WaveTransform

WaveTransform [***flags***] ***keyword srcWave***

The WaveTransform operation transforms *srcWave* in various ways. If the /O flag is not specified then unless otherwise indicated the output is stored in the wave W_WaveTransform, which will be of the same data type as *srcWave* and saved in the current data folder.

Parameters

keyword is one of the following:

abs	Calculates the absolute value of the entries in *srcWave*. It stores results in W_Abs if *srcWave* is 1D or M_Abs otherwise. It will overwrite *srcWave* when used with the /O flag. *srcWave* must be single or double precision real wave.
acos	Calculates the inverse cosine of the entries in *srcWave*. It stores results in W_Acos if *srcWave* is 1D or M_Acos otherwise. It will overwrite *srcWave* when used with the /O flag. *srcWave* must be single or double precision real wave.
asin	Calculates the inverse sine of the entries in *srcWave*. It stores results in W_Asin if *srcWave* is 1D or M_Asin otherwise. It will overwrite *srcWave* when used with the /O flag. *srcWave* must be single or double precision real wave.
atan	Calculates the inverse tangent of the entries in *srcWave*. It stores results in W_Atan if *srcWave* is 1D or M_Atan otherwise. It will overwrite *srcWave* when used with the /O flag. *srcWave* must be single or double precision real wave.
cconjugate	Calculates the complex conjugate of *srcWave*. Stores results in W_CConjugate or M_CConjugate, depending on wave dimensionality, or overwrites *srcWave* if /O is used.
cos	Calculates the cosine of the entries in *srcWave*. It stores results in W_Cos if *srcWave* is 1D or M_Cos otherwise. It will overwrite *srcWave* when used with the /O flag. *srcWave* must be single or double precision real wave.
crystalToRect	Converts triplet (three column {x,y,z}) waves from nonorthogonal crystallographic coordinates to rectangular cartesian system. The parameters provided in the /P flag are the crystallographic definition of the coordinate system given by {a, b, c, alpha, beta, gamma}. The three angles are assumed to be expressed in radians unless the /D flag is specified. The transformation sets the first component parallel to the vector a and the third component parallel to c*. The output is stored in the current data folder in the wave M_CrystalToRect which has the same data type. If the /O flag is specified, the output overwrites the original data.
flip	Flips the data in *srcWave* about its center. If /O flag is used, *srcWave* is overwritten. Otherwise a new wave is created in the current data folder. The wave is named W_flipped or M_flipped according to the dimensionality of *srcWave*.
index	Fills *srcWave* as in `jack=p`. If /P is specified then `jack=p+`*`param1`*. The /O flag does not apply here.
inverse	Computes `1/srcWave[i]` for each point in *srcWave* and stores it in W_Inverse or M_Inverse depending on the dimensionality of *srcWave*.
inverseIndex	Fills *srcWave* as in `jack=numPnts-1-p`. If /P is specified the `jack=numPnts-1-p+`*`param1`*.
magnitude	Creates a real-valued wave that is the magnitude of *srcWave*. If you do not specify the /O flag, the output is stored in W_Magnitude or M_Magnitude depending on the dimensionality of *srcWave*; the output precision will be the same as *srcWave*.
magsqr	Creates a real-valued wave that is the magnitude squared of *srcWave*. If *srcWave* is a double precision complex wave, the output is also double precision, otherwise the output is a single precision wave. Stores the result in wave W_MagSqr or M_MagSqr, depending on the dimensionality of *srcWave*, or overwrites *srcWave* if /O is used.

max	Calculates the maximum of a point in *srcWave* and a fixed number specified as a single parameter with the /P flag. It stores results in W_max if *srcWave* is 1D or M_max otherwise. It will overwrite *srcWave* when used with the /O flag. See also the min keyword and the example below.
min	Calculates the minimum of a point in *srcWave* and a fixed number specified as a single parameter with the /P flag. It stores results in W_min if *srcWave* is 1D or M_min otherwise. It will overwrite *srcWave* when used with the /O flag. See also the max keyword and the example below.
normalizeArea	Calculates the area under the curve and rescales the wave so that the area is 1. Note that waves with negative areas will be rescaled to positive values. Applies to 1D real-valued waves. It does not affect wave scaling. Stores the result in the wave W_normalizedArea or overwrites *srcWave* if /O is used.
phase	Creates a real-valued wave containing the phase of the complex input wave. If the /O flag is not used, the output is stored in W_Phase or M_Phase depending on the dimensionality of *imageMatrix*. You can also use /P={*norm*} to divide the output wave by the value of *norm*.
rectToCrystal	Converts triplet (three column {x,y,z}) waves from cartesian coordinates to nonorthogonal crystallographic coordinate system. The parameters provided in the /P flag are the crystallographic definition of the coordinate system given by {a, b, c, alpha, beta, gamma}. The three angles are assumed to be expressed in radians unless the /D flag is specified. The transformation assumes the first component parallel to the vector a and the third component parallel to c*. The output is stored in the current data folder in the wave M_RectToCrystal which has the same data type. If the /O flag is specified, the output overwrites the original data.
setConstant	Sets *srcWave* points to a constant value specified by the /V flag. This keyword applies to real, numeric waves only. You can use /R with setConstant to set a subset of a wave. setConstant was added in Igor Pro 7.00.
setZero	Sets all *srcWave* points to zero. setZero was added in Igor Pro 7.00.
sgn	Sets the value to -1 if the entry is negative, 1 otherwise. Stores the results in W_Sgn or overwrites *srcWave* if /O is used. This operation will not work on UNSIGNED waves.
shift	Shifts the position of data in *srcWave* by the specified number of points. Unlike Rotate, WaveTransform discards data points that shift outside existing wave boundaries. After the shift, vacated wave points are set to the specified *fillValue*. The shift and the *fillValue* are specified with the /P flag using the syntax: /P={*numPoints*, *fillValue*}. If you do not provide a fill value, it will be 0 for integer waves and NaN for SP and DP.
sin	Calculates the sine of the entries in srcWave. Stores results in W_Sin if *srcWave* is 1D or M_Sin otherwise. Overwrites *srcWave* when used with the /O flag. *srcWave* must be a real single or double precision floating point wave.
sqrt	Calculates the square root of the entries in *srcWave*. It stores results in W_sqrt if *srcWave* is 1D or M_sqrt otherwise. It will overwrite *srcWave* when used with the /O flag. *srcWave* must be single- or double-precision real wave.
tan	Calculates the tangent of the entries in *srcWave*. The results are stored in W_tan if *srcWave* is 1D or M_tan otherwise. It will overwrite *srcWave* when used with the /O flag. *srcWave* must be single- or double-precision real wave.
zapINFs	Deletes elements whose value is infinity or -infinity. This is relevant for 1D single-precision and double-precision floating point waves only and does nothing for other types of 1D waves. It is not suitable for multidimensional waves and returns an error if *srcWave* is multidimensional. Use **MatrixOp** `replace` for multidimensional waves.