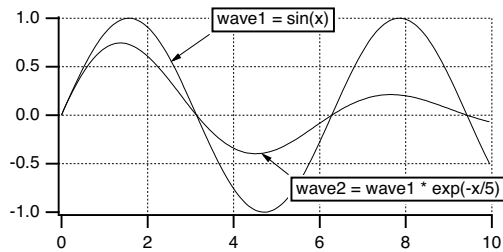


Wave assignment is an important Igor feature and it will be worth your time to understand it. You may need to reread this material and do some experimentation on the command line. If you are new to Igor, you may prefer to return to this topic after you have gained experience.

## Wave Assignment Example

This command sequence illustrates some of the ideas explained above.

```
Make/N=200 wave1, wave2           // Two waves, 200 points each
SetScale/P x, 0, .05, wave1, wave2 // Set X values from 0 to 10
Display wave1, wave2              // Create a graph of waves
wave1 = sin(x)                    // Assign values to wave1
wave2 = wave1 * exp(-x/5)          // Assign values to wave2
```



Since wave1 has 200 points, the wave assignment statement `wave1=sin(x)` evaluates `sin(x)` 200 times, once for each point in wave1. The first point of wave1 is point number 0 and the last point of wave1 is point number 199. The symbol `p`, not used in this example, goes from 0 to 199. The symbol `x` steps through the 200 X values for wave1 which start from 0 and step by .05, as specified by the `SetScale` command. The result of each evaluation is stored in the corresponding point in wave1, making wave1 about 1.5 cycles of a sine wave.

Since wave2 also has 200 points, the wave assignment statement `wave2=wave1*exp(-x/5)` evaluates `wave1*exp(-x/5)` 200 times, once for each point in wave2. In this assignment, the right-hand expression contains a wave, wave1. As Igor executes the assignment, `p` goes from 0 to 199. Each of the 200 times the right side is evaluated, wave1 returns its data value for the corresponding point. The result of each evaluation is stored in the corresponding point in wave2 making wave2 about 1.5 cycles of a damped sine wave.

The effect of a wave assignment statement is to set the data values of the destination wave. Igor does not remember the functional relationship implied by the assignment. In this example, if you changed wave1, wave2 would not change automatically. If you wanted wave2 to have the same functional relationship to wave1 as it had before you changed wave1, you would have to reexecute the `wave2=wave1*exp(-x/5)` assignment.

There is a special kind of wave assignment statement that *does* establish a functional relationship. It should be used sparingly. See **Wave Dependency Formulas** on page II-84 for details.

## More Wave Assignment Features

Just as the symbol `p` returns the current element number in the rows dimension, the symbols `q`, `r` and `s` return the current element number in the columns, layers and chunks dimensions of multidimensional waves. The symbol `x` in the rows dimension has analogs `y`, `z` and `t` in the columns, layers and chunks dimensions. See Chapter II-6, **Multidimensional Waves**, for details.

You can use multiple processors to execute a waveform assignment statement that takes a long time. See **Automatic Parallel Processing with MultiThread** on page IV-323 for details.

The right-hand expression is evaluated in the context of the data folder containing the destination wave. See **Data Folders and Assignment Statements** on page II-111 for details.