To interpolate a 3D scalar distribution that is not sampled on a regular lattice, *srcWave* is a four column 2D wave where the columns correspond to *x*, *y*, *z*, f(*z*, *y*, *z*), respectively. You must also use a "triangulation" wave for *srcWave* (use `Triangulate3D/out=1` to obtain the triangulation wave). If P falls within the convex domain defined by the tetrahedra in *triangulationWave*, the function returns the barycentric linear interpolation for P using the tetrahedron where P is found. If P is outside the convex domain the function returns NaN.

Examples

```
Make/O/N=(10,20,30) ddd=gnoise(10)
Print interp3D(ddd,1,0,0)
Print interp3D(ddd,1,1,1)
```

```
Make/O/N=(10,4) ddd=gnoise(10)
Triangulate3D/OUT=1 ddd
Print interp3D(ddd,1,0,0,M_3DVertexList)
Print interp3D(ddd,1,1,1,M_3DVertexList)
```

See Also

The **Interpolate3D** operation. **Interpolation** on page III-114.

Interp3DPath

Interp3DPath *3dWave tripletPathWave*

The Interp3DPath operation computes the trilinear interpolated values of *3dWave* for each position specified by a row of in *tripletPathWave*, which is a 3 column wave in which the first column represents the X coordinate, the second represents the Y coordinate and the third represents the Z coordinate. Interp3DPath stores the resulting interpolated values in the wave W_Interpolated. Interp3DPath is equivalent to calling the Interp3D() function for each row in *tripletPathWave* but it is computationally more efficient.

If the position specified by the *tripletPathWave* is outside the definition of the *3dWave* or if it contains a NaN, the operation stores a NaN in the corresponding output entry.

Both *3dWave* and *tripletPathWave* can be of any numeric type. W_Interpolated is always of type NT_FP64.

See Also

The **ImageInterpolate** operation and the **Interp3D** and **interp** functions. **Interpolation** on page III-114.

Interpolate2

Interpolate2 [*flags*] [*xWave*,] *yWave*

The Interpolate2 operation performs linear, cubic spline and smoothing cubic spline interpolation on 1D waveform or XY data. It produces output in the form of a waveform or an XY pair.

The cubic spline interpolation is based on a routine from "Numerical Recipes in C".

The smoothing spline is based on "Smoothing by Spline Functions", Christian H. Reinsch, *Numerische Mathematic* 10, 177-183 (1967).

For background information, see **The Interpolate2 Operation** on page III-115.

Parameters

xWave specifies the wave which supplies the X coordinates for the input curve. If you omit it, X coordinates are taken from the X values of *yWave*.

yWave specifies the wave which supplies the Y coordinates for the input curve.

Flags

/A=*a* Controls pre-averaging. Pre-averaging is deprecated - use the smoothing spline (/T=3) instead.

If *a* is zero, Interpolate2 does no pre-averaging. If *a* is greater than one, it specifies the number of nodes through which you want the output curve to go. Interpolate2 creates the nodes by averaging the raw input data.

Pre-averaging does not work correctly with the log-spaced output mode (/I=2). This is because the pre-averaging is done on linearly-spaced intervals but the input data is log-spaced.

/E=*e* Controls how the end points are determined for cubic spline interpolation only.

e=1: Match first derivative (default)

e=2: Match second derivative (natural)

/F=*f* *f* is the smoothing factor used for the smoothing spline.

f=0 is nearly identical to the cubic spline.

f>0 gives increasing amounts of smoothing as f increases.

See **Smoothing Spline Parameters** on page III-119 for details.

/FREE Creates output waves as free waves (see **Free Waves** on page IV-91).

/FREE is allowed only in functions. If you use /FREE then the output waves specified by /X and /Y must be either simple names or valid wave references.

/FREE was added in Igor Pro 9.00.

/I[=*i*] Determines at what X coordinates the interpolation is done.

i=0: Gives output values at evenly-spaced X coordinates that span the X range of the input data. This is the default setting if /I is omitted.

i=1: Same as *i*=0 except that the X input values are included in the list of X coordinates at which to interpolate. This is rarely needed and is not available if no X destination wave is specified. /I is equivalent to /I=1. Both are not recommended.

i=2: Gives output values at X coordinates evenly-spaced on a logarithmic scale. Ignores any non-positive values in your input X data.

This mode is not recommended. See **Interpolating Exponential Data** on page III-118 for an alternative.

i=3: Gives output values at X coordinates that you specify by setting the X coordinates of the destination wave before calling Interpolate2. You must create your destination wave or waves before doing the

If you omit /X=*xDest* then the X coordinates come from the X values of the output waveform designated by /Y=*yDest*.

If you include /X=*xDest* then the X coordinates come from the data values of the specified X output wave.

When using /I=3, the number of output points is determined by the destination wave and the /N flag is ignored.

See **Destination X Coordinates from Destination Wave** on page III-119 for further details.

/J=*j* Controls the use of end nodes with pre-averaging (/A). Pre-averaging is deprecated - use the smoothing spline (/T=3) instead.

j=0: Turns end nodes off.

j=1: Creates end nodes by cubic extrapolation.

j=2: Creates end nodes equal to the first and last data points of the input data set, not counting points that contain NaNs or INFs.

/N=*n* Controls the number of points in the output wave or waves. *n* defaults to the larger of 200 and the number of points in the source waves. This value is ignored if you /I=3 (X from dest mode).

/S=*s* *s* is the estimated standard deviation of the noise of the Y data. It is used for the smoothing spline only. *s* is used as the estimated standard deviation for all points in the Y data.

If neither /S nor /SWAV are present, Interpolate2 arbitrarily assumes an *s* equal to .05 times the amplitude of the Y data.

/SWAV=*stdDevWave*