APMath

APMath [***flags***] ***destStr = Expression***

The APMath operation provides arbitrary precision calculation of basic mathematical expressions. It converts the final result into the assigned string *destStr*, which can then be printed or used to represent a value (at the given precision) in another APMath operation.

Parameters

destStr	Specifies a destination string for the assignment expression. If *destStr* is not an existing variable, it is created by the operation. When executing in a function, *destStr* will be a local variable if it does not already exist.
Expression	Algebraic expression containing constants, local, global, and reference variables or strings, as well as wave elements together with the operators shown below.

APMath Operators

+	Scalar addition	Lowest precedence
-	Scalar subtraction	Lowest precedence
*	Scalar multiplication	Medium precedence
/	Scalar division	Medium precedence
^	Exponentiation	Highest precedence

APMath Functions on Scalar Parameters

The following functions are supported for arbitrary precision math on scalar parameters:

sqrt(*x*)	Square root of *x*.
cbrt(*x*)	Cube root of *x*.
pi	Value of π (without parentheses).
sin(*x*)	Sine of *x*.
cos(*x*)	Cosine of *x*.
tan(*x*)	Tangent of *x*.
asin(*x*)	Inverse sine of *x*.
acos(*x*)	Inverse cosine of *x*.
atan(x)	Inverse tangent of *x*.
atan2(*y*,*x*)	Inverse tangent of *y*/*x*.
log(*x*)	Logarithm of *x*.
log10(*x*)	Logarithm based 10 of *x*.
exp(*x*)	Exponential function e^*x*.
pow(*x*,*n*)	*x* to the power *n* (*n* not necessarily integer).
sinh(*x*)	Hyperbolic sine of *x*.
cosh(*x*)	Hyperbolic cosine of *x*.
tanh(*x*)	Hyperbolic tangent of *x*.
asinh(*x*)	Inverse hyperbolic sine of *x*.
acosh(*x*)	Inverse hyperbolic cosine of *x*.
atanh(*x*)	Inverse hyperbolic tangent of *x*.

`ceil(x)` Smallest integer larger than *x*.

`comp(x,y)` Returns 0 for *x* == *y*, 1 if *x* > *y* and -1 if *y* > *x*.

`factorial(n)` Factorial of integer *n*.

`floor(x)` Greatest integer smaller than *x*.

`gcd(x,y)` Greatest common divisor of *x* and *y*.

`lcd(x,y)` Lowest common denominator of *x* and *y* (given by *x***y*/gcd(*x*,*y*).

`sgn(x)` Sign of *x* or zero if *x* == 0.

`binomial(n,k)` Returns the the binomial function for integers n and k.

`Bernoulli(n)` Returns the Bernoulli number Bn (with Bn(1)=-1/2).

`Stirling2(n,k)` Returns the Stirling number of the second kind.

APMath Functions on Wave Parameters

The following functions are supported for arbitrary precision math on waves.

These restrictions apply to all of these APMath functions on waves:

1. The parameter *w* must be a simple wave reference. It can not be a data folder path to a wave or a $ expression pointing to a wave.
2. The wave can be a real numeric wave or a text wave containing arbitrary precision numbers in string form.
3. Complex waves are not allowed.
4. 64-bit integer waves are not allowed.
5. The functions return an error if the wave contains NaNs or INFs.
6. Multidimensional waves are treated as 1D.

`kurtosis(w)` Returns the kurtosis of the entire wave *w*. See **WaveStats** for a discussion of kurtosis. The kurtosis function was added in Igor Pro 8.00.

`mean(w)` Returns the mean of the entire wave *w*. The mean function was added in Igor Pro 8.00.

`skew(w)` Returns the skewness of the entire wave *w*. See **WaveStats** for a discussion of skewness. The skew function was added in Igor Pro 8.00.

`sum(w)` Returns the sum of the entire wave *w*. The sum function was added in Igor Pro 8.00.

`variance(w)` Returns the variance of the entire wave *w*. See **WaveStats** for a discussion of variance. The variance function was added in Igor Pro 8.00.

Flags

/EX=*exDigits* Specifies the number of extra digits added to the precision digits (/N) for intermediate steps in the calculation.

/N=*numDigits* Specifies the precision of the final result. To add digits to the intermediate computation steps, use /EX.

/V Verbose mode; prints the result in the history in addition to performing the assignment.

/Z No error reporting.

Details

By default, all arbitrary precision math calculations are performed with *numDigits*=50 and *exDigits*=6, which yields a final result using at least 56 decimal places. Because none of the built-in variable types can express numbers with such high accuracy, the arbitrary precision numbers must be stored as strings. The operation automatically converts between strings and constants. It evaluates all of the numerical functions listed