References

A. Hyvarinen and E. Oja (2000) Independent Component Analysis: Algorithms and Applications, Neural Networks, (13)411-430.

See Also

PCA

if-elseif-endif

```
if ( <expression1> )
   <TRUE part 1>
elseif ( <expression2> )
   <TRUE part 2>
[…]
[else
   <FALSE part>]
endif
```

In an if-elseif-endif conditional statement, when an expression first evaluates as TRUE (nonzero), then only code corresponding to the TRUE part of that expression is executed, and then the conditional statement is exited. If all expressions evaluate as FALSE (zero) then *FALSE part* is executed when present. After executing code in any TRUE part or the FALSE part, execution will next continue with any code following the if-elseif-endif statement.

See Also

If-Elseif-Endif on page IV-40 for more usage details.

if-endif

```
if ( <expression> )
   <TRUE part>
[else
   <FALSE part>]
endif
```

An if-endif conditional statement evaluates *expression*. If *expression* is TRUE (nonzero) then the code in *TRUE part* is executed, or if FALSE (zero) then the optional *FALSE part* is executed.

See Also

If-Else-Endif on page IV-40 for more usage details.s

IFFT

IFFT [*flags*] ***srcWave***

The IFFT operation calculates the Inverse Discrete Fourier Transform of *srcWave* using a multidimensional fast prime factor decomposition algorithm. This operation is the inverse of the **FFT** operation.

Output Wave Name

For compatibility with earlier versions of Igor, if you use IFFT without /ROWS or /COLS, the operation overwrites *srcWave*.

If you use the /ROWS flag, IFFT uses the default output wave name M_RowFFT and if you use the /COLS flag, IFFT uses the default output wave name M_ColFFT.

We recommend that you use the /DEST flag to make the output wave explicit and to prevent overwriting *srcWave*.

Parameters

srcWave is a complex wave. The IFFT of *srcWave* is a either a real or complex wave, according to the length and flags.

Flags

/C	Forces the result of the IFFT to be complex. Normally, the IFFT produces a real result unless certain special conditions are detected as described in **Details**.

/COLS — Computes the 1D IFFT of 2D *srcWave* one column at a time, storing the results in the destination wave. You must specify a destination wave using the /DEST flag (no other flags are allowed). See the /ROWS flag and corresponding flags of the **FFT** operation.

/DEST=*destWave* — Specifies the output wave created by the IFFT operation.

It is an error to specify the same wave as both *srcWave* and *destWave*.

In a function, IFFT by default creates a real wave reference for the destination wave. See **Automatic Creation of WAVE References** on page IV-72 for details.

/FREE — Creates *destWave* as a free wave.

/FREE is allowed only in functions and only if *destWave*, as specified by /DEST, is a simple name or wave reference structure field.

See **Free Waves** on page IV-91 for more discussion.

The /FREE flag was added in Igor Pro 7.00.

/R — Forces real output when, due to a power of 2 number of points, IFFT would otherwise automatically produce a complex result.

/ROWS — Calculates the IFFT of only the first dimension of 2D *srcWave*. It computes the 1D FFT one row at a time. You must specify a destination wave using the /DEST flag (no other flags are allowed). See the /COLS flag and corresponding flags of the **FFT** operation.

/Z — Will not rotate *srcWave* when computing the IDFT of a complex wave whose length is an integral power of 2.

This length indicates that the Inverse DFT result will also be a complex wave. When the result is complex, *and* the x scaling of *srcWave* is such that the first point is *not* x=0, it normally rotates *srcWave* by -N/2 points before performing the IFFT. This inverts the process of performing an FFT on a complex wave. However when /Z is specified, it does not perform this rotation.

Details

The data type of *srcWave* must be complex and must not be an integer type. You should be aware that an IFFT on a number of points that is prime can be slow.

By default, IFFT assumes you are performing an inverse transform on data that was originally real and therefore it produces a real result. However, for historical and compatibility reasons, IFFT detects the special conditions of a one-dimensional wave containing an integral power of 2 data points and automatically creates a complex result.

When the result is complex, the number of points (N) in the resulting wave will be of the same length. Otherwise the resulting wave will be real and of length (N-1)*2.

In either the complex or real case the X units of the output wave are changed to "s". The X scaling also is changed appropriately, cancelling out the adjustments made by the **FFT** operation. When the data is multidimensional, the same considerations apply to the additional dimensions. The scaling description and IDFT equation below pretend that the IFFT is not performed in-place. After computing the IFFT values, the X scaling of *waveOut* is changed as if Igor had executed these commands:

```
Variable points                  // time-domain points, N_timeDomain
if( waveIn was complex wave )
   points= numpnts(waveIn)
else                             // waveIn was real wave
   points= (numpnts(waveIn) - 1) * 2
endif
Variable deltaT= 1 / (points*deltaX(waveIn))      // 1/(N_timeDomain dx)
SetScale/P waveOut 0,deltaT,"s"
```

The IDFT equation is:

$$waveOut[n] = \frac{1}{N}\sum_{k=0}^{N-1} waveIn[k]\exp\left(\frac{2\pi ikn}{N}\right), \quad where \quad i=\sqrt{-1}.$$