

Guide to Custom Metric Expressions



In addition to this guide, see the list of [Custom Expression Examples](#) that demonstrates some of the possibilities.

Operators Supported in Custom Metric Expressions:

Boolean Not	!
Unary Plus, Unary Minus	+x, -x
Modulus	%
Division	/
Multiplication	*
Addition, Subtraction	+, -
Less or Equal, More or Equal	<=, >=
Less Than, Greater Than	<, >
Not Equal, Equal	!=, ==
Boolean And	&&
Boolean Or	
Assignment	=

Variables Supported in Custom Metric Expressions:

Variable Assignment

Assignment allows the values of variables to be set by using the = operator in equations so it is possible to assign values with expressions like

```
x = 3
```

then use the assigned values in an expression such as

```
y = x^2
```

and y will have the value 9. Assignment is enabled by default. Variable names can not have "." in the name.

Functions Supported in Custom Metric Expressions:

Trigonometric Functions

All functions accept arguments of the Double and Complex type, except atan2 which only accepts Double arguments.

Description	Function Name
Sine	sin(x)
Cosine	cos(x)
Tangent	tan(x)
Arc Sine ²	asin(x)
Arc Cosine ²	acos(x)
Arc Tangent	atan(x)

Arc Tan with 2 parameters	atan2(y, x)
Secant	sec(x)
Cosecant	cosec(x)
Co-tangent	cot(x)
Hyperbolic Sine	sinh(x)
Hyperbolic Cosine	cosh(x)
Hyperbolic Tangent	tanh(x)
Inverse Hyperbolic Sine	asinh(x)
Inverse Hyperbolic Cosine ¹	acosh(x)
Inverse Hyperbolic Tangent ¹	atanh(x)

Log and Exponential Functions

All functions accept arguments of the Double and Complex types.

Description	Function Name
Natural Logarithm ¹	ln(x)
Logarithm base 10 ¹	log(x)
Logarithm base 2 ¹	lg(x)
Exponential (e^x)	exp(x)
Power ¹	pow(x)

Statistical Functions

All functions accept either a vector (e.g. min(1,2,3)) or a set of numbers (e.g.min(1,2,3)).

Description	Function Name
Average	avg(x1,x2,x3,...)
Minimum	min(x1,x2,x3,...)
Maximum	max(x1,x2,x3,...)
Vector Sum	vsum(x1,x2,x3,...)

Rounding Functions

Description	Function Name
Round	round(x), round(x, p)
Round to integer	rint(x), rint(x, p)
Floor	floor(x)
Ceiling	ceil(x)

Miscellaneous Functions

Description	Function Name
If	if(cond, trueval, falseval)
Str (convert number to string)	str(x)
Absolute Value / Magnitude	abs(x)

Random number (between 0 and 1)	rand()
Modulus	mod(x,y) = x % y
Square Root ¹	sqrt(x)
Sum	sum(x,y,...)
Binomial coefficients	binom(n, i)
Signum (-1,0,1 depending on sign of argument)	signum(x)

Complex Functions

Description	Function Name
Real Component	re(c)
Imaginary Component	im(c)
Complex Modulus (Absolute Value)	cmod(c)
Argument (Angle of complex value, in radians)	arg(c)
Complex conjugate	conj(c)
Complex, constructs a complex number from real and imaginary parts	complex(x, y)
Polar, constructs a complex number from modulus and argument	polar(r, theta)

Notes

¹ By default functions like sqrt(-1) will return a complex result. These functions have constructors with a flag to control their behaviour for out of range values, if the flag is true, then the function will return Double.NaN for out of range real values. If the flag is false they will return the appropriate complex result.

² By default functions like acos(2) will return Double.NaN. A constructor is available which sets a flag to make the function return the appropriate complex result for such cases. Trigonometric Functions
All functions accept arguments of the Double and Complex type, except atan2 which only accepts Double arguments.

Other Functions

Class	Description
LogTwoArg	Two argument log function where second argument is the base.
Remainder	Calculates the remainder and quotient the arguments. Constructors allow different conventions for remainder to be used.
RoundSF	Rounds arguments to a specific number of significant figures.
FromBase	Converts a string in a given base to numbers.
Switch	A switch statement. Returns the value of the argument based on the value of the first argument.
SwitchDefault	A switch statement with a default value as final argument.
Case	A case statement, first argument is test condition, following arguments are in pairs with a test value and corresponding result.
IsNull	Tests if the argument is null.
IsNaN	Tests if the argument is NaN.
IsInfinite	Tests if the argument is infinite.
IsType	Test is the argument is of the type specifies in the constructor. E.g. isDouble