

### Overview

Analysis tasks in Igor range from simple experiments using no programming to extensive systems tailored for specific fields. Chapter I-2, **Guided Tour of Igor Pro**, shows examples of the former. WaveMetrics' "Peak Measurement" technical note is an example of the latter.

The Signal Processing chapter covers basic analysis operations with emphasis on signal transformations.

### Fourier Transforms

Igor uses the Fast Fourier Transform (FFT) algorithm to compute a Discrete Fourier Transform (DFT). The FFT is usually called from an Igor procedure as one step in a larger process, such as finding the magnitude and phase of a signal. Igor's FFT uses a prime factor decomposition multidimensional algorithm. Prime factor decomposition allows the algorithm to work on nearly any number of data points. Previous versions of Igor were restricted to a power-of-two number of data points.

This section concentrates on the one-dimensional FFT. See **Multidimensional Fourier Transform** on page II-98 for information on multidimensional aspects of the FFT.

You can perform a Fourier transform on a wave by choosing Analysis→Fourier Transforms. This displays the Fourier Transforms dialog.

Select the type of transform by clicking the Forward or Reverse radio button. Select the wave that you want to transform from the Wave list. If you enable the From Target box under the Wave list, only appropriate waves in the target window will appear in the list.

#### Why Some Waves Aren't Listed

What do we mean by "appropriate" waves?

The data can be either real or complex. If the data are real, the number of data points must be even. This is an artificial limitation that was introduced in order to guarantee that the inverse transform of a forward-transformed wave is equal to the original wave. For multidimensional data, only the number of rows must be even. You can work around some of the restrictions of the inverse FFT with the command line.

The inverse FFT requires complex data. There are no restrictions on the number of data points. However, for historic and compatibility reasons, certain values for the number of points are treated differently as described in the next sections.

#### Changes in Wave Type and Number of Points

If the wave is a 1D real wave of  $N$  points ( $N$  must be even), the FFT operation results in a complex wave consisting of  $N/2+1$  points containing the "one-sided spectrum". The negative spectrum is not computed, since it is identical for real input waves.

If the wave is complex (even if the imaginary part is zero), its data type and number of points are unchanged by the forward FFT. The FFT result is a "two-sided spectrum", which contains both the positive *and* the negative frequency spectra, which are different if the imaginary part of the complex input data is nonzero.

The diagram below shows the two-sided spectrum of 128-point data containing a zero imaginary component.