Use the Fourier Series to Approximate Discontinuous Functions and to Interpret Music

Introduction

OBJECTIVE: Use *Mathematica* to calculate Fourier series and to build even and odd Fourier representations of selected function.

In signal processing and communications, it is necessary to construct periodic functions, some with discontinuities. The Fourier series provides us with a tool to analyze such functions. One very important type of signal that you probably receive every day is music. We can use the Fourier series to build mathematical models of musical tones, to look at their graphs, and even to play back the signal to hear how close our model is to the real thing. We usually call a device that can do this a synthesizer.

As you have probably noticed, the computations involved in arriving at a Fourier series approximation to a function can be tedious. Fortunately, the computer can perform such computations for us, enabling you to not only get the results but to visualize the results

It should be noted that *Mathematica* has a package that allows you to merely call upon the Fourier Series. The package is called by executing <<Calculus`FourierTransform` The specific commands for getting the Fourier Series expansions are different in older versions. You can get details on this from the Help window, by looking under Add-Ons and then Calculus packages and then the FourierTransform. This module calls upon the FourierTransform package at times, but it also uses the Integrate commands to compute the Fourier coefficients, so that it is less of a "black box." If you wish to use this package for a discontinuous function, that function must defined in a special way, for example, using the Mod function or the UnitStep function.

▪ Technology Guidelines

NOTE: If you have just finished a module, restart *Mathematica* or close the *Kernel* before executing a new module.
TO OPEN CELLS, put your cursor on the right cell bracket and double click.
TO STOP AN EXECUTION
Select the *Kernel* pull-down menu and click on *Abort Evaluation.*
ORDER OF EXECUTION
Execute cells in the order given. Do not skip any Input cells within a given notebook.
SAVING NOTEBOOKS
You can save anytime to any directory you choose, and it is wise to save often.
However, before you do your final save, it is a good idea to delete all your output by selecting the
Delete All Output selection under the *Kernel* pull-down menu.
EXPERIENCING MAJOR PROBLEMS
Save if appropriate, then shut down *Mathematica* and start it up again.

Part I - Fourier Series Approximations for Non-Periodic Functions

Section 11.11

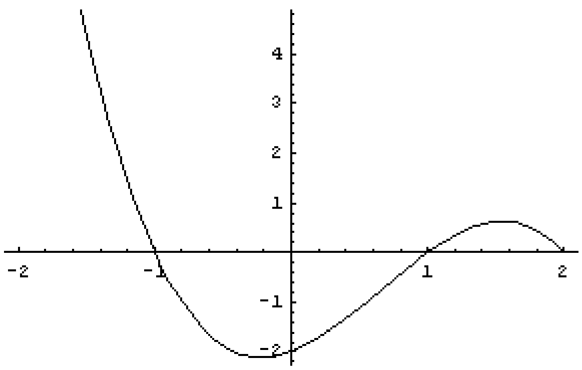
The following is an example of how you can use a Fourier series to approximate a continuous function. If you have not done so already, you should read Section 11.11 in your text before proceeding. We define the coefficients for the sine and cosine functions as they are defined in your text. Because of the way that *a[n]* and *b[n]* are stored, if you decide to enter a new function, you must **be sure to clear the *a* and *b* values first**.

The function we begim looking at is *f(x)* = (1 - x)(x+1)(x-2) and we will find its Fourier series approximation over the interval from x = -2 to +2. We will begin by defining the function and looking at its plot.

In[1]:=

```
Off[General::spell]

Off[General::spell1]

Clear[a, b, f, x, length, foursine, fourcos, a

f[x_] := (1 - x) (x + 1) (x - 2)
```

```
length = 2;

Plot[f[x], {x, -length, length}];
```



Now we compute the Fourier coefficients using the formulations in your text.

"> About *Mathematica*

In[7]:=

```
b[n_] :=
 b[n] = 1/length Integrate[Sin[n π/2 x] f[x],
    {x, -length, length}]

a[n_] :=
 a[n] = 1/length Integrate[Cos[n π/2  x] f[x],
    {x, -length, length}]
```

Next, we will put these coefficients into the series formulas and go out to the n = 10 term in both the sine and cosine components.

In[9]:=

```
foursine[x_, 10] = Sum[b[j] Sin[j π/2 x], {j,

fourcos[x_, 10] = a[0]/2 + Sum[a[j ] Cos[j π/2

all[x_, 10] = foursine[x, 10] + fourcos[x, 10]
```

Out[11]=

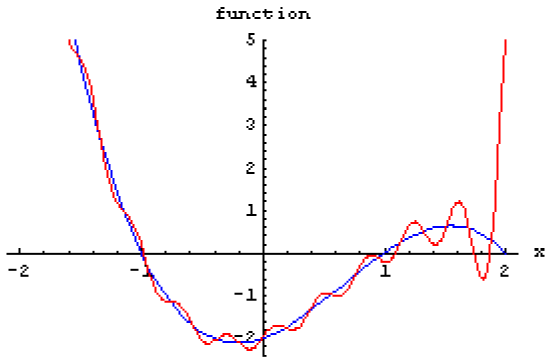
$$\frac{2}{3} - \frac{32\,\text{Cos}\left[\frac{\pi x}{2}\right]}{\pi^2} + \frac{8\,\text{Cos}[\pi x]}{\pi^2} - \frac{32\,\text{Cos}\left[\frac{3\pi x}{2}\right]}{9\pi^2} + \frac{2\,\text{Cos}[2\pi x]}{\pi^2} - \frac{32\,\text{Cos}\left[\frac{5\pi x}{2}\right]}{25\pi^2} + \frac{8\,\text{Cos}[3\pi x]}{9\pi^2} - \frac{32\,\text{Cos}\left[\frac{7\pi x}{2}\right]}{49\pi^2} + \frac{\text{Cos}[4\pi x]}{2\pi^2} - \frac{32\,\text{Cos}\left[\frac{9\pi x}{2}\right]}{81\pi^2} + \frac{8\,\text{Cos}[5\pi x]}{25\pi^2} - \frac{12\,(-8+\pi^2)\,\text{Sin}\left[\frac{\pi x}{2}\right]}{\pi^3} + \frac{6\,(-2+\pi^2)\,\text{Sin}[\pi x]}{\pi^3} - \frac{4\,(-8+9\pi^2)\,\text{Sin}\left[\frac{3\pi x}{2}\right]}{9\pi^3} + \frac{(-3+6\pi^2)\,\text{Sin}[2\pi x]}{2\pi^3} - \frac{12\,(-8+25\pi^2)\,\text{Sin}\left[\frac{5\pi x}{2}\right]}{125\pi^3} + \frac{2\,(-2+9\pi^2)\,\text{Sin}[3\pi x]}{9\pi^3} - \frac{12\,(-8+49\pi^2)\,\text{Sin}\left[\frac{7\pi x}{2}\right]}{343\pi^3} + \frac{3\,(-1+8\pi^2)\,\text{Sin}[4\pi x]}{16\pi^3} - \frac{4\,(-8+81\pi^2)\,\text{Sin}\left[\frac{9\pi x}{2}\right]}{243\pi^3} + \frac{6\,(-2+25\pi^2)\,\text{Sin}[5\pi x]}{125\pi^3}$$

I expect that you are glad that you did not have to calculate that by hand! Now let's look at a plot of the function together with its Fourier series approximation.

In[12]:=

```
Plot[{f[x], Evaluate[all[x, 10]]}, {x, -lengt
 PlotStyle -> {RGBColor[0, 0, 1], RGBColor[1,
 AxesLabel -> {"x", "function"}];

Print[
 "The function is plotted in blue and its F
   series approximation over the interval fr
 length, " to x = ", length, " is plotted in
```

```
Print[
 "The Fourier series approximation out to n
 N[all[x, 10]] // ComplexExpand]
```



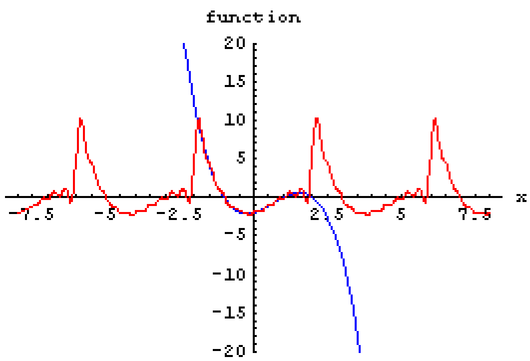

```
The function is plotted in blue and
   its Fourier series approximation
   over the interval from x = -
 2 to x = 2 is plotted in red.


The Fourier series
   approximation out to n = 10 is:
 0.666667 - 3.24228 Cos[1.5708 x] +
  0.810569 Cos[3.14159 x] -
  0.360253 Cos[4.71239 x] +
  0.202642 Cos[6.28319 x] -
  0.129691 Cos[7.85398 x] +
  0.0900633 Cos[9.42478 x] -
  0.0661689 Cos[10.9956 x] +
  0.0506606 Cos[12.5664 x] -
  0.0400281 Cos[14.1372 x] +
  0.0324228 Cos[15.708 x] -
  0.723571 Sin[1.5708 x] +
  1.52284 Sin[3.14159 x] -
  1.15857 Sin[4.71239 x] +
  0.906552 Sin[6.28319 x] -
  0.739175 Sin[7.85398 x] +
  0.622286 Sin[9.42478 x] -
  0.536647 Sin[10.9956 x] +
  0.471418 Sin[12.5664 x] -
  0.420166 Sin[14.1372 x] +
  0.378876 Sin[15.708 x]
```

The ComplexExpand command in the last input expression shows the series in real form.

What happens if you extend the plot out further? You will notice that the Fourier Series repeats the pattern of the function over the interval [-2, 2], whereas the polynomial does its own thing.

In[15]:=

```
Plot[{f[x], Evaluate[all[x, 10]]}, {x, -4 leng
 PlotRange -> {-20, 20},
 PlotStyle -> {RGBColor[0, 0, 1], RGBColor[1,
 AxesLabel -> {"x", "function"}];
```



The Fourier series gives a reasonable approximation to the function over the interval [-2,2], but outside that interval, all bets are off. The extended Fourier series is called the periodic extension of the function.

▪ Using the Fourier Transform Package

We could have arrived at the Fourier trigonometric expansion for our function by loading a special calculus package.

In[16]:=

```
<< Calculus`FourierTransform`
```

The default periodic length in this package is 1, so, when we call upon the trig series function, we will need to set our period at something other than 1. To do this, we change the second FourierParameter to 1/period. In this case, the width of the period is 4 or (2 length). The "10" specifies how many terms we want.

In[17]:=

```
fastfs[x_] = FourierTrigSeries[f[x], x, 10,
  FourierParameters → {1, 1/(2 length)}]
```

Out[17]=

$$\frac{1}{4}\left(\frac{8}{3}-\frac{128\,\mathrm{Cos}\left[\frac{\pi x}{2}\right]}{\pi^2}+\frac{32\,\mathrm{Cos}[\pi x]}{\pi^2}-\frac{128\,\mathrm{Cos}\left[\frac{3\pi x}{2}\right]}{9\pi^2}+\frac{8\,\mathrm{Cos}[2\pi x]}{\pi^2}-\frac{128\,\mathrm{Cos}\left[\frac{5\pi x}{2}\right]}{25\pi^2}+\frac{32\,\mathrm{Cos}[3\pi x]}{9\pi^2}-\frac{128\,\mathrm{Cos}\left[\frac{7\pi x}{2}\right]}{49\pi^2}+\frac{2\,\mathrm{Cos}[4\pi x]}{\pi^2}-\frac{128\,\mathrm{Cos}\left[\frac{9\pi x}{2}\right]}{81\pi^2}+\frac{32\,\mathrm{Cos}[5\pi x]}{25\pi^2}-\frac{48(-8+\pi^2)\,\mathrm{Sin}\left[\frac{\pi x}{2}\right]}{\pi^3}+\frac{24(-2+\pi^2)\,\mathrm{Sin}[\pi x]}{\pi^3}-\frac{16(-8+9\pi^2)\,\mathrm{Sin}\left[\frac{3\pi x}{2}\right]}{9\pi^3}+\frac{2(-3+6\pi^2)\,\mathrm{Sin}[2\pi x]}{\pi^3}-\frac{48(-8+25\pi^2)\,\mathrm{Sin}\left[\frac{5\pi x}{2}\right]}{125\pi^3}+\frac{8(-2+9\pi^2)\,\mathrm{Sin}[3\pi x]}{9\pi^3}-\frac{48(-8+49\pi^2)\,\mathrm{Sin}\left[\frac{7\pi x}{2}\right]}{343\pi^3}+\frac{3(-1+8\pi^2)\,\mathrm{Sin}[4\pi x]}{4\pi^3}-\frac{16(-8+81\pi^2)\,\mathrm{Sin}\left[\frac{9\pi x}{2}\right]}{243\pi^3}+\frac{24(-2+25\pi^2)\,\mathrm{Sin}[5\pi x]}{125\pi^3}\right)$$

To verify that this agrees with what we had previously found, we will look at the numerical approximation and graph this along with our original function.

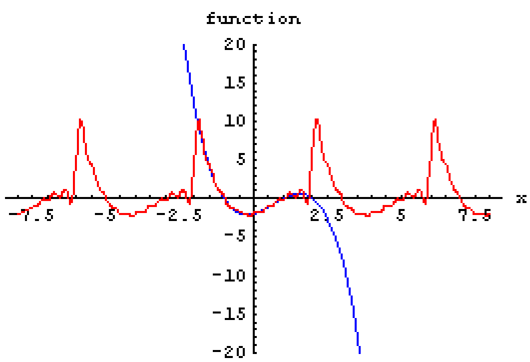
In[18]:=

```
fastfs[x] // N // Simplify
```

```
Plot[{f[x], fastfs[x]}, {x, -4 length, 4 lengt]
 PlotRange -> {-20, 20},
 PlotStyle -> {RGBColor[0, 0, 1], RGBColor[1,
 AxesLabel -> {"x", "function"}];
```

Out[18]=

```
0.666667 - 3.24228 Cos[1.5708 x] +
 0.810569 Cos[3.14159 x] -
 0.360253 Cos[4.71239 x] +
 0.202642 Cos[6.28319 x] -
 0.129691 Cos[7.85398 x] +
 0.0900633 Cos[9.42478 x] -
 0.0661689 Cos[10.9956 x] +
 0.0506606 Cos[12.5664 x] -
 0.0400281 Cos[14.1372 x] +
 0.0324228 Cos[15.708 x] -
 0.723571 Sin[1.5708 x] +
 1.52284 Sin[3.14159 x] -
 1.15857 Sin[4.71239 x] +
 0.906552 Sin[6.28319 x] -
 0.739175 Sin[7.85398 x] +
 0.622286 Sin[9.42478 x] -
 0.536647 Sin[10.9956 x] +
 0.471418 Sin[12.5664 x] -
 0.420166 Sin[14.1372 x] +
 0.378876 Sin[15.708 x]
```

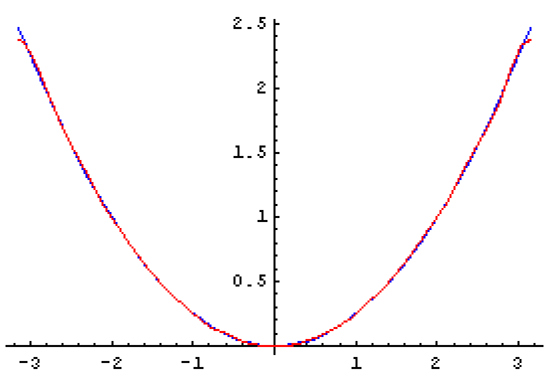


You Try It - Part I

Find the Fourier series for the function $\frac{x^2}{4}$ over the interval from -π to π. The following will get you started.

In[20]:=

```
Clear[a, b, f, x, length, foursine, fourcos, a

f[x_] := x^2 / 4

length = π;

b[n_] :=
 b[n] =
  1 / length Integrate[Sin[n x] f[x], {x, -leng

a[n_] :=
 a[n] = 1 / length Integrate[Cos[n x] f[x], {x,

foursine[x_, 10] = Sum[b[j] Sin[j x], {j, 1, 10

fourcos[x_, 10] = a[0] / 2 + Sum[a[j] Cos[j x], {

all[x_, n_] = foursine[x, 10] + fourcos[x, 10];

Plot[{f[x], Evaluate[all[x, 10]]}, {x, -lengt
 PlotStyle -> {RGBColor[0, 0, 1], RGBColor[1,

Print[
 "The function is plotted in blue and its F
   series approximation over the interval fr
 length, " to x = ", length, " is plotted in

Print[
 "The Fourier series approximation out to n
 all[x, 10]]
```



```
The function is plotted in blue and
   its Fourier series approximation
   over the interval from x = -
 π to x = π is plotted in red.
```

```
The Fourier series
   approximation out to n = 10 is:
```

$$\frac{\pi^2}{12} - \text{Cos}[x] + \frac{1}{4}\text{Cos}[2x] - \frac{1}{9}\text{Cos}[3x] + \frac{1}{16}\text{Cos}[4x] - \frac{1}{25}\text{Cos}[5x] + \frac{1}{36}\text{Cos}[6x] - \frac{1}{49}\text{Cos}[7x] + \frac{1}{64}\text{Cos}[8x] - \frac{1}{81}\text{Cos}[9x] + \frac{1}{100}\text{Cos}[10x]$$

Use this result to verify that the series that converges to $\frac{\pi^2}{6}$ is $1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \ldots$. What should you let x equal in your f[x] and in your Fourier series to get the desired result?

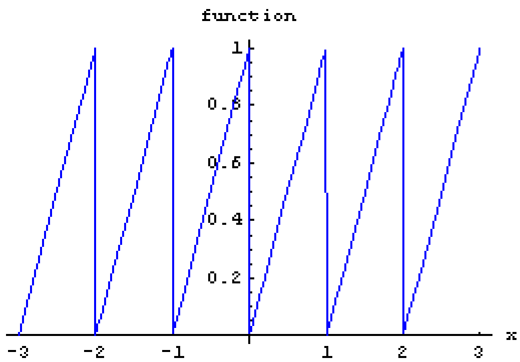
Can you come up with the sum of any other infinite series by this method?

Part II - Fourier Coefficients for the Sawtooth Function

Consider the Sawtooth Function that is essentially the line y = x over the interval from 0 to 1, and then that segment just gets repeated over and over. We can draw this by defining it using the **Mod [x, 1]** function. This maps every *x* onto the interval between 0 and 1. For example, if *x* is between 0 and 1, the function will be *x*. If *x* is between 1 and two, the function will be x-1, etc. Let's look at a graph of this function

In[31]:=

```
Clear[a, b, x, n, f, g]

g[x_] := Mod[x, 1]

Plot[g[x], {x, -3, 3}, PlotStyle -> RGBColor[0
 AxesLabel -> {"x", "function"}];
```

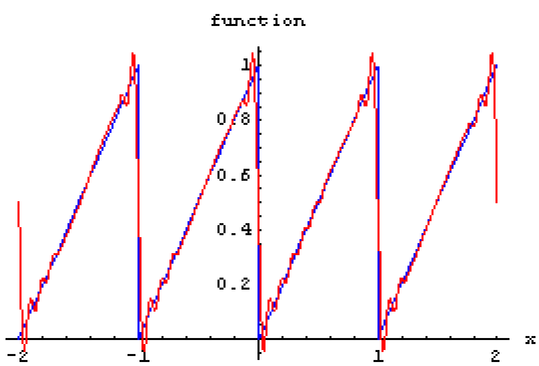


In order to compute Fourier coefficients for this function, we need to focus on the function over the interval from -1 to +1. The function is (x+1) when x goes from -1 to 0 and then becomes x when x goes from 0 to +1. Here we will put all our commands together and show the results at the end.

In[34]:=

```
Clear[a, b, x, length, foursine, fourcos, all]

length = 1;

b[n_] :=
 b[n] =
  1/length
   (Integrate[Sin[n Pi / length x] (x + 1), {x, ·
Integrate[Sin[n Pi / length x] x, {x, 0, length]

a[n_] :=
 a[n] =
  1/length
   (Integrate[Cos[n Pi / length x] (x + 1), {x, ·
     Integrate[Cos[n Pi / length x] x, {x, 0, le

foursine[x_, 20] = Sum[b[j] Sin[j Pi / length x]

fourcos[x_, 20] =
 a[0] / 2 + Sum[a[j] Cos[j Pi / length x], {j, 1,

all[x_, 20] = foursine[x, 20] + fourcos[x, 20];
```

```
Plot[{g[x], Evaluate[all[x, 20]]}, {x, -2 leng
 PlotStyle -> {RGBColor[0, 0, 1], RGBColor[1,
 AxesLabel -> {x, function}];

Print[
 "The function is plotted in blue and its F
   series approximation over the interval fr
 length, " to x = ", length, " is plotted in

Print[
 "The Fourier series approximation out to n
 all[x, 20]]
```




```
The function is plotted in blue and
   its Fourier series approximation
   over the interval from x = -
 1 to x = 1 is plotted in red.


The Fourier series
   approximation out to n = 20 is:
```

$$\frac{1}{2} - \frac{\text{Sin}[2\pi x]}{\pi} - \frac{\text{Sin}[4\pi x]}{2\pi} - \frac{\text{Sin}[6\pi x]}{3\pi} - \frac{\text{Sin}[8\pi x]}{4\pi} - \frac{\text{Sin}[10\pi x]}{5\pi} - \frac{\text{Sin}[12\pi x]}{6\pi} - \frac{\text{Sin}[14\pi x]}{7\pi} - \frac{\text{Sin}[16\pi x]}{8\pi} - \frac{\text{Sin}[18\pi x]}{9\pi} - \frac{\text{Sin}[20\pi x]}{10\pi}$$

Since this is neither an even nor an odd function, we need both the sine and cosine series in order to fit the function. However, you should note that there is only one nonzero term in the Fourier cosine series.

You should notice what is referred to as Gibb's phenomena. That refers to the poor (wiggly) fit at the points of discontinuity. There are far fewer wiggles over the continuous sections. This phenomena would occur no matter how many terms we extended our series.

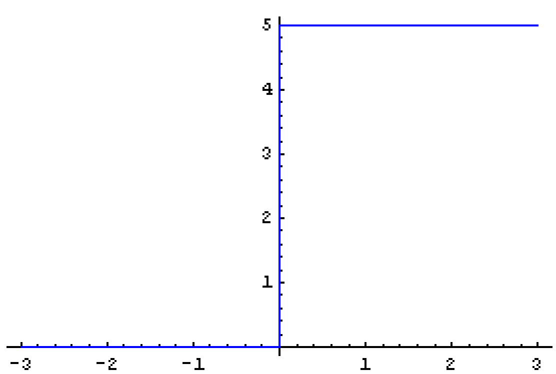
You Try It - Part II

Suppose you have a step function so that it takes on the value 0 from -3 to 0 and then jumps up to 5 when *x* is between 0 and 3. First we will define the function and look at its graph.

"> About *Mathematica*

In[44]:=

```
Clear[g, x]

length = 3;

g[x_] := 0 /; -length ≤ x ≤ 0;

g[x_] := 5 /; 0 < x < length;

Plot[g[x], {x, -length, length}, PlotStyle -> :
```

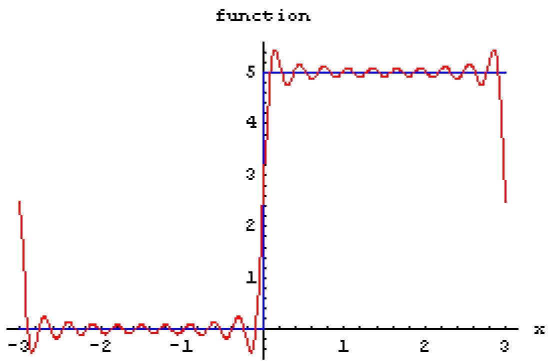


Alter the expressions in red to generate the series you want for this function.
For more efficient computation, we define the series out to twenty terms. If you want it out to more or fewer terms, you can change the term in green.

In[49]:=

```
Clear[a, b, x, length, foursine, fourcos, all]
```

```
length = 3;

b[n_] :=
 b[n] =
  1/length
   (Integrate[Sin[n Pi/length x] * 0, {x, -len
              Integrate[Sin[n Pi/length x]
      {x, 0, length}])

a[n_] :=
 a[n] =
  1/length
   (Integrate[Cos[n Pi/length x] * 0, {x, -len
     Integrate[Cos[n Pi/length x] * 5, {x, 0, l

terms = 20;

foursine[x_, terms] = Sum[b[j] Sin[j Pi/lengtl

fourcos[x_, terms] =
 a[0]/2 + Sum[a[j] Cos[j Pi/length x], {j, 1,

all[x_, terms] = foursine[x, terms] + fourcos[x

Plot[{g[x], Evaluate[all[x, terms]]}, {x, -le
 PlotStyle -> {RGBColor[0, 0, 1], RGBColor[1,
 AxesLabel -> {x, function}];

Print[
 "The function is plotted in blue and its Fo
   series approximation over the interval fr
 length, " to x = ", length, " is plotted in

Print["The Fourier series approximation out
 terms, " is:  ", all[x, terms]]
```



The function is plotted in blue and its Fourier series approximation over the interval from x = -3 to x = 3 is plotted in red.

The Fourier series approximation out to n =20 is:

$$\frac{5}{2}+\frac{10\,\mathrm{Sin}\left[\frac{\pi x}{3}\right]}{\pi}+\frac{10\,\mathrm{Sin}[\pi x]}{3\pi}+\frac{2\,\mathrm{Sin}\left[\frac{5\pi x}{3}\right]}{\pi}+\frac{10\,\mathrm{Sin}\left[\frac{7\pi x}{3}\right]}{7\pi}+\frac{10\,\mathrm{Sin}[3\pi x]}{9\pi}+\frac{10\,\mathrm{Sin}\left[\frac{11\pi x}{3}\right]}{11\pi}+\frac{10\,\mathrm{Sin}\left[\frac{13\pi x}{3}\right]}{13\pi}+\frac{2\,\mathrm{Sin}[5\pi x]}{3\pi}+\frac{10\,\mathrm{Sin}\left[\frac{17\pi x}{3}\right]}{17\pi}+\frac{10\,\mathrm{Sin}\left[\frac{19\pi x}{3}\right]}{19\pi}$$

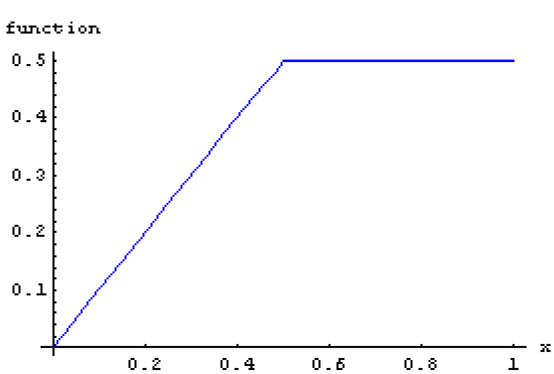
Part III - Define a Function to be Either Even or Odd

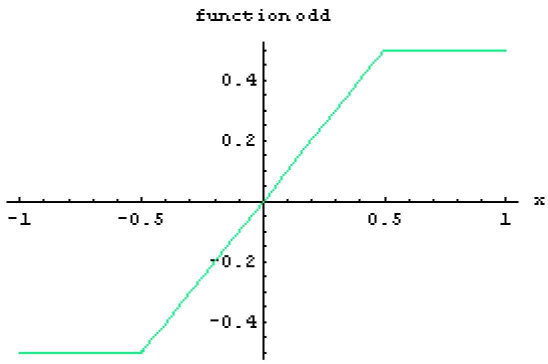
Sometimes, all we want is a Fourier series approximation for a function over an interval and we don't care about how the periodic extension of the function of the function behaves. Consequently, we have the choice of constructing a Fourier series that is either even (cosine terms only) or odd (sine terms only). There may be a reason why you would choose to have only cosine terms or only sine terms to approximate the function. The following is an example of how you can do either. Suppose that the function for which we want the Fourier series takes on the value *x* for ($0 \le x < \frac{1}{2}$) and the value $\frac{1}{2}$ for ($\frac{1}{2} \le x < 1$).

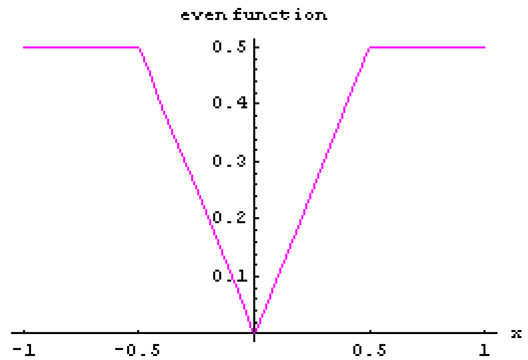
We will begin by defining g[x] over the interval from 0 to 1 and then we will extend the definition to an odd function and to an even function.

In[60]:=

```
Clear[g, x]

g[x_] := x /; 0 ≤ x ≤ .5;

g[x_] := .5 /; .5 < x < 1;

Plot[g[x], {x, 0, 1}, PlotStyle -> RGBColor[0,
 AxesLabel -> {x, function}];

godd[x_] := -.5 /; -1 ≤ x ≤ -.5;

godd[x_] := x /; -.5 ≤ x ≤ 0;

godd[x_] := x /; 0 ≤ x ≤ .5;

godd[x_] := .5 /; .5 ≤ x ≤ 1;

Plot[godd[x], {x, -1, 1}, PlotStyle -> RGBColo
 AxesLabel -> {x, odd function}];

geven[x_] := .5 /; -1 ≤ x ≤ -.5;

geven[x_] := -x /; -.5 ≤ x ≤ 0;

geven[x_] := x /; 0 ≤ x ≤ .5;

geven[x_] := .5 /; .5 ≤ x ≤ 1;

Plot[geven[x], {x, -1, 1}, PlotStyle -> RGBCol
 AxesLabel -> {x, even function}];
```





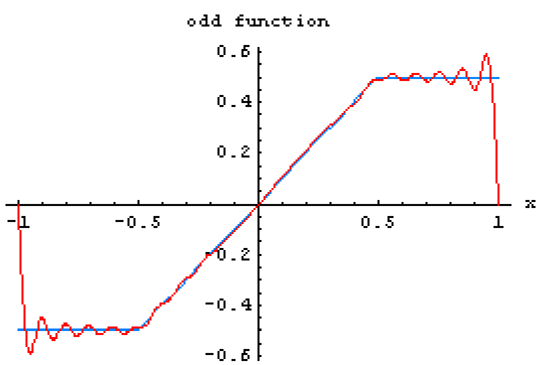


Here we will call upon the simplified formulas used when you are looking only for a sine or only for a cosine series. We double the integrals, but integrate over only half the interval and multiply the integral by 2. We will go to n = 20 in both the sine and cosine series for these evaluations.

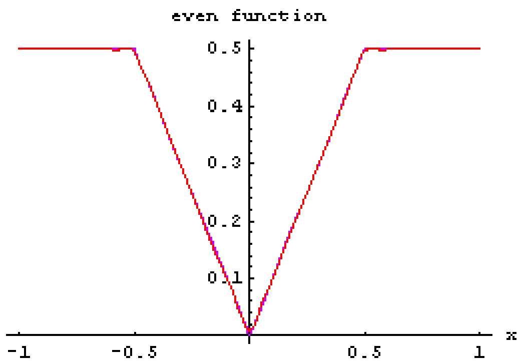
In[74]:=

```
Clear[a, b, x, fousine, fourcos]

length = 1;

b[n_] :=
 b[n] =
  2 / length
   (Integrate[Sin[n π / length x] x, {x, 0, len
     Integrate[Sin[n π / length x] * 0.5,
      {x, length / 2, length}])
```

```
a[n_] :=
 a[n] =
  2 / length
   (Integrate[Cos[n π / length x] x, {x, 0, len
     Integrate[Cos[n π / length x] * 0.5,
      {x, length / 2, length}])

foursine[x_, 20] = Sum[b[j] Sin[j π / length x]

fourcos[x_, 20] =
 a[0] / 2 + Sum[a[j] Cos[j π / length x], {j, 1, 2

Plot[{godd[x], Evaluate[foursine[x, 20]]},
 {x, -length, length},
 PlotStyle -> {RGBColor[0, .5, 1], RGBColor[1,
 AxesLabel -> {x, "odd function"}];

Print["The Fourier sine series is ", foursin

Plot[{geven[x], Evaluate[fourcos[x, 20]]},
 {x, -length, length},
 PlotStyle -> {RGBColor[.8, 0, 1], RGBColor[1,
 AxesLabel -> {x, "even function"}, PlotRange

Print["The Fourier cosine series is ", fourc
```



The Fourier sine series is foursine[x, 30.]



The Fourier cosine series is
0.375 - 0.202642 Cos[3.14159 x] -
0.101321 Cos[6.28319 x] -
0.0225158 Cos[9.42478 x] -
0.00810569 Cos[15.708 x] -
0.0112579 Cos[18.8496 x] -
0.00413556 Cos[21.9911 x] -
0.00250176 Cos[28.2743 x] -
0.00405285 Cos[31.4159 x] -
0.00167473 Cos[34.5575 x] -
0.00119907 Cos[40.8407 x] -
0.00206778 Cos[43.9823 x] -
0.000900633 Cos[47.1239 x] -
0.000701185 Cos[53.4071 x] -
0.00125088 Cos[56.5487 x] -
0.000561336 Cos[59.6903 x]

Note that both the sine and the cosine expansions fit the function over the interval 0 to 1, but, over a broader range, one is even and one is odd. The cosine series seems to be a better fit.

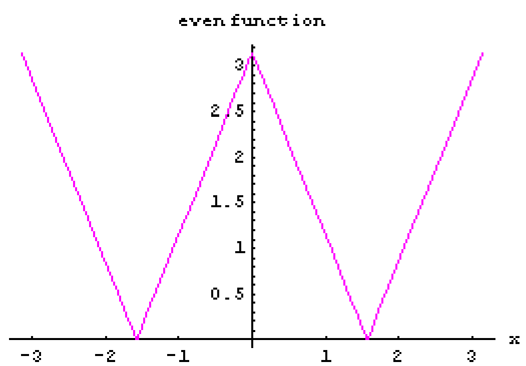
You Try It - Part III

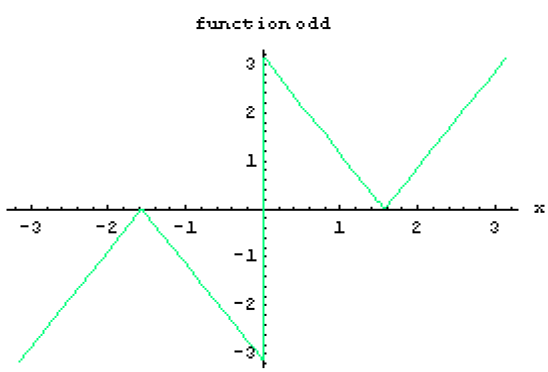
Find both a Fourier sine series and cosine series expansion for the function $|2x - \pi|$ over the interval from 0 to π. We could simply define the function as given, but, for clarity in plotting, we will define both its even counterpart and its odd counterpart over the interval from $-\pi$ to π. You should note that this can be done by shifting the function to the left a distance of π units.

In[84]:=

```
Clear[geven, godd]

geven[x_] := Abs[2 (x + π) - π] /; -π ≤ x < 0;
```

```
geven[x_] := Abs[2 x - π] /; 0 ≤ x < π;

Plot[geven[x], {x, -π, π}, PlotStyle -> RGBCol
 AxesLabel -> {x, even function}];

godd[x_] := -Abs[2 (x + π) - π] /; -π ≤ x < 0;

godd[x_] := Abs[2 x - π] /; 0 ≤ x < π;

Plot[godd[x], {x, -π, π}, PlotStyle -> RGBCol
 AxesLabel -> {x, odd function}];
```

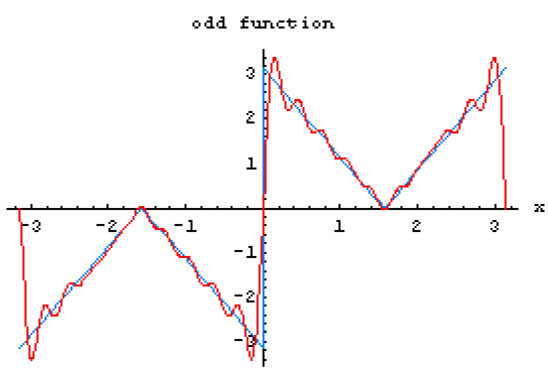




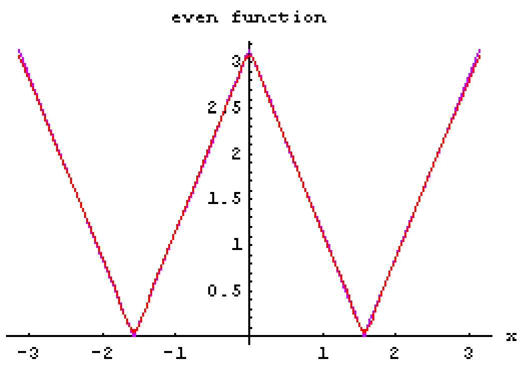
Note that this function is $-(2x - \pi)$ between 0 and $\pi/2$, but $+(2x - \pi)$ from $\pi/2$ to π. Put in the appropriate expressions for the terms in red.

In[91]:=

```
Clear[a, b, x, n, foursine, fourcos]
length = π;
b[n_] :=
  b[n] =
   2 / length
    (Integrate[Sin[n π / length x] (-(2 x - π)),
       {x, 0, length / 2}]     +
      Integrate[Sin[n π / length x] (2 x - π),
       {x, length / 2, length}])
a[n_] :=
  a[n] =
   2 / length
    (Integrate[Cos[n π / length x] (-(2 x - π)),
       {x, 0, length / 2}] +
      Integrate[Cos[n π / length x] (2 x - π),
       {x, length / 2, length}])
foursine[x_, 20] = N[Sum[b[j] Sin[j π / length
Plot[{godd[x], Evaluate[foursine[x, 20]]},
  {x, -length, length},
  PlotStyle -> {RGBColor[0, .5, 1], RGBColor[1
  AxesLabel -> {x, "odd function"}, PlotRange
Print["The Fourier sine series is: ", foursi
fourcos[x_, 20] =
  N[a[0] / 2 + Sum[a[j] Cos[j π / length x], {j,
Plot[{geven[x], Evaluate[fourcos[x, 20]]},
  {x, -length, length},
  PlotStyle -> {RGBColor[.8, 0, 1], RGBColor[1
  AxesLabel -> {x, "even function"}, PlotRange
Print["The Fourier cosine series is: ", four
```



```
The Fourier sine series is:
 1.45352 Sin[x] + 1.61628 Sin[3. x] +
  0.698141 Sin[5. x] + 0.623398 Sin[7. x] +
  0.413006 Sin[9. x] + 0.384682 Sin[11. x] +
  0.292624 Sin[13. x] + 0.277984 Sin[15. x] +
  0.226483 Sin[17. x] + 0.21758 Sin[19. x]
```




```
The Fourier cosine series is:
 1.5708 + 1.27324 Cos[2. x] +
  0.141471 Cos[6. x] + 0.0509296 Cos[10. x] +
  0.0259845 Cos[14. x] + 0.015719 Cos[18. x]
```

Out[91]=

Null^{10}

Is the cosine function a better fit again? Try this out with other functions.

Part IV - Analyze Musical Tones of a Clarinet

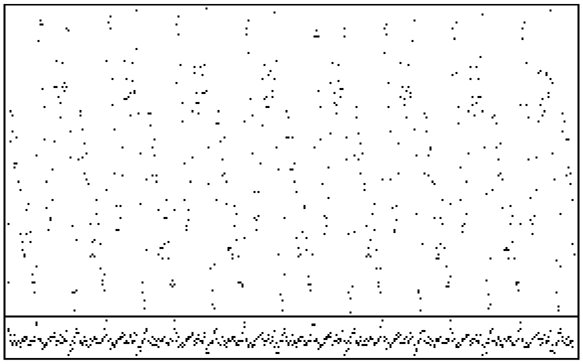
The harmonies in mathematics are heard in musical instruments such as the piano and clarinet and are seen in the Fourier Series of the form: a[0]+a[1] Cos[$\frac{2\pi 1 t}{\lambda}$]+b[1] Sin[$\frac{2\pi 1 t}{\lambda}$]+a[2] Cos[$\frac{2\pi 2 t}{\lambda}$]+b[2] Sin[$\frac{2\pi 2 t}{\lambda}$]+...+a[n] Cos[$\frac{2\pi n t}{\lambda}$]+b[n] Sin[$\frac{2\pi n t}{\lambda}$]
A group of students from Carroll College in Montana recorded musical tones from a clarinet and from a piano using a sound sensor that interfaces with a graphing calculator. The recorded signals represent a measure of the loudness of the tones as a function of time. After sampling the tones and graphically observing the periodic pattern, they selected what appeared to be approximately one period of the signal and then used Riemann sums to approximate the integrals for the Fourier coefficients. The following data sets are those that were collected by the students. Lambda (λ) represents the length of the interval selected for what appears to be

one period in the recorded signal. Increasing the number of Fourier terms gives better approximations of the signal.

First we look at all the sample points collected when the note A is played on a clarinet. You can hear the recording of the sound itself with the *Mathematica* **ListPlay** command, provided your speakers are turned on. It is a very short tone.

In[92]:=

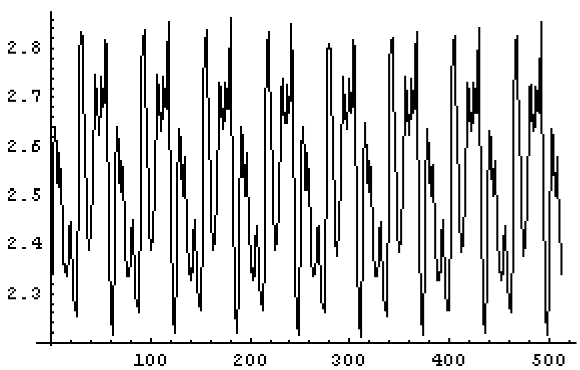
```
clarinet = {2.27817, 2.40131, 2.57483, 2.642, 2.6308
  2.53005, 2.51886, 2.58603, 2.53005, 2.49087, 2.4740
  2.34534, 2.36213, 2.33974, 2.33414, 2.37892, 2.4293
  2.38452, 2.36213, 2.29496, 2.27817, 2.27817, 2.2557
  2.54684, 2.78753, 2.82112, 2.83231, 2.82112, 2.6867
  2.58043, 2.49087, 2.43489, 2.39012, 2.40691, 2.4125
  2.51326, 2.57483, 2.69238, 2.74835, 2.69238, 2.6699
  2.62521, 2.69238, 2.74276, 2.69798, 2.68118, 2.6923
  2.80992, 2.59722, 2.54125, 2.39571, 2.25578, 2.2557
  2.26137, 2.36773, 2.53565, 2.642, 2.63081, 2.60282,
  2.50766, 2.56923, 2.55244, 2.49647, 2.48527, 2.4013
  2.36213, 2.33974, 2.33414, 2.36773, 2.4237, 2.45169
  2.35653, 2.30615, 2.27257, 2.28376, 2.26137, 2.2949
  2.74835, 2.82672, 2.81552, 2.83791, 2.70917, 2.6532
  2.50766, 2.44049, 2.39012, 2.39012, 2.4125, 2.44049
  2.56364, 2.66439, 2.74835, 2.70917, 2.66439, 2.6699
  2.67559, 2.74276, 2.70357, 2.68118, 2.67559, 2.7707
  2.6364, 2.55244, 2.44609, 2.27817, 2.25578, 2.22219
  2.34534, 2.49647, 2.6364, 2.6364, 2.60842, 2.56364,
  2.54684, 2.58043, 2.50206, 2.49087, 2.4293, 2.33974
  2.35093, 2.32854, 2.35653, 2.4125, 2.45169, 2.4125,
  2.32854, 2.27257, 2.28376, 2.26697, 2.27257, 2.4293
  2.82112, 2.80433, 2.83791, 2.73716, 2.65879, 2.6196
  2.45169, 2.39571, 2.39012, 2.40691, 2.4293, 2.49087
  2.642, 2.73156, 2.72037, 2.65879, 2.67559, 2.6364,
  2.73156, 2.71477, 2.67559, 2.67559, 2.73716, 2.8603
  2.56364, 2.48527, 2.30615, 2.26137, 2.23339, 2.2221
  2.45169, 2.61401, 2.642, 2.61401, 2.57483, 2.50206,
  2.58603, 2.50766, 2.49087, 2.44609, 2.35093, 2.3453
  2.32854, 2.33974, 2.39571, 2.44049, 2.4237, 2.36773
  2.27817, 2.27817, 2.27257, 2.26697, 2.37332, 2.6196
  2.80433, 2.83231, 2.76514, 2.66439, 2.63081, 2.5524
  2.4125, 2.39012, 2.40691, 2.4181, 2.46848, 2.53005,
  2.70917, 2.73716, 2.66439, 2.66999, 2.6476, 2.6476,
  2.72596, 2.68118, 2.66999, 2.70917, 2.84911, 2.7483
  2.51326, 2.33974, 2.25018, 2.23898, 2.21659, 2.2949
  2.58603, 2.642, 2.62521, 2.58603, 2.51326, 2.51326,
  2.53005, 2.49087, 2.46848, 2.37332, 2.33414, 2.3509
  2.33974, 2.38452, 2.43489, 2.44049, 2.37332, 2.3453
  2.27257, 2.27257, 2.26137, 2.33974, 2.56364, 2.7987
  2.80992, 2.79873, 2.68118, 2.6476, 2.58043, 2.48527
  2.37892, 2.39571, 2.4125, 2.45728, 2.51886, 2.59162
  2.74276, 2.67559, 2.65879, 2.6532, 2.6364, 2.70357,
  2.68678, 2.66999, 2.68678, 2.81552, 2.79873, 2.5860
  2.37892, 2.25578, 2.24458, 2.211, 2.26697, 2.38452,
```



To get a clearer understanding of the signal you are hearing, let's look at a graph of the signal with the points connected. Note the periodic behavior of the musical tone. The units on the vertical axis represent the sound levels in the units recorded by the particular sound sensor.

In[93]:=

```
ListPlot[clarinet, PlotJoined -> True];
```

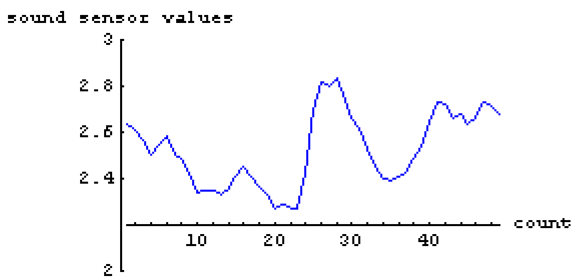


The students chose the points from 129 through 177 to represent one period. Does that look about right? That means that the interval will be of length $\lambda = 177 - 129$.

In[94]:=

```
Clear[period]

start = 129;

stop = 177;

λ = stop - start;

period = Table[clarinet[[c]], {c, start, stop}];
```

```
clarinetplot = ListPlot[period, PlotRange → {(
  PlotJoined → True,
  AxesLabel -> {count, "sound sensor values"}
  AxesOrigin -> {0, 2.2}, PlotStyle -> RGBColo
```



We want to get a Fourier series approximation for the set of points you see in the graph. In what is to follow, note the use of Riemann sums rather than integrals to approximate the Fourier coefficients. Because you have a set of discrete values (assigned to the symbol period) instead of a mathematical function to approximate, you can estimate the areas of each small rectangle to be the width of the rectangle (1/λ) times the appropriate height (sound sensor values as in the above graph). Adding up these areas gives approximations to the integrals over the interval of length λ.

In[100]:=

```
Clear[a, b, n, t, terms, clarinetfour]

terms = 20;
```

$$a[0] = \frac{\sum_{t=1}^{\lambda} \text{period}[[t]]}{\lambda};$$

$$a[n\_] := a[n] = \frac{2 \sum_{t=1}^{\lambda} \text{period}[[t]]\, N[\text{Cos}[\frac{2 \pi n t}{\lambda}]]}{\lambda}$$

$$b[n\_] := b[n] = \frac{2 \sum_{t=1}^{\lambda} \text{period}[[t]]\, N[\text{Sin}[\frac{2 \pi n t}{\lambda}]]}{\lambda}$$

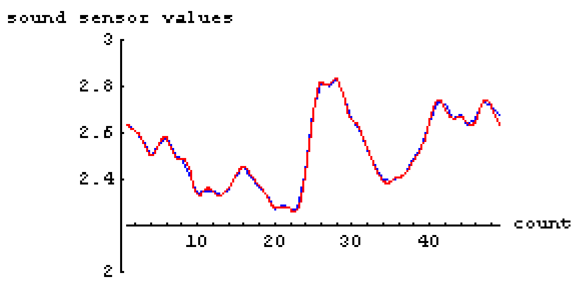
clarinetfour[t_] =

$$N[a[0] + \sum_{n=1}^{\text{terms}} \left(a[n]\, \text{Cos}[\frac{2 \pi n t}{\lambda}] + b[n]\, \text{Sin}[\frac{2 \pi}{\lambda}\right.$$

Now we can see what our simulated clarinet plot looks like.

In[105]:=

```
clarinetfourplot = Plot[Evaluate[clarinetfou
  {t, 1, λ + 1}, PlotStyle → {RGBColor[1, 0, 0]
  AxesLabel -> {count, "sound sensor values"}
  AxesOrigin -> {0, 2.2}, DisplayFunction -> I

Show[clarinetplot, clarinetfourplot,
 DisplayFunction -> $DisplayFunction];

Print[
 "The original plot is in blue and the Four
   plot is in red."]
```




```
The original plot is in blue and
  the Fourier series plot is in red.
```

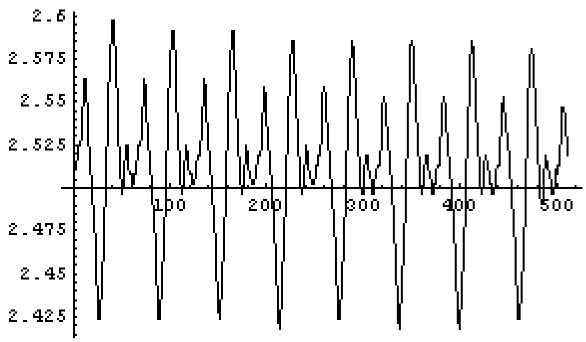
Based upon what we have done, do you think that the Fourier series we found can be used to reproduce the sound of a note? To learn more about this, you might check into the way in which music synthesizers work and the technique used to replicate a phone number dialed on a touch-tone phone. The important point that you should gather from all this is that periodic phenomena, discrete or continuous, can be approximated by Fourier series representation.

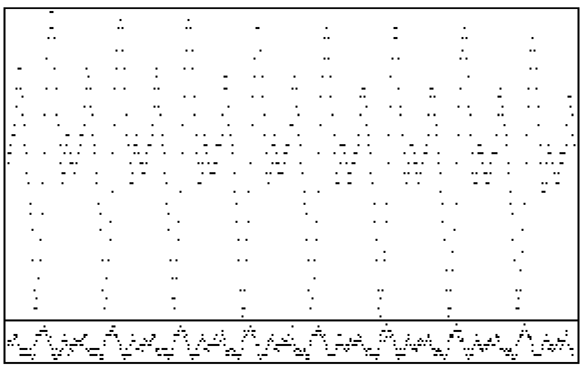
You Try It - Part IV - The Piano

As with the clarinet, we first we look at and listen to all the sample points collected when the note middle-C is played on a piano.

In[108]:=

piano = {2.50766, 2.51326, 2.51326, 2.51886, 2.52445
2.52445, 2.53005, 2.54125, 2.55244, 2.56364, 2.5636
2.54684, 2.53565, 2.53005, 2.51886, 2.51326, 2.5020
2.48527, 2.47967, 2.46848, 2.45169, 2.43489, 2.4237
2.4293, 2.44049, 2.45169, 2.46288, 2.47967, 2.49647
2.53565, 2.55244, 2.56923, 2.58043, 2.58603, 2.5972
2.58603, 2.58043, 2.56364, 2.55244, 2.53565, 2.5300
2.50766, 2.50206, 2.49647, 2.50206, 2.51326, 2.5188
2.52445, 2.51326, 2.50766, 2.50766, 2.50766, 2.5020
2.50766, 2.50766, 2.51326, 2.51886, 2.52445, 2.5244
2.53005, 2.54125, 2.55244, 2.56364, 2.55804, 2.5524
2.53565, 2.52445, 2.51886, 2.51326, 2.50206, 2.4964
2.47967, 2.46848, 2.45169, 2.43489, 2.4237, 2.4237,
2.44049, 2.45169, 2.46288, 2.47408, 2.49087, 2.5132
2.55244, 2.56364, 2.57483, 2.58603, 2.59162, 2.5916
2.57483, 2.56364, 2.55244, 2.53565, 2.52445, 2.5188
2.49647, 2.49647, 2.50206, 2.51326, 2.51886, 2.5244
2.51326, 2.50766, 2.50766, 2.50766, 2.50206, 2.5020
2.50766, 2.51326, 2.51886, 2.51886, 2.51886, 2.5244
2.54125, 2.55244, 2.56364, 2.55804, 2.55244, 2.5412
2.52445, 2.51886, 2.51326, 2.50206, 2.49647, 2.4852
2.46848, 2.45169, 2.44049, 2.4293, 2.4237, 2.4293,
2.45169, 2.46288, 2.47408, 2.49087, 2.51326, 2.5300
2.56364, 2.57483, 2.58603, 2.59162, 2.59162, 2.5860
2.56364, 2.54684, 2.53565, 2.52445, 2.51326, 2.5076
2.49647, 2.50206, 2.50766, 2.51886, 2.52445, 2.5188
2.50766, 2.50766, 2.50206, 2.50206, 2.50206, 2.5020
2.50766, 2.51886, 2.51886, 2.51886, 2.51886, 2.5244
2.55244, 2.55804, 2.55804, 2.55244, 2.54125, 2.5300
2.51886, 2.51326, 2.50206, 2.49087, 2.48527, 2.4740
2.45169, 2.43489, 2.4237, 2.4181, 2.4237, 2.43489,
2.46288, 2.47408, 2.49087, 2.50766, 2.53005, 2.5468
2.56923, 2.58603, 2.58603, 2.58603, 2.58603, 2.5748
2.54684, 2.53565, 2.52445, 2.51326, 2.50206, 2.4964
2.50206, 2.50766, 2.51886, 2.52445, 2.51886, 2.5132
2.50766, 2.50206, 2.50206, 2.50206, 2.50206, 2.5076
2.51326, 2.51886, 2.51886, 2.51886, 2.53005, 2.5300
2.55804, 2.55804, 2.55244, 2.54125, 2.53565, 2.5244
2.50766, 2.50206, 2.49087, 2.48527, 2.47967, 2.4628
2.43489, 2.4237, 2.4237, 2.4293, 2.44049, 2.45169,
2.47408, 2.49087, 2.51326, 2.53005, 2.54684, 2.5580
2.58043, 2.58603, 2.58603, 2.58043, 2.56923, 2.5580
2.53565, 2.52445, 2.51326, 2.50206, 2.49647, 2.4964
2.50766, 2.51886, 2.51886, 2.51886, 2.51326, 2.5076
2.50206, 2.49647, 2.49647, 2.50206, 2.50766, 2.5076



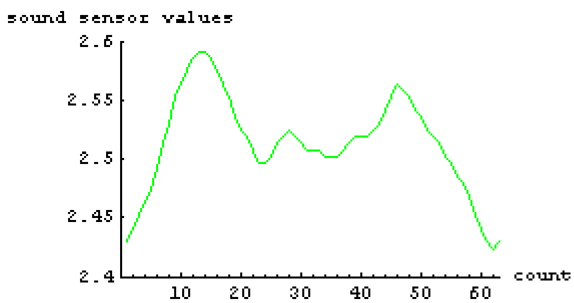


The students chose the points from 90 through 152 to represent one period. Does this look about right?

You will note that this makes the period slightly longer for the note "C" than for the note "A". Might this have anything to do with the notes themselves?

In[109]:=

```
start = 90;
stop = 152;
λ = stop - start;
period = Table[piano[[c]], {c, start, stop}];
pianoplot =
 ListPlot[period, PlotRange → {{0, λ + 1}, {2.·
  AxesLabel -> {count, "sound sensor values"}
  AxesOrigin -> {0, 2.4}, PlotJoined → True,
  PlotStyle -> RGBColor[0, 1, 0]];
```



In[110]:=

```
Clear[a, b, n, t, terms, pianofour]
```

```
terms = 30;
```

$$a[0] = \frac{\sum_{t=1}^{\lambda} \text{period}[[t]]}{\lambda};$$

$$a[n\_] := a[n] = \frac{2 \sum_{t=1}^{\lambda} \text{period}[[t]]\, N\left[\text{Cos}\left[\frac{2\pi n t}{\lambda}\right]\right]}{\lambda}$$

$$b[n\_] := b[n] = \frac{2 \sum_{t=1}^{\lambda} \text{period}[[t]]\, N\left[\text{Sin}\left[\frac{2\pi n t}{\lambda}\right]\right]}{\lambda}$$

pianofour[t_] =

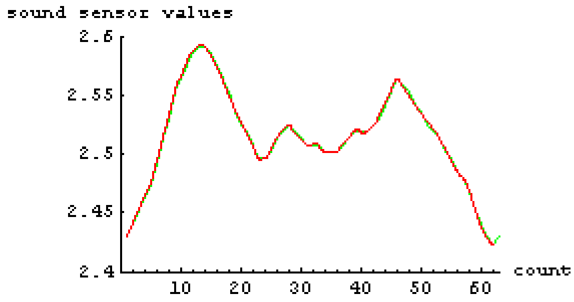
$$N\left[a[0] + \sum_{n=1}^{\text{terms}} \left(a[n]\, \text{Cos}\left[\frac{2\pi n t}{\lambda}\right] + b[n]\, \text{Sin}\left[\frac{2\pi}{\lambda}\right.\right.\right.$$

In[116]:=

```
pianofourplot = Plot[Evaluate[pianofour[t]],
  PlotStyle → {RGBColor[1, 0, 0]},
  AxesLabel -> {count, "sound sensor values"}
  AxesOrigin -> {0, 2.4}, DisplayFunction -> I
```

```
Show[pianoplot, pianofourplot,
 DisplayFunction -> $DisplayFunction];
```

```
Print[
 "The original plot is in green and the Fou
   plot is in red."]
```




```
The original plot is in green and
  the Fourier series plot is in red.
```

Compare the clarinet and the piano. The clarinet is in blue and the piano is in green in the following graph.

In[119]:=

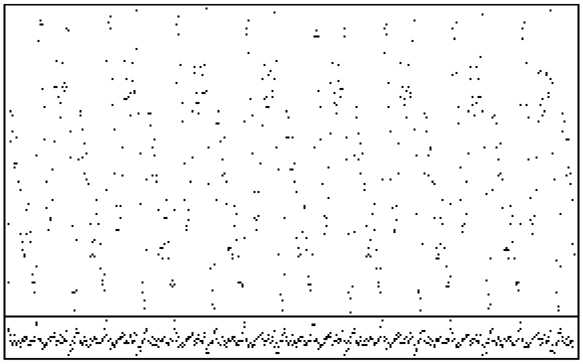
```
Show[clarinetplot, pianoplot, PlotRange -> Al
```

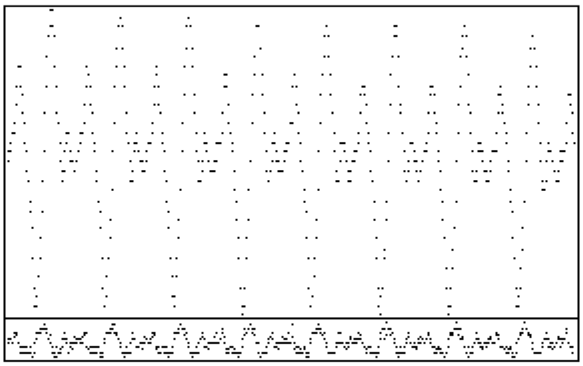
Why might the piano tone be smoother than the clarinet tone? Listen to their sounds in succession

In[120]:=

```
ListPlay[clarinet];
```

```
ListPlay[piano];
```





About *Mathematica*

NOTE: The following double assignment terminology used, ***b[n_]:= b[n]* = ...** , forces *Mathematica* to "remember" the value of b[n] for each value of n, making subsequent computations which repeatedly use b[n] much more efficiently. Because of the way it is stored, if you decide to enter a new function, **be sure that you Clear a and b first**.
Go back to where you were in the notebook.

There are many ways to define piecewise continuous functions in Mathematica. This is one of them. The **/;** is used to specify the values of x over which f[x] is to take on the specified value. Other ways to define piecewise continuous functions include using the **UnitStep** function or the **Which** function. You can look these up in the *Help* menu to get more information on their use. You will need to use the UnitStep function if you want to call upon the *Mathematica* **FourierTrigSeries**.
Go back to where you were in the notebook.

Created by *Mathematica* (April 2, 2005)