In this example, we use a subrange of wave4 as the destination of our wave assignment statements. The right-hand expressions index the appropriate values of wave1, wave2 and wave3. Remember that p ranges over the points being evaluated in the destination. So, p ranges from 0 to 99 in the first assignment, from 100 to 199 in the second assignment and from 200 to 299 in the third assignment. In each of the assignments, the wave on the right-hand side has only 100 points, from point 0 to point 99. Therefore we must offset p on the right-hand side to pick out the 100 values of the source wave.

Example: Decomposing Waves

Suppose we have a 300 point wave, wave4, that we want to decompose into three waves of 100 points each: wave1, wave2 and wave3. Here is the sequence of commands to do this.

```
Make/N=100 wave1,wave2,wave3
wave1 = wave4[p]              // Get first third of wave4
wave2 = wave4[p+100]          // Get second third of wave4
wave3 = wave4[p+200]          // Get last third of wave4
```

In this example, we use a subrange of wave4 as the source of our data. We index the desired segment of wave4 using point number indexing. Since wave1, wave2 and wave3 each have 100 points, p ranges from 0 to 99. In the first assignment, we access points 0 to 99 of wave4. In the second assignment, we access points 100 to 199 of wave4. In the third assignment, we access points 200 to 299 of wave4.

You could also use the **Duplicate** operation (see page V-185) to make a wave from a section of another wave.

Example: Complex Wave Calculations

Igor includes a number of built-in functions for manipulating complex numbers and complex waves. These are illustrated in the following example.

We make a time domain waveform and do an FFT on it to generate a complex wave. The example function shows how to pick out the real and imaginary part of the complex wave, how to find the sum of squares and how to convert from rectangular to polar representation. For more information on frequency domain processing, see **Fourier Transforms** on page III-270.

```
Function ComplexWaveCalculations()
   // Make a time domain waveform
   Make/O/N=1024 wave0
   SetScale x 0, 1, "s", wave0            // Goes from 0 to 1 second
   wave0 = sin(2*PI*x) + sin(6*PI*x)/3 + sin(10*PI*x)/5 + sin(14*PI*x)/7
   Display wave0 as "Time Domain"

   // Do the FFT
   FFT/DEST=cwave0 wave0                  // cwave0 is complex, 513 points
   cwave0 /= 512;cwave0[0] /= 2           // Normalize amplitude
   Display cwave0 as "Frequency Domain";SetAxis bottom, 0, 25

   // Calculate magnitude and phase
   Make/O/N=513 mag0, phase0, power0      // These are real waves
   CopyScales cwave0, mag0, phase0, power0
   mag0 = real(r2polar(cwave0))
   phase0 = imag(r2polar(cwave0))
   phase0 *= 180/PI                       // Convert to Degrees
   Display mag0 as "Magnitude and Phase"; AppendToGraph/R phase0
   SetAxis bottom, 0, 25
   Label left, "Magnitude";Label right, "Phase"

   // Calculate power spectrum
   power0 = magsqr(cwave0)
   Display power0 as "Power Spectrum";SetAxis bottom, 0, 25
End
```