Details

The MultiTaperPSD operation estimates the PSD of *srcWave* by computing a set of discrete prolate spheroidal functions (Slepian **DPSS**) and using them as optimal window functions. The window functions/tapers are applied to the input signal and squares of the resulting Fourier transforms are weighted together to produce the PSD estimate.

srcWave must be a real-valued numeric wave of single or double precision and must not contain any INFs or NaNs.

The mean value of the input is subtracted prior to multiplication by the tapers. Like DSPPeriodogram, the MultiTaperPSD operation leaves the normalization to the user.

The default PSD estimate is calculated by combining the Fourier transforms of the tapered signal with weights from the DPSS calculation. You can use the /A flag to improve the PSD estimate for increasing tapers. Thomson's adaptive algorithm is reasonably efficient and also provides an estimate of the effective degrees of freedom as a function of frequency.

The operation sets the variable V_Flag to zero if successful or to a -1 if it encounters an error. If you are using Thomson's adaptive algorithm (/A) V_Flag is set to the number of frequencies at which the algorithm failed to converge.

See Also

FFT, **DSPPeriodogram**, **DPSS**, **ImageWindow**, **Hanning**, **LombPeriodogram**

Demos

See the “MultiTaperPSD Demo” example experiment.

References

D.J. Thomson: "Spectrum Estimation and Harmonic Analysis", Proc. IEEE 70 (9) 1982 pp. 1055.

D. Slepian, "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty -- V: the Discrete Case", Bell System Tech J. Vol 57 (5) May-June 1978.

Lees, J. M. and J. Park (1995). Multiple-taper spectral analysis: A stand-alone C-subroutine: Computers & Geosciences: 21, 199-236.

MultiThread

MultiThread `[ /N=`***`numThreads`***` ]` *wave* = *expression*

In user-defined functions, the MultiThread keyword can be inserted in front of wave assignment statements to speed up execution on multiprocessor computer systems.

Warning: Misuse of this keyword can result in a performance penalty or even a crash. Be sure to read **Automatic Parallel Processing with MultiThread** on page IV-323 before using MultiThread.

The expression must be thread-safe. This means that if it calls a function, the function must be thread-safe. This goes for both built-in and user-defined functions.

Not all built-in functions are thread-safe. Use the Command Help tab in the Igor Help Browser to see which functions are thread-safe.

User-defined functions are thread-safe if they are defined using the **ThreadSafe** keyword. See **ThreadSafe Functions** on page IV-106 for details.

You can specify the number of threads using the /NT flag. The default value, which takes effect if you omit /NT or specify /NT=0, uses the number returned by **ThreadProcessorCount**. /NT=1 specifies a single thread and is equivalent to omitting the MultiThread keyword; you might want to turn off threading to avoid its overhead when the wave has few points. *numThreads* values greater than one specify the desired number of threads to be used. Igor may use fewer threads depending on how it is able to partition the task. The /NT flag was added in Igor Pro 8.00.

See Also

Automatic Parallel Processing with MultiThread on page IV-323.

Waveform Arithmetic and Assignments on page II-74.