```
L
```

4

```
Clear[A, k]
A[k_] := Coefficient[riggedsinefit[x], Sin[(k π) x / L]];
coeffs = Table[A[k], {k, 1, n}]
```

{1.2771, 0.0004125, -0.112591, 0.000257171, 0.0703317, 0.00720654, -0.0191593, 0.00184198, 0.0319138, 0.000188565, 0.00454808, 0.00156063, 0.0182007, -0.00432241, 0.0171384, -0.00180089, -0.00760997, 0}

Now you're done because you can write down position[x,t]:

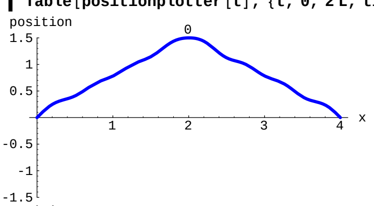
```
Clear[position, x, t]
position[x_, t_] = Sum[A[k] Cos[(k π) t / L] Sin[(k π) x / L], {k, 1, Length[coeffs]}]
```

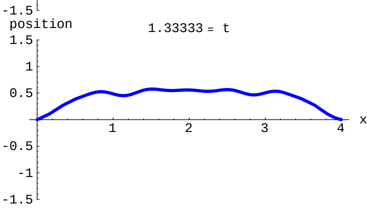
$$1.2771\, \text{Cos}\left[\frac{\pi t}{4}\right] \text{Sin}\left[\frac{\pi x}{4}\right] + 0.0004125\, \text{Cos}\left[\frac{\pi t}{2}\right] \text{Sin}\left[\frac{\pi x}{2}\right] - 0.112591\, \text{Cos}\left[\frac{3 \pi t}{4}\right] \text{Sin}\left[\frac{3 \pi x}{4}\right] + 0.000257171\, \text{Cos}[\pi t]\, \text{Sin}[\pi x] + 0.0703317\, \text{Cos}\left[\frac{5 \pi t}{4}\right] \text{Sin}\left[\frac{5 \pi x}{4}\right] + 0.00720654\, \text{Cos}\left[\frac{3 \pi t}{2}\right] \text{Sin}\left[\frac{3 \pi x}{2}\right] - 0.0191593\, \text{Cos}\left[\frac{7 \pi t}{4}\right] \text{Sin}\left[\frac{7 \pi x}{4}\right] + 0.00184198\, \text{Cos}[2 \pi t]\, \text{Sin}[2 \pi x] + 0.0319138\, \text{Cos}\left[\frac{9 \pi t}{4}\right] \text{Sin}\left[\frac{9 \pi x}{4}\right] + 0.000188565\, \text{Cos}\left[\frac{5 \pi t}{2}\right] \text{Sin}\left[\frac{5 \pi x}{2}\right] + 0.00454808\, \text{Cos}\left[\frac{11 \pi t}{4}\right] \text{Sin}\left[\frac{11 \pi x}{4}\right] + 0.00156063\, \text{Cos}[3 \pi t]\, \text{Sin}[3 \pi x] + 0.0182007\, \text{Cos}\left[\frac{13 \pi t}{4}\right] \text{Sin}\left[\frac{13 \pi x}{4}\right] - 0.00432241\, \text{Cos}\left[\frac{7 \pi t}{2}\right] \text{Sin}\left[\frac{7 \pi x}{2}\right] + 0.0171384\, \text{Cos}\left[\frac{15 \pi t}{4}\right] \text{Sin}\left[\frac{15 \pi x}{4}\right] - 0.00180089\, \text{Cos}[4 \pi t]\, \text{Sin}[4 \pi x] - 0.00760997\, \text{Cos}\left[\frac{17 \pi t}{4}\right] \text{Sin}\left[\frac{17 \pi x}{4}\right]$$

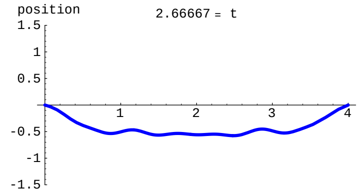
That's all there is to it.

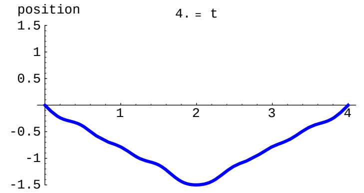
Here comes the movie:

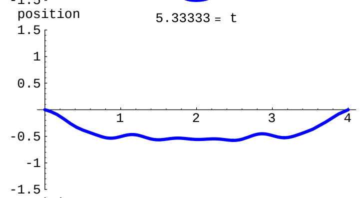
```
Clear[positionplotter]
positionplotter[t_] :=
 Plot[position[x, t], {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  PlotRange → {-1.5, 1.5}, AxesLabel → {"x", " position"},
  PlotLabel → N[t] "= t", AspectRatio → 1/2];
timejump = L/3;
Table[positionplotter[t], {t, 0, 2 L, timejump}]
```

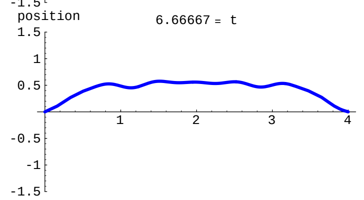


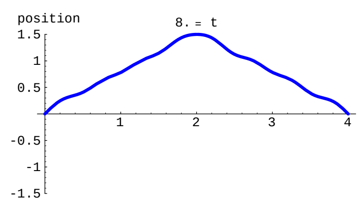












{- Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -}

Grab all the plots and animate.

The Cosines in the formula for position[x,t] are responsible for the periodic motion (waves) of the string:

```
position[x, t]
```

$$1.2771\, \text{Cos}\left[\frac{\pi t}{4}\right] \text{Sin}\left[\frac{\pi x}{4}\right] + 0.0004125\, \text{Cos}\left[\frac{\pi t}{2}\right] \text{Sin}\left[\frac{\pi x}{2}\right] - 0.112591\, \text{Cos}\left[\frac{3 \pi t}{4}\right] \text{Sin}\left[\frac{3 \pi x}{4}\right] + 0.000257171\, \text{Cos}[\pi t]\, \text{Sin}[\pi x] + 0.0703317\, \text{Cos}\left[\frac{5 \pi t}{4}\right] \text{Sin}\left[\frac{5 \pi x}{4}\right] + 0.00720654\, \text{Cos}\left[\frac{3 \pi t}{2}\right] \text{Sin}\left[\frac{3 \pi x}{2}\right] - 0.0191593\, \text{Cos}\left[\frac{7 \pi t}{4}\right] \text{Sin}\left[\frac{7 \pi x}{4}\right] + 0.00184198\, \text{Cos}[2 \pi t]\, \text{Sin}[2 \pi x] + 0.0319138\, \text{Cos}\left[\frac{9 \pi t}{4}\right] \text{Sin}\left[\frac{9 \pi x}{4}\right] + 0.000188565\, \text{Cos}\left[\frac{5 \pi t}{2}\right] \text{Sin}\left[\frac{5 \pi x}{2}\right] + 0.00454808\, \text{Cos}\left[\frac{11 \pi t}{4}\right] \text{Sin}\left[\frac{11 \pi x}{4}\right] + 0.00156063\, \text{Cos}[3 \pi t]\, \text{Sin}[3 \pi x] + 0.0182007\, \text{Cos}\left[\frac{13 \pi t}{4}\right] \text{Sin}\left[\frac{13 \pi x}{4}\right] - 0.00432241\, \text{Cos}\left[\frac{7 \pi t}{2}\right] \text{Sin}\left[\frac{7 \pi x}{2}\right] + 0.0171384\, \text{Cos}\left[\frac{15 \pi t}{4}\right] \text{Sin}\left[\frac{15 \pi x}{4}\right] - 0.00180089\, \text{Cos}[4 \pi t]\, \text{Sin}[4 \pi x] - 0.00760997\, \text{Cos}\left[\frac{17 \pi t}{4}\right] \text{Sin}\left[\frac{17 \pi x}{4}\right]$$

DE.09 The Heat Equation

$\partial_{\{x,2\}}$ temp[x, t] = ∂_t temp[x, t]

and the Wave Equation

$\partial_{\{x,2\}}$ position[x, t] = $\partial_{\{t,2\}}$ position[x, t]

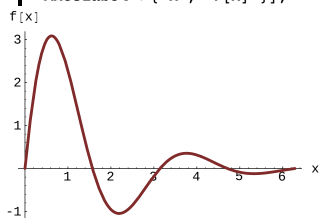
Give It a Try!

G.1) Pure Sine fits

□G.1.a.i)

Here's a function f[x] and its plot on [0,L] for L = 2 π:

```
Clear[f, x]
f[x_] = 5 E^(-Random[Real, {0.5,1}] x) Sin[2 x];
L = 2 π;
Plot[f[x], {x, 0, L}, PlotStyle → {{Thickness[0.01], Brown}},
 AxesLabel → {"x", "f[x]"}];
```



Here's a fast Fourier fit of this function:

```
Chop[ComplexExpand[FastFourierfit[f, L, 8, x]]]
```

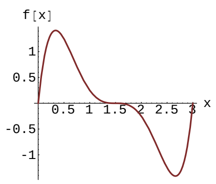
0.330685 + 0.740269 Cos[x] + 0.148714 Cos[2 x] − 0.422141 Cos[3 x] − 0.268166 Cos[4 x] − 0.18768 Cos[5 x] − 0.148714 Cos[6 x] − 0.130448 Cos[7 x] + 0.309308 Sin[x] + 1.10624 Sin[2 x] + 0.344108 Sin[3 x] + 0.103842 Sin[4 x] + 0.0430098 Sin[5 x] + 0.0199921 Sin[6 x] + 0.00820957 Sin[7 x]

Mixed Sines and Cosines.
How does the plot indicate why the fast Fourier fit of f[x] on [0,L] did not turn out to be a pure sine fit?

□G.1.a.ii)

Here's a function f[x] and its plot on [0,L] for L = 3

```
Clear[f, x]
f[x_] = x (x - 1.5)^3 (x - 3);
L = 3;
Plot[f[x], {x, 0, L},
 PlotStyle → {{Thickness[0.01], Brown}}, PlotRange -> All,
 AxesLabel → {"x", "f[x]"}];
```



Here's a fast Fourier fit of this function:

```
Chop[ComplexExpand[FastFourierfit[f, L, 12, x]]]
```

$0.901849 \mathrm{Sin}\left[\frac{2\pi x}{3}\right] + 0.670802 \mathrm{Sin}\left[\frac{4\pi x}{3}\right] + 0.229051 \mathrm{Sin}[2\pi x] + 0.100747 \mathrm{Sin}\left[\frac{8\pi x}{3}\right] + 0.052185 \mathrm{Sin}\left[\frac{10\pi x}{3}\right] + 0.0299988 \mathrm{Sin}[4\pi x] + 0.0184008 \mathrm{Sin}\left[\frac{14\pi x}{3}\right] + 0.011664 \mathrm{Sin}\left[\frac{16\pi x}{3}\right] + 0.00737094 \mathrm{Sin}[6\pi x] + 0.00436864 \mathrm{Sin}\left[\frac{20\pi x}{3}\right] + 0.00203906 \mathrm{Sin}\left[\frac{22\pi x}{3}\right]$

Pure Sine fit.
How does the plot indicate why the fast Fourier fit of f[x] on [0,L] did turn out to be a pure sine fit?

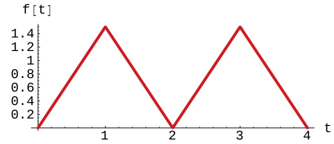
□G.1.a.iii)

Look at this:

```
Clear[f, t]
L = 4;
f[t_] = 3 Abs[0.5 t - Round[0.5 t]];
Chop[ComplexExpand[FastFourierfit[f, L, 12, t]]]
```

0.75 − 0.622008 Cos[π t] − 0.0833333 Cos[3 π t] − 0.0446582 Cos[5 π t]

```
Plot[f[t], {t, 0, L}, PlotStyle → {{Thickness[0.01], VenetianRed}},
 AxesLabel → {"t", "f[t]"}];
```



How does the plot explain why the fast Fourier fit of f[t] on [0,L] is not a pure sine fit?
Rig f[t] on [0,2 L] to get a pure sine fit of f[t] on [0,L]. Show off your work with a plot.

□G.1.a.iv)

Look at these:

```
Clear[f, t]
L = 2 Pi;
f[t_] = Sin[12 t];
Chop[ComplexExpand[FastFourierfit[f, L, 6, t]]]
Chop[ComplexExpand[FastFourierfit[f, L, 14, t]]]
Chop[ComplexExpand[FastFourierfit[f, L, 20, t]]]
Chop[ComplexExpand[FastFourierfit[f, L, 200, t]]]
```

0
1. Sin[12 t]
1. Sin[12 t]
1. Sin[12 t]

Got any idea why that happened?

□Tip:

If you're going to do a lot of plots, then you're doing too much.

The answer is staring at you from the screen.

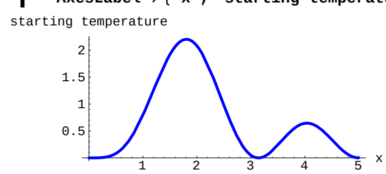
G.2) Sine fit and the heat equation; Sine fit and the wave equation

□G.2.a) Heat Equation

Start with a heated wire L units long with the temperature allowed to vary from position to position on the wire. Think of the wire as the interval 0 ≤ x ≤ L. At the start of the experiment, you instantly cool the ends at x = 0 and x = L and maintain these ends at temperature 0, and you take pains to guarantee that the rest of the wire is perfectly insulated.
At the start of this partricular experiment, the temperature of the wire at position x (for 0 ≤ x ≤ L = 5) is given by the following function startertemp[x]:

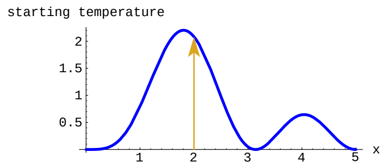
In this problem, the starting temperature of the wire at x is jointly proportional to the squares of the distance of x from the endpoints of the wire.

```
L = 5;
Clear[startertemp, x]
startertemp[x_] = 0.07 x^2 Sin[x]^2 (x - 5)^2;
starterplot = Plot[startertemp[x],
  {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  AxesLabel → {"x", "starting temperature"}];
```



To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] :=
 Arrow[{0, startertemp[x]}, Tail → {x, 0}, VectorColor -> Goldenrod];
Show[starterplot, pointer[2]];
```



Think of the interval [0,L] = [0, 5] as the wire.
The tip of the pointer tells you the starting temperature (at time t = 0) at the tail of the pointer.
Your problem here is to come up with the function temp[x,t] that estimates the temperature of the wire at position x at time t after the experiment begins.
This means that you are looking for a function temp[x,t] satisfying the heat equation

$\partial_{\{x,2\}}$ temp[x, t] = ∂_t temp[x, t]

with
-> temp[x,0] a good approximation of startertemp[x] and
-> temp[0,t] = startertemp[0] and temp[L ,t] = startertemp[L] for all t's.
Do it.
Make a movie.

□G.2.a.ii)

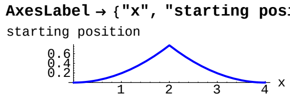
Run your movie again to get your opinion about the answer of the following question:
As time goes, on does any of the excess heat on the right warm up the left?

□G.2.b.i) Wave Equation

The ends of a guitar string are anchored at 0 and L on the x-axis and the string is pulled to an initial position and then allowed to vibrate on its own starting with initial velocity 0.

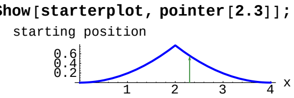
At the start of this particular experiment, the position of the wire at position x (for 0 ≤ x ≤ L = 5) is given by the following function starterposition[x]:

```
L = 4;
Clear[starterposition, x]
starterposition[x_] = 3.1 (0.25 x - Round[0.25 x])^2;
starterplot = Plot[starterposition[x], {x, 0, L},
  PlotStyle → {{Thickness[0.01], Blue}}, PlotRange → All,
  AxesLabel → {"x", "starting position"}];
```



Think of the curve as the the starting position of the guitar string.
To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] := Arrow[
  {0, starterposition[x]}, Tail → {x, 0}, VectorColor → CobaltGreen]
Show[starterplot, pointer[2.3]];
```



The tip of the pointer tells you the starting position (at time t = 0) at the tail of the pointer.
Your problem here is to come up with the function position[x,t] that estimates the position of the guitar string at position x on the x-axis at time t after the experiment begins.
This means that you are looking for a function position[x,t] satisfying the wave equation

$\partial_{\{x,2\}}$ position[x, t] = $\partial_{\{t,2\}}$ position[x, t]

with
-> position[x,0] a good approximation of starterposition[x] and
-> position[0,t] = starterposition[0] and position[L ,t] = starterposition[L] for all t's, and
-> ∂_t position[x, t] /. t → 0 = 0 for all x's
Do it.
Make a movie and explain what the movie dipicts.
Do you notice anything worth commenting on?

□ **Tip:**

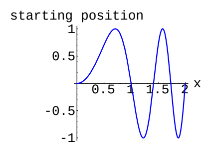
Run your t from 0 to 2 L to reveal one complete oscillation of this vibrating string.

□G.2.c.i)

The ends of a guitar string are anchored at 0 and L on the x-axis and the string is pulled to an initial position and then allowed to vibrate on its own starting with initial velocity 0.

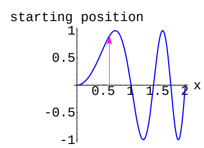
At the start of this particular experiment, the position of the wire at position x (for 0 ≤ x ≤ L = 2) is given by the following function starterposition[x]:

```
L = 2;
Clear[starterposition, x]
starterposition[x_] = Sin[π x^2];
starterplot = Plot[starterposition[x], {x, 0, L},
  PlotStyle → {{Thickness[0.01], Blue}}, PlotRange → All,
  AxesLabel → {"x", "starting position"}];
```



Think of the curve as the the starting position of the guitar string.
To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] :=
 Arrow[{0, starterposition[x]}, Tail → {x, 0}, VectorColor → Magenta]

Show[starterplot, pointer[0.6]];
```



The tip of the pointer tells you the starting position (at time t = 0) at the tail of the pointer.
Your problem here is to come up with the function position[x,t] that estimates the position of the guitar string at position x on the x-axis at time t after the experiment begins.
This means that you are looking for a function position[x,t] satisfying the wave equation

$\partial_{\{x,2\}}$ position[x, t] = $\partial_{\{t,2\}}$ position[x, t]

with
-> position[x,0] a good approximation of starterposition[x] and
-> position[0,t] = starterposition[0] and position[L ,t] = starterposition[L] for all t's, and
-> D[position[x,t],t]/.t->0 = 0 for all x's
Do it.
Make a movie and explain what the movie depicts.

□ **Tip:**

Run your t from 0 to 2 L to reveal one complete oscillation of this vibrating string.

□G.2.c.ii)

Run your movie again to get your opinion about the answer of the following question:
As time goes, on does any of the action on the right move over to the left or does it stay where it started?

□G.2.c.iii)

If you did a good job with your movies in parts b) and c), then you ran frames by plotting position[x,t] in increments of t running from 0 to 2 L.
The reason you did this was to plot one complete oscillation of the string.
In both problems, position[x,t] was given by

$$\text{position}[x, t] = \sum_{k=1}^{n} A[k]\,\text{Cos}[\tfrac{(k\pi)t}{L}]\,\text{Sin}[\tfrac{(k\pi)x}{L}]$$

Use this formula to try to explain why you are guaranteed that the the will be in its starting position when t = 0, 2 L, 4 L, . . .

□G.2.c.iv)

If you did a good job with your movies in parts b) and c), then you ran frames by plotting position[x,t] in increments of t running from 0 to 2 L.
The reason you did this was to plot one complete oscillation of the string.
In both problems, position[x,t] was given by

$$\text{position}[x, t] = \sum_{k=1}^{n} A[k]\,\text{Cos}[\tfrac{(k\pi)t}{L}]\,\text{Sin}[\tfrac{(k\pi)x}{L}]$$

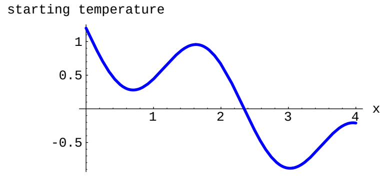
Use this formula to try to explaina short string can be expected to vibrate faster than a long string.

G.3) Different starting values on the left and right

□G.3.a) Heat Equation

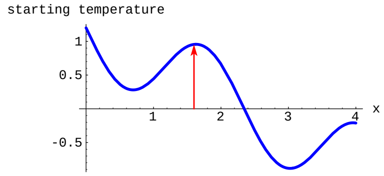
Start with a heated wire L units long with the temperature allowed to vary from position to position on the wire. Think of the wire as the interval $0 \le x \le L$. At the start of the experiment, you apply apparatus that maintains the temperatures at the end points of then wire. At the start of this particular experiment, the temperature of the wire at position x for $0 \le x \le L = 4$ is given by the following function startertemp[t]:

```
L = 4;
Clear[startertemp, x]
startertemp[x_] = 1.2 - 0.5 x - 0.6 Sin[2.7 x];
starterplot = Plot[startertemp[x], {x, 0, L},
  PlotStyle → {{Thickness[0.01], Blue}}, PlotRange → All,
  AxesLabel → {"x", "starting temperature"}];
```



To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] :=
 Arrow[{0, startertemp[x]}, Tail → {x, 0}, VectorColor → Red];
Show[starterplot, pointer[1.6]];
```



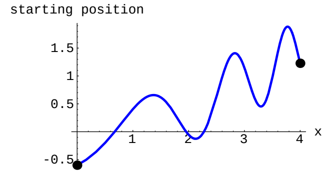
Think of the interval [0,L] = [0, 4] as the wire.
The tip of the pointer tells you the starting temperature (at time t = 0) at the tail of the pointer.
Your problem here is to come up with the function temp[x,t] that estimates the temperature of the wire at position x at time t after the experiment begins. This means that you are looking for a function temp[x,t] satisfying the heat equation
$\partial_{\{x,2\}}\text{temp}[x, t] = \partial_t \text{temp}[x, t]$
with
-> temp[x,0] a good approximation of startertemp[x]
and
-> temp[0,t] = startertemp[0] and temp[L ,t] = startertemp[L] for all t's.
Do it.
Make a movie.
Describe what the movie tells you.
Describe the steady state temperature of the wire?

□G.3.b.i)

The ends of a guitar string are anchored at the positions on the left and right shown below and the string is pulled to an initial position and then allowed to vibrate on its own starting with initial velocity 0.

At the start of this particular experiment, the position of the wire at position x (for $0 \le x \le L = 4$) is given by the following function starterposition[x]:

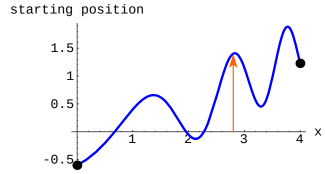
```
L = 4;
Clear[starterposition, x]
starterposition[x_] = -0.6 + 0.5 x + 0.6 Sin[x^2];
starterplot = Plot[starterposition[x],
  {x, 0, L}, PlotStyle → {{Thickness[0.01], Blue}},
  PlotRange → All, AxesLabel → {"x", "starting position"},
  Epilog → {{PointSize[0.04], Point[{0, starterposition[0]}]},
    {PointSize[0.04], Point[{L, starterposition[L]}]}}];
```



The black dots indicate the points at which the string is anchored.

Think of the curve as the the starting position of the guitar string.
To fully understand this plot, look at this:

```
Clear[pointer]
pointer[x_] := Arrow[{0, starterposition[x]},
  Tail → {x, 0}, VectorColor → CadmiumOrange];
Show[starterplot, pointer[2.8]];
```



The tip of the pointer tells you the starting position (at time t = 0) at the tail of the pointer.
Your problem here is to come up with the function position[x,t] that estimates the position of the guitar string at position x on the x-axis at time t after the experiment begins.
This means that you are looking for a function position[x,t] satisfying the wave equation
$\partial_{\{x,2\}}\text{position}[x, t] = \partial_{\{t,2\}}\text{position}[x, t]$
with
-> position[x,0] a good approximation of starterposition[x] and
-> position[0,t] = starterposition[0] and position[L ,t] = starterposition[L] for all t's, and
-> D[position[x,t],t]/.t->0 = 0 for all x's
Do it.

Make a movie and explain what the movie dipicts.
Do you notice anything worth commenting on?

Tip:

This problem wasn't addressed in the Basics or Tutorials.

Don't panic. The first idea you come up with will probably work.

G.3.b.ii)

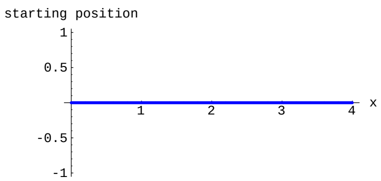
How does the wave equation explain why the little stunt with the line works?

G.4) The wave equation with starter position and starter velocity

G.4.a)

The ends of a guitar string are anchored at 0 and L on the x-axis .
At the start of this particular experiment, the position of the wire at position x (for $0 \le x \le L = 5$) is given by the following function starterposition[x]:

```
L = 4;
Clear[starterposition, x]
starterposition[x_] = 0;
starterplot = Plot[starterposition[x], {x, 0, L},
  PlotStyle → {{Thickness[0.01], Blue}}, PlotRange → All,
  AxesLabel → {"x", "starting position"}];
```



This is not an error.
But the string is given a starting vertical velocity - different at different x's:

for all positive integers k. This suggests that for a fixed time t, you can fit position[x,t] with a rigged Sine fit like this:

```
Clear[position, t, x, u, L]
n = 8;
position[x_, t_] = Sum[u[t, k] Sin[(k π) x / L], {k, 1, n}]
```

$\text{Sin}[\frac{\pi x}{L}]\, u[t,1] + \text{Sin}[\frac{2\pi x}{L}]\, u[t,2] + \text{Sin}[\frac{3\pi x}{L}]\, u[t,3] + \text{Sin}[\frac{4\pi x}{L}]\, u[t,4] + \text{Sin}[\frac{5\pi x}{L}]\, u[t,5] + \text{Sin}[\frac{6\pi x}{L}]\, u[t,6] + \text{Sin}[\frac{7\pi x}{L}]\, u[t,7] + \text{Sin}[\frac{8\pi x}{L}]\, u[t,8]$

There is nothing magic about setting n = 8.

The Fourier fit coefficients u[t,k] depend on t as well as k because you expect a different rigged sine fit at different times t.

The wave equation says

$\partial_{\{x,2\}}\text{position}[x, t] = \partial_{\{t,2\}}\text{position}[x, t].$

Plug

$\text{position}[x, t] = \sum_{k=1}^{n} u[t, k]\, \text{Sin}[\frac{(k\pi)x}{L}]$

into the heat equation and see that

$\sum_{k=1}^{n} u[t, k]\, (\frac{k\pi}{L})^2\, (-\text{Sin}[\frac{(k\pi)x}{L}]) = \sum_{k=1}^{n} D[u[t, k], \{t, 2\}]\, \text{Sin}[\frac{(k\pi)x}{L}].$

You can make this happen by setting

$D[u[t,k],\{t,2\}] = -(\frac{k\pi}{L})^2\, u[t, k].$

This gives you

$\partial_{\{t,2\}}u[t, k] + (\frac{k\pi}{L})^2\, u[t, k] = 0$

```
Clear[startervelocity, x]
startervelocity[x_] = 0.3 x (4 - x)
```

0.3 (4 - x) x

Your problem here is to come up with the function position[x,t] that estimates the position of the guitar string at position x on the x-axis at time t after the experiment begins.
This means that you are looking for a function position[x,t] satisfying the wave equation

$\partial_{\{x,2\}}\text{position}[x, t] = \partial_{\{t,2\}}\text{position}[x, t]$

with
-> D[position[x,t],t]/.t->0 a good approximation of startervelocity[x]
and
-> position[0,t] = starterposition[0] and position[L ,t] = starterposition[L] for all t's, and
-> position[x,0] = 0 for all x's.
Make a movie.

Heavy Tip:

Engineering studies have shown that after the appropriate unit adjustments are made, the function position[x,t] satisfies the wave equation

$\partial_{\{x,2\}}\text{position}[x, t] = \partial_{\{t,2\}}\text{position}[x, t]$

with

-> D[position[x,t],t]/.t->0 = startervelocity[x]

-> position[0,t] = 0 and position[L ,t] = 0 for all t's

because the ends of the guitar string are attached at the ends and

-> position[x,t] = 0 for t = 0

The key is the boundary conditions

position[0,t] = 0 and position[L ,t] = 0.

These match up well with the fact that

$\text{Sin}[\frac{k\pi x}{L}] = 0$ for x = 0 and x = L

This is a big break in your favor because this is the undamped unforced oscillator:

$u[t, k] = A[k]\, \text{Cos}[\frac{(k\pi)t}{L}] + B[k]\, \text{Sin}[\frac{(k\pi)t}{L}]$

Here the real constants A[k] and B[k] have yet to be determined.

You get a different constant for each k.

Now look at the condition

position[x,0] = 0

Because

$\text{position}[x, t] = \sum_{k=1}^{n} u[t, k]\, \text{Sin}[\frac{(k\pi)x}{L}],$

you can achieve

position[x,0] = 0

by insisting that:

u[0,k]=0

This tells you to set A[k] = 0:

Substitute

u[t,k] = B[k] Sin[(k Pi/L) t]

into position[x,t]

to get

$\text{position}[x, t] = \sum_{k=1}^{n} B[k]\, \text{Sin}[\frac{k\pi t}{L}]\, \text{Sin}[\frac{(k\pi)x}{L}],$

Look at what happens for t = 0:

$$\text{position}[x, t] = \sum_{k=1}^{n} B[k]\, 0\, \text{Sin}[\tfrac{(k\pi)x}{L}] = 0.$$

Good.

Now the question remaining for you is how to use a rigged pure sine fit of the starting velocity to set the constants B[k].

Here is a pregnant clue:

To get the initial velocity,differentiate position[x,t] with respect to t and get

$$\sum_{k=1}^{n} B[k]\, \tfrac{k\pi}{L}\, \text{Cos}[\tfrac{k\pi t}{L}]\, \text{Sin}[\tfrac{(k\pi)x}{L}].$$

and set t = 0; to get

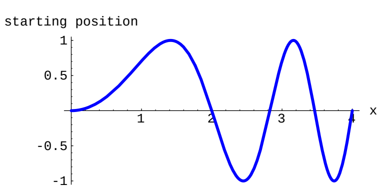
approxstartervelocity[x]

$$\sum_{k=1}^{n} B[k]\, \tfrac{k\pi}{L}\, \text{Sin}[\tfrac{(k\pi)x}{L}]$$

□G.4.b)

The ends of a guitar string are anchored at 0 and L on the x-axis .
At the start of this particular experiment, the position of the wire at position x (for 0 ≤ x ≤ L = 4) is given by the following function startertemp[x]:

```
L = 4;
Clear[starterposition, x]
starterposition[x_] = Sin[π (x/2)^2];
starterplot = Plot[starterposition[x], {x, 0, L},
  PlotStyle → {{Thickness[0.01], Blue}}, PlotRange → All,
  AxesLabel → {"x", "starting position"}];
```



In addition, the string is given a starting velocity - different at different x's:

```
Clear[startervelocity, x]
startervelocity[x_] = 0.3 x (4 - x)
```

0.3 (4 − x) x

Your problem here is to come up with the function position[x,t] that estimates the position of the guitar string at position x on the x-axis at time t after the experiment begins.
This means that you are looking for a function position[x,t] satisfying the wave equation

$\partial_{\{x,2\}}$ position[x, t] = $\partial_{\{t,2\}}$ position[x, t]

with
-> position[x,0] a good approximation of starterposition[x]
-> D[position[x,t],t]/.t->0 a good approximation of startervelocity[x]
and
-> position[0,t] = starterposition[0] and position[L ,t] = starterposition[L] for all t's.
Make a movie.

□Another Heavy Tip:

You can recycle your answer to part a) by using the old technique of dividing and conquering.

Here's how it goes:

□Step 1:

Come up with a function position1[x,t] satisfying

$\partial_{\{x,2\}}$ position1[x, t] = $\partial_{\{t,2\}}$ position1[x, t]

with

-> position1[x,0] a good approximation of starterposition[x]

-> D[position1[x,t],t]/.t->0 = 0 for 0 ≤ x ≤ L

-> position1[0,t] = starterposition[0] and position1[L ,t] = starterposition[L] for all t's.

You have done problems like this in G.3) above.

□Step 2:

Come up with a function position2[x,t] satisfying

$\partial_{\{x,2\}}$ position2[x, t] = $\partial_{\{t,2\}}$ position2[x, t]

with

-> position2[x,0] = 0 for 0 ≤ x ≤ L

-> D[position2[x,t],t]/.t->0 a good approximation of startervelocity[x]

-> position2[0,t] = starterposition[0] and position2[L ,t] = starterposition[L] for all t's.

You did this very problem in part a) immediately above.

□Step 3:

Go with position[x,t] = position1[x,t] + position2[x,t].

This will give you
$\partial_{\{x,2\}}$ position[x, t] = $\partial_{\{t,2\}}$ position[x, t]
with

→ position[x,0]

= position1[x,0] + position2[x,0]

= a good approximation of starterposition[x] + 0

→ D[position[x,t],t]/.t->0

= D[position1[x,t],t]/.t->0 + D[position2[x,t],t]/.t->0

= 0 + a good approximation of startervelocity[x]

→ position[0,t] = 0 + 0 = starterposition[0] and

position[L ,t] = 0 + 0 = starterposition[L] for all t's.

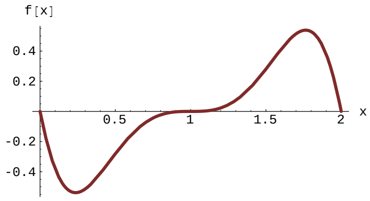
G.5) Pure Cosine fits

□G.5.a.i)

Here's a function f[x] and its plot on [0,L] for L =2:

```
Clear[f, x]
f[x_] = 3 x Sin[(1 - x)^3] (x - 2);
L = 2;
Plot[f[x], {x, 0, L},
 PlotStyle → {{Thickness[0.01], Brown}}, PlotRange -> All,
 AxesLabel → {"x", "f[x]"}];
```

Rig f[x] on [0, 2 L] so that Fourier fits of the resulting function on [0,2L] exhibit only Cosines and possibly a constant?



You can tell at a glance that Fourier fits on [0,L] are all pure sine fits:

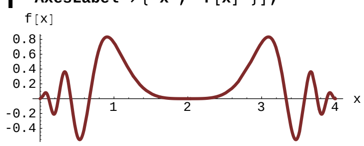
```
n = Random[Integer, {4, 16}];
Chop[ComplexExpand[FastFourierfit[f, L, n, x]]]
```

$$-0.348828\,\text{Sin}[\pi x] - 0.253942\,\text{Sin}[2\pi x] - 0.0800457\,\text{Sin}[3\pi x] - 0.0319527\,\text{Sin}[4\pi x] - 0.0153182\,\text{Sin}[5\pi x] - 0.00840115\,\text{Sin}[6\pi x] - 0.00505942\,\text{Sin}[7\pi x] - 0.00324873\,\text{Sin}[8\pi x] - 0.00217574\,\text{Sin}[9\pi x] - 0.00149122\,\text{Sin}[10\pi x] - 0.00102486\,\text{Sin}[11\pi x] - 0.000686094\,\text{Sin}[12\pi x] - 0.000422694\,\text{Sin}[13\pi x] - 0.000201747\,\text{Sin}[14\pi x]$$

Rerun a couple of times.

Now look at this plot of a function f[x] on [0,L] for L = 4:

```
Clear[f, x]
f[x_] = 0.3 x Sin[(2 - x)^4 ] (4 - x);
L = 4;
Plot[f[x], {x, 0, L},
 PlotStyle → {{Thickness[0.01], Brown}}, PlotRange -> All,
 AxesLabel → {"x", "f[x]"}];
```



Check out some Fourier fits on [0,L]

```
n = Random[Integer, {4, 16}];
Chop[ComplexExpand[FastFourierfit[f, L, n, x]]]
```

$$0.144927 - 0.0841849\,\text{Cos}\left[\frac{\pi x}{2}\right] - 0.340755\,\text{Cos}[\pi x] + 0.0313064\,\text{Cos}\left[\frac{3\pi x}{2}\right] + 0.241748\,\text{Cos}[2\pi x] + 0.0762827\,\text{Cos}\left[\frac{5\pi x}{2}\right] - 0.0793501\,\text{Cos}[3\pi x] - 0.0660752\,\text{Cos}\left[\frac{7\pi x}{2}\right] + 0.0334295\,\text{Cos}[4\pi x]$$

Rerun a couple of times.

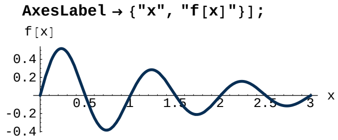
No Sines. Just Cosines and a constant.

How does the shape of the plot for f[x] tip you off in advance that you will get just Cosines and a constant?

□G.5.a.ii)

Here's a function f[x] and its plot on [0,L] for L = 3:

```
Clear[f, x]
f[x_] = 0.6 E^(-0.6 x) Sin[2 π x];
L = 3;
Plot[f[x], {x, 0, L},
 PlotStyle → {{Thickness[0.01], Indigo}}, PlotRange -> All,
 AxesLabel → {"x", "f[x]"}];
```



If you know what you're doing, then you can telljust by looking at the plot that Fourier fits of this function on [0,L] will exhibit both Sine and Cosine terms.
Check this out:

```
n = Random[Integer, {4, 16}];
Chop[ComplexExpand[FastFourierfit[f, L, n, x]]]
```

$$0.0246842 + 0.0555739\,\text{Cos}\left[\frac{2\pi x}{3}\right] + 0.086205\,\text{Cos}\left[\frac{4\pi x}{3}\right] + 0.00983398\,\text{Cos}[2\pi x] - 0.0658225\,\text{Cos}\left[\frac{8\pi x}{3}\right] - 0.032776\,\text{Cos}\left[\frac{10\pi x}{3}\right] - 0.0213481\,\text{Cos}[4\pi x] - 0.0159766\,\text{Cos}\left[\frac{14\pi x}{3}\right] - 0.0130848\,\text{Cos}\left[\frac{16\pi x}{3}\right] - 0.0114677\,\text{Cos}[6\pi x] - 0.0106338\,\text{Cos}\left[\frac{20\pi x}{3}\right] + 0.00416591\,\text{Sin}\left[\frac{2\pi x}{3}\right] + 0.0201758\,\text{Sin}\left[\frac{4\pi x}{3}\right] + 0.277575\,\text{Sin}[2\pi x] + 0.0206015\,\text{Sin}\left[\frac{8\pi x}{3}\right] + 0.00518941\,\text{Sin}\left[\frac{10\pi x}{3}\right] + 0.00216989\,\text{Sin}[4\pi x] + 0.00111077\,\text{Sin}\left[\frac{14\pi x}{3}\right] + 0.000617409\,\text{Sin}\left[\frac{16\pi x}{3}\right] + 0.000338796\,\text{Sin}[6\pi x] + 0.000151704\,\text{Sin}\left[\frac{20\pi x}{3}\right]$$

Rerun a couple of times.

Now you get the chance to invent some mathematics.