$$f(t) = \sum_a \sum_b c_{ab} \psi_{ab}(t),$$

where the two-parameter expansion coefficients are given by

$$c_{ab} = \int f(t) \psi_{ab}(t) dt$$

and the wavelets obey the condition

$$\psi_{ab}(t) = 2^{\frac{a}{2}} \Psi(2^a t - b).$$

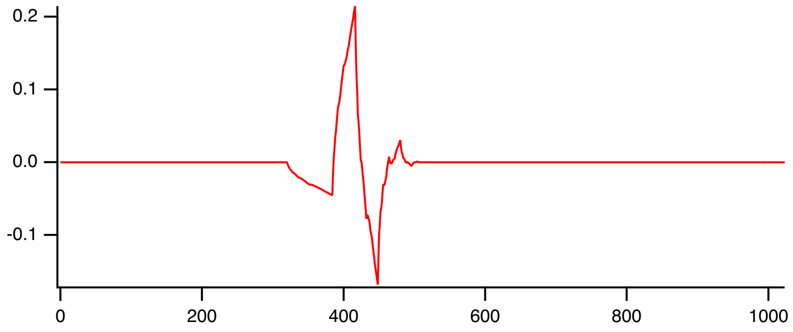
Here Ψ is the mother wavelet, *a* is the dilation parameter and *b* is the offset parameter.

The two parameter representation can complicate things quickly as one goes from 1D signal to higher dimensions. In addition, because the number of coefficients in each scale varies as a power of 2, the DWT of a 1D signal is not conveniently represented as a 2D image (as is the case with the CWT). It is therefore customary to "pack" the results of the transform so that they have the same dimensionality of the input. For example, if the input is a 1D wave of 128 (=27) points, there are 7-1=6 significant scales arranged as follows:

Scale	Storage Location
1	64-127
2	32-63
3	16-31
4	8-15
5	4-7
6	2-3

An interesting consequence of the definition of the DWT is that you can find out the shape of the wavelet by transforming a suitable form of a delta function. For example:

```
Make/N=1024 delta=0
delta[22]=1
DWT/I delta
Display W_DWT     // Daubechies 4 coefficient wavelet
```



Convolution

You can use convolution to compute the response of a linear system to an input signal. The linear system is defined by its impulse response. The convolution of the input signal and the impulse response is the output signal response. Convolution is also the time-domain equivalent of filtering in the frequency domain.

Smoothing is also a form of convolution – see **Smoothing** on page III-292.

The **FilterFIR** implements convolution in the time domain – see **Digital Filtering** on page III-299.

Igor implements general convolution with the **Convolve** operation. To use the Convolve operation, choose Analysis→Convolve.

The built-in Convolve operation computes the convolution of two waves named "source" and "destination" and overwrites the destination wave with the results. The operation can also convolve a single source wave with multiple destination waves (overwriting the corresponding destination wave with the results in each case). The Convolve dialog allows for more flexibility by preduplicating the second waves into new destination waves.

If the source wave is real-valued, each destination wave must be real-valued and if source wave is complex, each destination wave must be complex, too. Double and single precision waves may be freely intermixed; the calculations are performed in the higher precision.

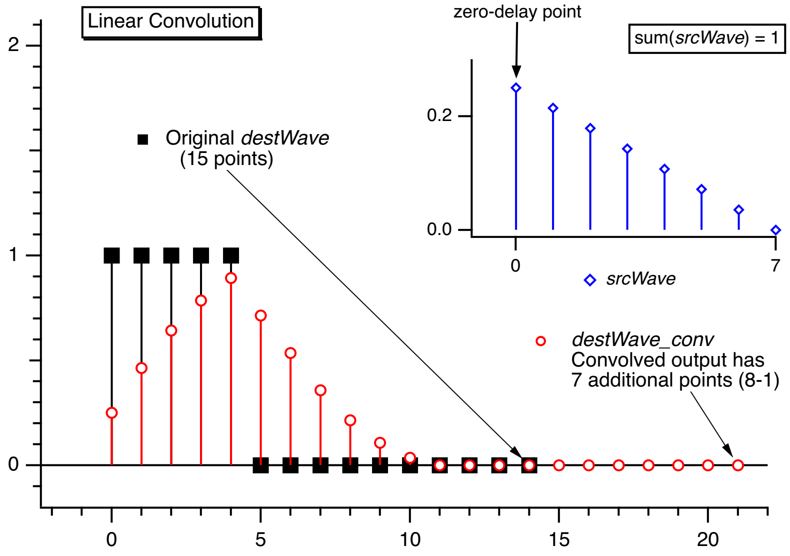
Convolve combines neighboring points before and after the point being convolved, and at the ends of the waves not enough neighboring points exist. This is a general problem in any convolution operation; the smoothing operations use the End Effect pop-up to determine what to do. The Convolve dialog presents three algorithms in the Algorithm group to deal with these missing points.

The Linear algorithm is similar to the Smooth operation's Zero end effect method; zeros are substituted for the values of missing neighboring points.

The Circular algorithm is similar to the Wrap end effect method; this algorithm is appropriate for data which is assumed to endlessly repeat.

The acausal algorithm is a special case of Linear which eliminates the time delay that Linear introduces.

Depending on the algorithm chosen, the number of points in the destination waves may increase by the number of points in the source wave, less one. For linear and acausal convolution, the destination wave is first zero-padded by one less than the number of points in the source wave. This prevents the "wrap-around" effect that occurs in circular convolution. The zero-padded points are removed after acausal convolution, and retained after linear convolution.



Use linear convolution when the source wave contains an impulse response (or filter coefficients) where the first point of *srcWave* corresponds to no delay ($t = 0$).

Use Circular convolution for the case where the data in the source wave and the destination waves are considered to endlessly repeat (or "wrap around" from the end back to the start), which means no zero padding is needed.