

To evaluate the function over a different domain, you need to reexecute the SetScale command with different parameters. This redefines “x” for the wave. Then you need to reexecute the waveform assignment statement. For example,

```
SetScale/I x, 0, 2*PI, wave0      // plot function from x=0 to x=2π
wave0 = 3*sin(x) + 1.5*sin(2*x + PI/6)
```

Reexecuting commands is easy, using the shortcuts shown in **History Area** on page II-9.

### Function Plotting Using Dependencies

If you get tired of reexecuting the waveform assignment statement each time you change the domain, you can use a dependency to cause Igor to automatically reexecute it. To do this, use `:=` instead of `=`.

```
wave0 := 3*sin(x) + 1.5*sin(2*x + PI/6)
```

See Chapter IV-9, **Dependencies**, for details.

You have made `wave0` depend on “X”. The SetScale operation changes the meaning of “X” for the wave. Now when you do a SetScale on `wave0`, Igor will automatically reexecute the assignment.

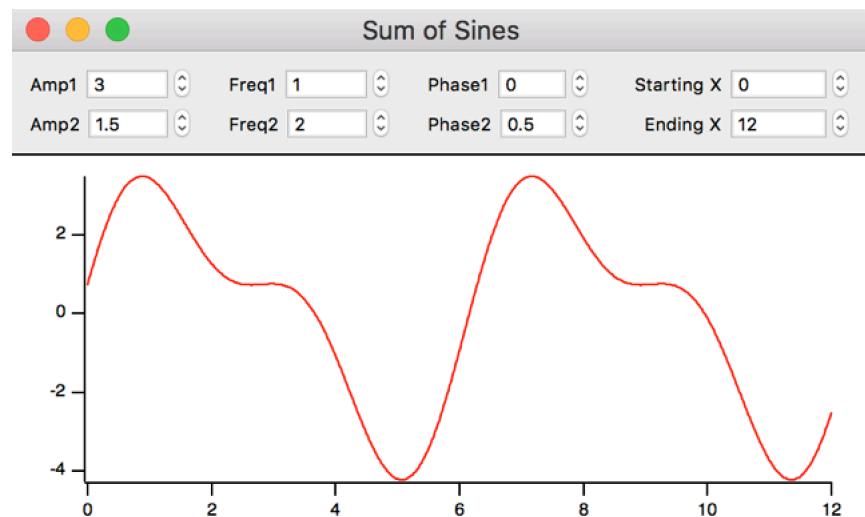
You can take this further by using global variables instead of literal numbers in the right-hand expression. For example:

```
Variable/G amp1=3, amp2=1.5, freq1=1, freq2=2, phase1=0, phase2=PI/6
wave0 := amp1*sin(freq1*x + phase1) + amp2*sin(freq2*x + phase2)
```

Now, `wave0` depends on these global variables. If you change them, Igor will automatically reexecute the assignment.

### Function Plotting Using Controls

For a slick presentation of function plotting, you can put controls in the graph to set the values of the global variables. When you change the value in the control, the global variable changes, which reexecutes the assignment. This changes the wave, which updates the graph. Here is what the graph would look like.



We've added two additional global variables and connected them to the Starting X and Ending X controls. This allows us to set the domain. These controls are both linked to an action procedure that does a SetScale on the wave.

Controls are explained in detail in Chapter III-14, **Controls and Control Panels**.