

Chapter II-5 — Waves

Example: Comparison Operators and Wave Synthesis

The comparison operators ==, >=, >, <= and < can be useful in synthesizing waves. Imagine that you want to set a wave so that its data values equal $-\pi$ for $x < 0$ and $+\pi$ for $x \geq 0$. The following wave assignment statement accomplishes this:

```
wave1 = -pi*(x<0) + pi*(x>=0)
```

This works because the conditional statements return 1 when the condition is true and 0 when it is false, and then the multiplication proceeds.

You can also make such assignments using the conditional operator:

```
wave0 = (x>0) ? pi : -pi
```

A series of impulses can be made using the **mod** function and ==. This wave equation will assign 5 to every tenth point starting with point 0, and 0 to all the other points:

```
wave1 = (mod(p,10)==0) * 5
```

Example: Wave Assignment and Indexing Using Labels

Dimension labels can be used to refer to wave values by a meaningful name. Thus, for example, you can create a wave to store coefficient values and directly refer to these values by the name of the coefficient (e.g., `coeff[%Friction]`) instead of a potentially confusing and less meaningful numeric index (e.g., `coeff[1]`). You can also view the wave values and labels in a table.

You create wave dimension labels using the **SetDimLabel** operation (see page V-838); for details see **Dimension Labels** on page II-93. Dimension labels may be up to 255 bytes in length; if you use liberal names, such as those containing spaces, make certain to enclose these names within single quotation marks.

Prior to Igor Pro 8.00, dimension labels were limited to 31 bytes. If you use long dimension labels, your wave files and experiments will require Igor Pro 8.00 or later.

In this example we create a wave and use the **FindPeak** operation (see page V-247) to get peak parameters of the wave. Next we create an output parameter wave with appropriate labels and then assign the Find-Peak results to the output wave using the labels.

```
// Make a wave and get peak parameters
Make test=sin(x/30)
FindPeak/Q test

// Create a wave with appropriate row labels
Make/N=6 PeakResult
SetDimLabel 0,0,'Peak Found', PeakResult
SetDimLabel 0,1,PeakLoc, PeakResult
SetDimLabel 0,2,PeakVal, PeakResult
SetDimLabel 0,3,LeadingEdgePos, PeakResult
SetDimLabel 0,4,TrailingEdgePos, PeakResult
SetDimLabel 0,5,'Peak Width', PeakResult

// Fill PeakResult wave with FindPeak output variables
PeakResult[%'Peak Found']    =V_flag
PeakResult[%PeakLoc]          =V_PeakLoc
PeakResult[%PeakVal]          =V_PeakVal
PeakResult[%LeadingEdgePos]   =V_LeadingEdgeLoc
PeakResult[%TrailingEdgePos] =V_TrailingEdgeLoc
PeakResult[%'Peak Width']    =V_PeakWidth

// Display the PeakResult values and labels in a table
Edit PeakResult.ld
```

In addition to the method illustrated above, you can also create and edit dimension labels by displaying the wave in a table and showing the dimension labels with the data. See **Showing Dimension Labels** on page II-235 for further details on using tables with labels.