The output wave M_UnwrappedPhase has the same wave scaling and dimension units as srcWave. The unwrapped phase is units of cycles; you will have to multiply it by 2π if you need the results in radians.

The operation creates two variables:

V_numResidues	Number of residues encountered(if using /M=1).
V_numRegions	Number of independent phase regions. In Goldstein's method the regions are bounded by branch cuts, but in Itoh's method they depend on the content of the ROI wave.

Examples

```
// Unwrap the phase of a complex wave wCmplx
MatrixOP/O phaseWave=atan2(imag(wCmplx),real(wCmplx))/(2*pi)
ImageUnwrapPhase/M=1 srcWave=phaseWave

// Find the locations of positive residues in the phase
ImageUnwrapPhase/M=1/L srcWave=phaseWave
MatrixOP/O ee=greater(bitAnd(M_PhaseLUT,2),0)

// Find the branch cuts
MatrixOP/O bc=greater(bitAnd(M_PhaseLUT,8),0)
```

See Also

The **Unwrap** operation and the **mod** function.

References

The following reference is an excellent text containing in-depth theory and detailed explanation of many two-dimensional phase unwrapping algorithms:

Ghiglia, Dennis C., and Mark D. Pritt, *Two Dimensional Phase Unwrapping — Theory, Algorithms and Software*, Wiley, 1998.

ImageWindow

`ImageWindow [/I/O/P=param] method srcWave`

The ImageWindow operation multiplies the named waves by the specified windowing method.

ImageWindow is useful in preparing an image for FFT analysis by reducing FFT artifacts produced at the image boundaries.

Parameters

srcWave	Two-dimensional wave of any numerical type. See **WindowFunction** for windowing one-dimensional data.
method	Selects the type of windowing filter. See **ImageWindow Methods** on page V-436.

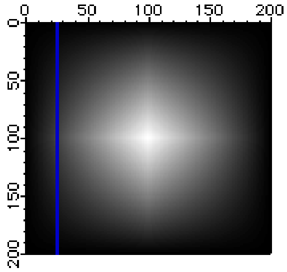
Flags

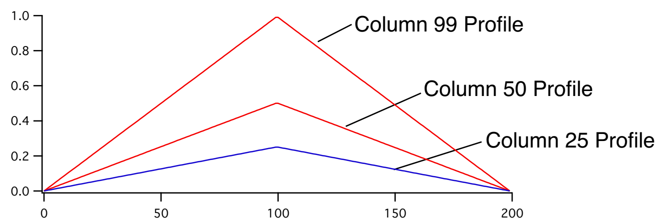
/I	Creates only the output wave containing the windowing filter values that are used to multiply each pixel in *srcWave*. It does not filter the source image.
/O	Overwrites the source image with the output image. If /O is not used then the operation creates the M_WindowedImage wave containing the filtered source image.
/P=*param*	Specifies the design parameter for the Kaiser window.

Details

The 1-dimensional window for each column is multiplied by the value of the corresponding row's window value. In other words, each point is multiplied by the both the row-oriented and column-oriented window value.

This means that all four edges of the image are decreased while the center remains at or near its original value. For example, applying the Bartlett window to an image whose values are all equal results in a trapezoidal pyramid of values:





The default output wave is created with the same data type as the source image. Therefore, if the source image is of type unsigned byte (/b/u) the result of using /I will be identically zero (except possibly for the middle-most pixel). If you keep in mind that you need to convert the source image to a wave type of single or double precision in order to perform the FFT, it is best if you convert your source image (e.g., `Redimension/S` *`srcImage`*) before using the ImageWindow operation.

The windowed output is in the M_WindowedImage wave unless the source is overwritten using the /O flag.

The necessary normalization value (equals to the average squared window factor) is stored in V_value.

ImageWindow Methods

This section describes the supported keywords for the *method* parameter. In all equations, L is the array width and n is the pixel number.

Hanning:

$$w(n) = \frac{1}{2}\left[1-\cos\left(\frac{2\pi n}{L-1}\right)\right] \quad 0 \le n \le L-1$$

Hamming:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{L-1}\right) \quad 0 \le n \le L-1$$

Bartlet: Synonym for Bartlett.

Bartlett:

$$w(n) = \begin{cases} \dfrac{2n}{L-1} & 0 \le n \le \dfrac{L-1}{2} \\ 2-\dfrac{2n}{L-1} & \dfrac{L-1}{2} \le n \le L-1 \end{cases}$$

Blackman:

$$w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{L-1}\right) + 0.08\cos\left(\frac{4\pi n}{L-1}\right)$$

$$0 \le n \le L-1$$