Wolfram: https://mathworld.wolfram.com/OsculatingCircle.html

Ming-Kuang Hsu, Jiun-Chyuan Sheu, and Cesar Hsue, "Overcoming the Negative Frequencies - Instantaneous Frequency and Amplitude Estimation using Oculating Circle Method, *Journal of Marine Science and Technology*, Vol 19, No 5, pp. 514-521, 2011.

See Also

STFT, **HilbertTransform**, **DSPPeriodogram**

Int

`int localName`

In a user-defined function or structure, declares a local 32-bit integer in IGOR32, a local 64-bit integer in IGOR64.

Int is available in Igor Pro 7 and later. See **Integer Expressions** on page IV-38 for details.

See Also

Int64, **UInt64**

Int64

`int64 localName`

Declares a local 64-bit integer in a user-defined function or structure.

Int64 is available in Igor Pro 7 and later. See **Integer Expressions** on page IV-38 for details.

See Also

Int, **UInt64**

Integrate

```
Integrate [type flags][flags] yWaveA [/X = xWaveA][/D = destWaveA]
    [, yWaveB [/X = xWaveB][/D = destWaveB][, …]]
```

The Integrate operation calculates the 1D numeric integral of a wave. X values may be supplied by the X-scaling of the source wave or by an optional X wave. Rectangular integration is used by default.

Integrate is multi-dimension-aware in the sense that it computes a 1D integration along the dimension specified by the /DIM flag or along the rows dimension if you omit /DIM.

Complex waves have their real and imaginary components integrated individually.

Flags

/DIM= *d* — Specifies the wave dimension along which to integrate when *yWave* is multidimensional.

d=-1:	Treats entire wave as 1D (default).
d=0:	Integrates along rows.
d=1:	Integrates along columns.
d=2:	Integrates along layers.
d=3:	Integrates along rows.

For example, for a 2D wave, /DIM=0 integrates each row and /DIM=1 integrates each column.

/METH=*m* — Sets the integration method.

m=0:	Rectangular integration (default). Results at a point are stored at the same point (rather than at the next point as for /METH=2). This method keeps the dimension size the same.
m=1:	Trapezoidal integration.
m=2:	Rectangular integration. Results at a point are stored at the next point (rather than at the same point as for /METH=0). This method increases the dimension size by one to provide a place for the last bin.

/P	Forces point scaling.
/T	Trapezoidal integration. Same as /METH=1.

Type Flags *(used only in functions)*

Integrate also can use various type flags in user functions to specify the type of destination wave reference variables. These type flags do not need to be used except when it needed to match another wave reference variable of the same name or to identify what kind of expression to compile for a wave assignment. See **WAVE Reference Types** on page IV-73 and **WAVE Reference Type Flags** on page IV-74 for a complete list of type flags and further details.

For example, when the input (and output) waves are complex, the output wave will be complex. To get the Igor compiler to create a complex output wave reference, use the /C type flag with /D=destwave:

```
Make/O/C cInput=cmplx(sin(p/8), cos(p/8))
Make/O/C/N=0 cOutput
Integrate/C cInput /D=cOutput
```

Wave Parameters

Note: ***All*** wave parameters must follow *yWave* in the command. All wave parameter flags and type flags must appear immediately after the operation name (`Integrate`).

/D=*destWave*	Specifies the name of the wave to hold the integrated data. It creates *destWave* if it does not already exist or overwrites it if it exists.
/X=*xWave*	Specifies the name of corresponding X wave. For rectangular integration, the number of points in the X wave must be one greater than the number of elements in the Y wave dimension being integrated.

Details

The computation equation for rectangular integration using /METH=0 is:

$$waveOut[p] = \sum_{i=0}^{p} waveIn[i] \cdot \Delta x.$$

The computation equation for rectangular integration using /METH=2 is:

$$waveOut[0] = 0$$

$$waveOut[p+1] = \sum_{i=0}^{p} (x_{i+1} - x_i) waveIn[i].$$

The inverse of this rectangular integration is the backwards difference.

Trapezoidal integration (/METH=1) is a more accurate method of computing the integral than rectangular integration. The computation equation is:

$$waveOut[0] = 0$$

$$waveOut[p] = waveOut[p-1] + \frac{\Delta x}{2}\left(waveIn[p-1] + waveIn[p]\right).$$

If the optional /D = *destWave* flag is omitted, then the wave is integrated in place overwriting the source wave.

When using an X wave, the X wave must be a 1D wave with data type matching the Y wave (excluding the complex type flag). Rectangular integration (/METH=0 or 2) requires an X wave having one more point than the number of elements in the dimension of the Y wave being integrated. X waves with number points plus one are allowed for rectangular integration with methods needing only the number of points. X waves are not used with integer source waves.