WaveShaping Synthesis on the NeXT

DSP56001 Unit Generator and MusicKit interface by
Eric J. Graves (ejgraves@phoenix.princeton.edu)

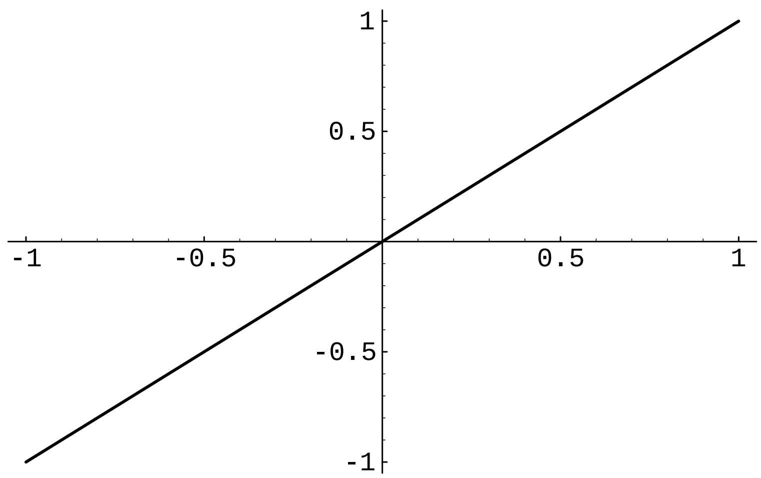
First, we should examine the theory of Waveshaping synthesis (as described by Arfib and LeBrun). First, let us examine the output from a 1st order Chebychev polynomial--that is, the graph that it will produce:

(Ernie.Princeton.edu) In[1]:=

```
cheby1[x_] := x
```

(Ernie.Princeton.edu) In[2]:=

```
Plot[cheby1[x],{x,-1,1}]
```



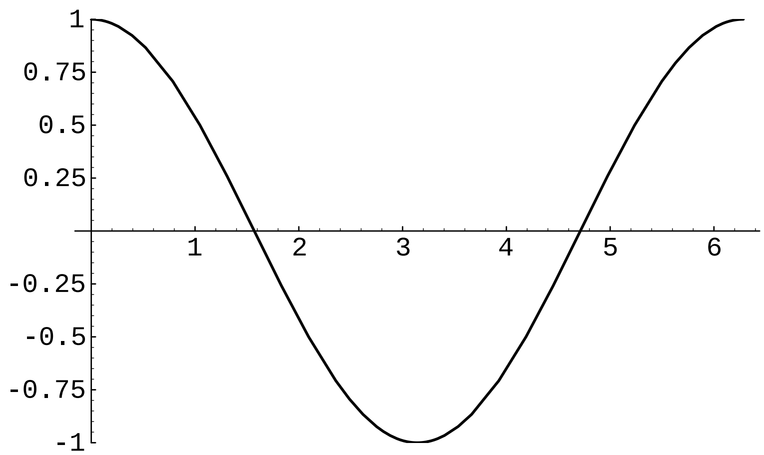
(Ernie.Princeton.edu) Out[2]=

```
-Graphics-
```

Note that this will produce exactly the same value at the output as is input to it; this is the default wavetable that the Waveshaping SynthPatch uses.

(Ernie.Princeton.edu) In[3]:=

```
Plot[Cos[time],{time,0,2*Pi},PlotRange->{-1,1}]
```

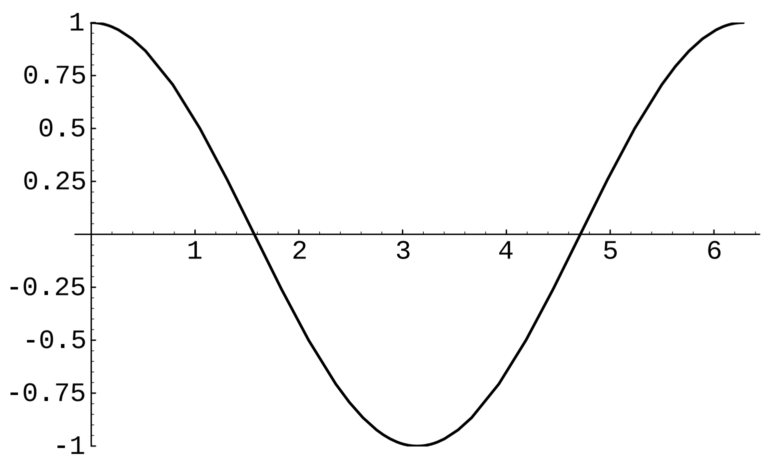


(Ernie.Princeton.edu) Out[3]=

```
-Graphics-
```

(Ernie.Princeton.edu) In[4]:=

```
Plot[cheby1[Cos[time]],{time,0,2*Pi},PlotRange->{-1,1}]
```



(Ernie.Princeton.edu) Out[4]=

```
-Graphics-
```

As you can see, the first chebychev polynomail simply takes an input and performs a linear mapping of it into the output and therefore is the First Harmonic. This in itself is actually entirely useless--why insert the overhead of a wavetable lookup when you just want a sine wave output? There is no reason, in fact. However, WaveShaping becomes an extremely useful and efficient method of synthesizing complex sounds, as you will see below:

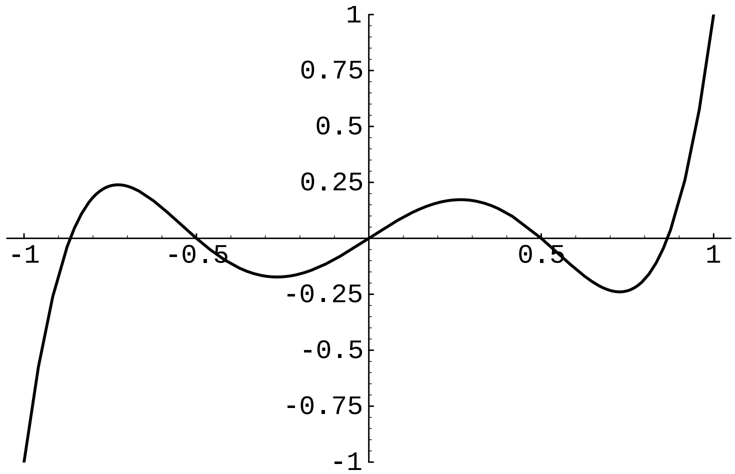
First, let us consider a plot of the first 3 odd Chebychev polynomials.

(Ernie.Princeton.edu) In[5]:=

```
cheby2[x_] := (x/3 + (4*x^3 -3*x)/3 +(16*x^5 - 20*x^3 +5*x)
```

(Ernie.Princeton.edu) In[6]:=

```
Plot[cheby2[x],{x,-1,1},PlotRange->{-1,1}]
```



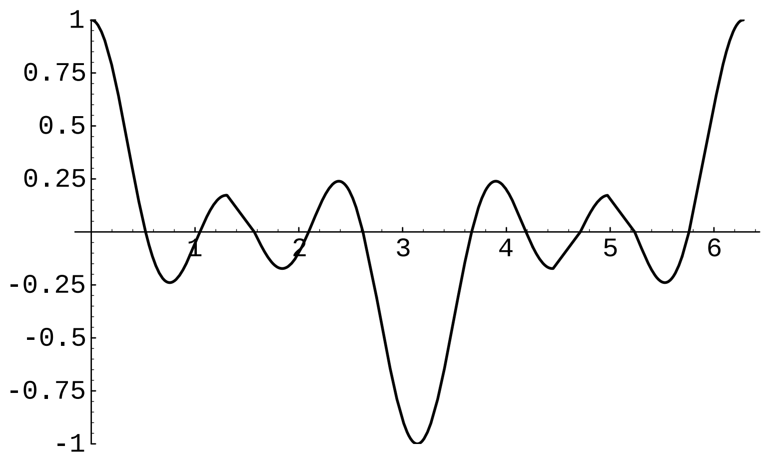
(Ernie.Princeton.edu) Out[6]=

```
-Graphics-
```

The above table is however, somewhat unclear as to what impact it will have on the input values, so let us examine the output when we feed a sine wave into it:

(Ernie.Princeton.edu) In[7]:=

```
Plot[cheby2[Cos[time]],{time,0,2*Pi},PlotRange->{-1,1}]
```



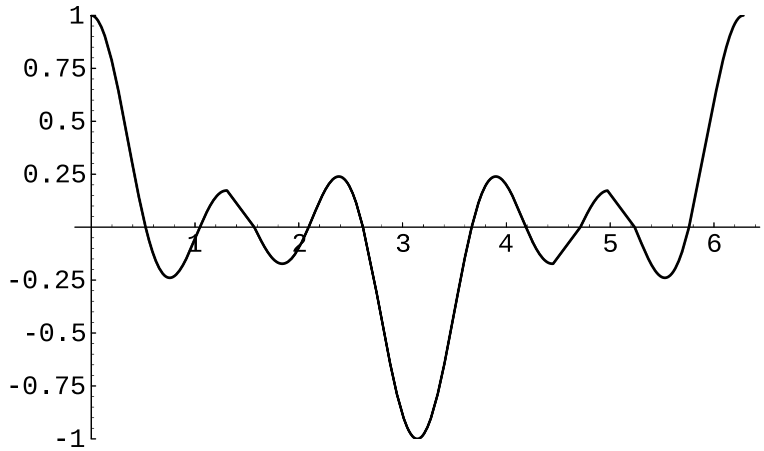
(Ernie.Princeton.edu) Out[7]=

```
-Graphics-
```

As we can see, either visually or through a mathematical proof, this is in fact the same as doing additive synthesis of the first three harmonics:

(Ernie.Princeton.edu) In[8]:=

```
Plot[(Cos[time]/3 + Cos[3*(time)]/3 + Cos[5*(time)]/3),
    {time,0,2*Pi},PlotRange->{-1,1}]
```



(Ernie.Princeton.edu) Out[8]=

```
-Graphics-
```

Next, we can examine adding harmonics 1, 2, 4, 6, and 8. (Through wavetable synthesis using the appropriate chebychev polynomails (1, 2, 4, 6, 8).

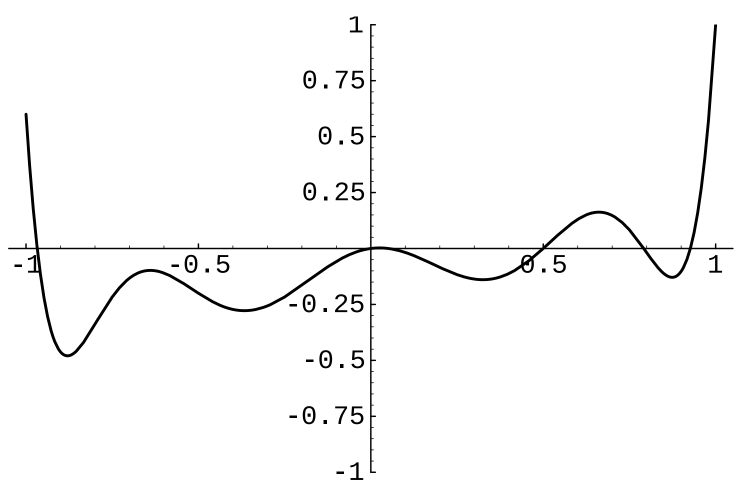
(Ernie.Princeton.edu) In[9]:=

```
cheby3[x_] := .2*x + .2*(2*x^2 -1) + .2*(8*x^4 - 8*x^2 +1)
```

First, let's look at the table:

(Ernie.Princeton.edu) In[10]:=

```
Plot[cheby3[x],{x,-1,1},PlotRange->{-1,1}]
```



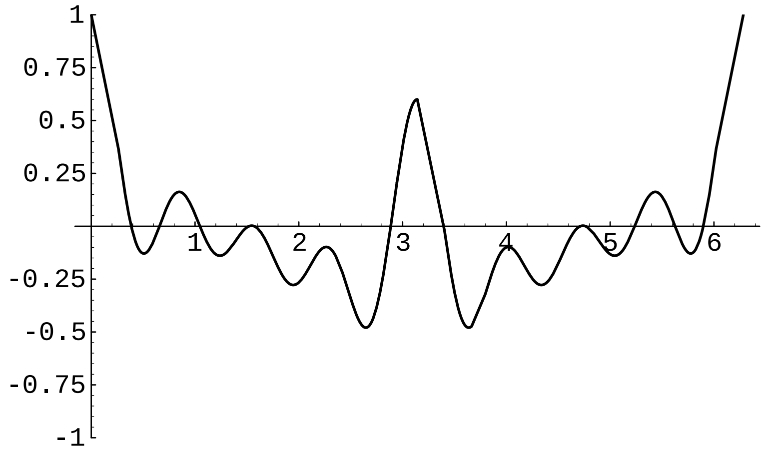
(Ernie.Princeton.edu) Out[10]=

```
-Graphics-
```

And now, one cycle of a wave created by it:

(Ernie.Princeton.edu) In[11]:=

```
Plot[cheby3[Cos[time]],{time,0,2*Pi},PlotRange->{-1,1}]
```



(Ernie.Princeton.edu) Out[11]=

```
-Graphics-
```

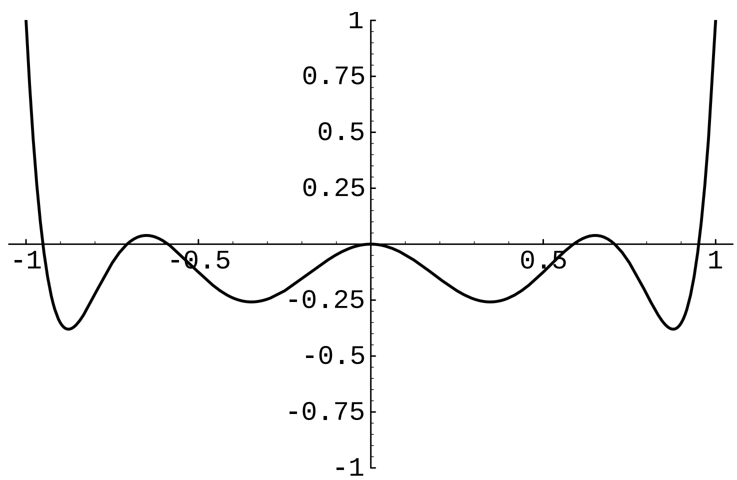
As another option, let us examine adding the 2nd, 4th, 6th, and 8th Chebychev polynomials. First, the wavetable:

(Ernie.Princeton.edu) In[12]:=

```
cheby4[x_] := .25*(2*x^2 -1) + .25*(8*x^4 - 8*x^2 +1) + .25
```

(Ernie.Princeton.edu) In[13]:=

```
Plot[cheby4[x],{x,-1,1},PlotRange->{-1,1}]
```



(Ernie.Princeton.edu) Out[13]=

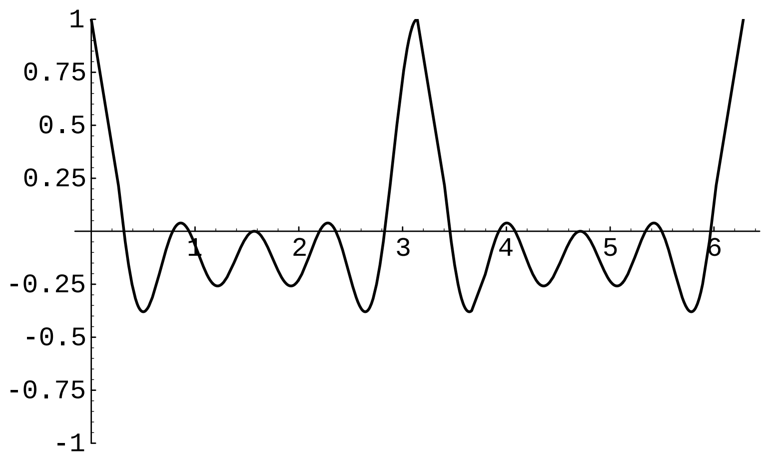
```
-Graphics-
```

As can bee seen, and/or proven mathematically, wavetables consisting of purely even harmonics (polynomials) will be symetrical about the y axis.

And, the waveform that feeding a sine wave into it results in:

(Ernie.Princeton.edu) In[14]:=

```
Plot[cheby4[Cos[time]],{time,0,2*Pi},PlotRange->{-1,1}]
```



(Ernie.Princeton.edu) Out[14]=

```
-Graphics-
```

However, we still have a problem with ChebyChev synthesis... Namely, we need a way of figuring out the kth ChebyChev polynomial; it is too inflexible to simply keep a table of them, such as the following:

(Ernie.Princeton.edu) In[23]:=

```
Ch0[x_] := 1;
Ch1[x_] := x;
Ch2[x_] := 2  x^2 - 1;
Ch3[x_] := 4  x^3 - 3 x;
Ch4[x_] := 8  x^4 - 8 x^2  + 1;
Ch5[x_] := 16 x^5 - 20x^3  + 5 x;
Ch6[x_] := 32 x^6 - 48x^4  + 18 x^2 -1;
Ch7[x_] := 64 x^7 - 112x^5 + 56 x^3 -7 x;
Ch8[x_] := 128x^8 - 256x^6 + 160x^4 -32x^2 +1;
```

While the table would work, it would not be flexible enough to be able to implemenet any additive synthesis partials we can come up with. However, math again comes to the rescue. Most higher math books give a few different methods of calculating the kth ChebyChev polynomial.

The first of those methods is the Recursive method--that is,

(Ernie.Princeton.edu) In[80]:=

```
Tch[0,x_] := 1;
Tch[1,x_] := x;
Tch[n_,x_] := 2 x Tch[n-1,x] - Tch[n-2,x]
```

While this is an interesting property of ChebyChev Polynomials, it is, unfortunately not the best way to calculate a table of the values.

For this task, we turn instead to an alternative representation (involving the "Choose" function--also known as the binomial coeffecient).

```
T[n_,x_] := x^n
        - Binomial[n,2] x^(n-2) (1-x^2)
        + Binomial[n,4] x^(n-4) (1-x^2)^2
        - ...
```

This, of course, can be written as the sum:

(Ernie.Princeton.edu) In[111]:=

```
Sum[(-1)^(k/2) Binomial[n,k] x^(n-2) (1-x^2)(k/2),{k,0,n}]
```