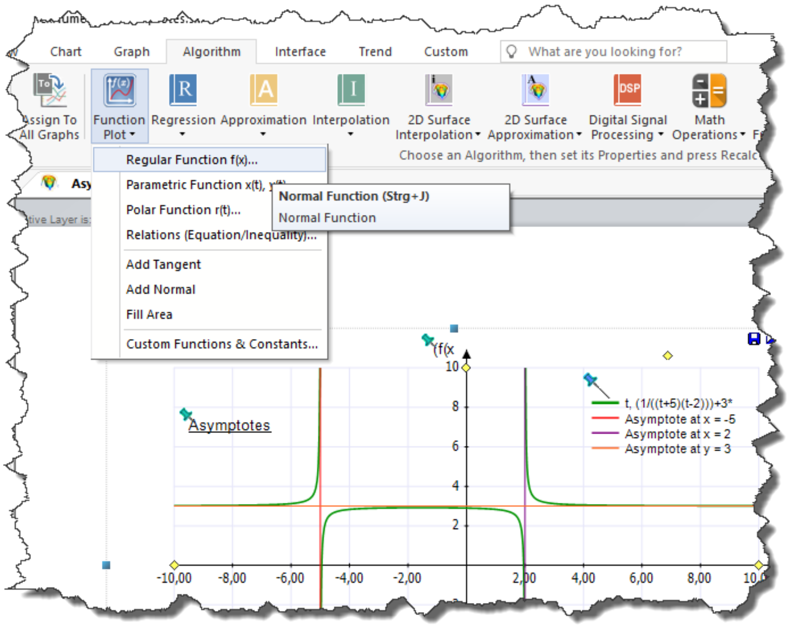
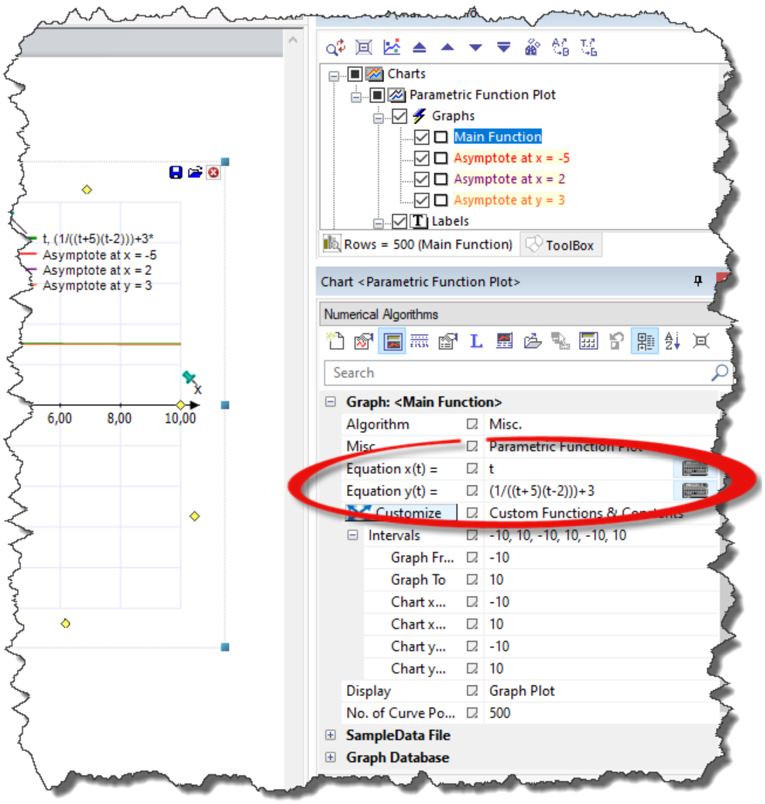
Info zu *SimplexNumerica* V17

Liste der Formel-Parser-Funktionen

Explizit eingegebene Formeln werden an verschiedenen Stellen im Programm benötigt. Hier das bekannteste Beispiel: Der Funktionsplot. Erreichbar in der Algorithmus-Ribbonbar, Icon Funktionsplot.

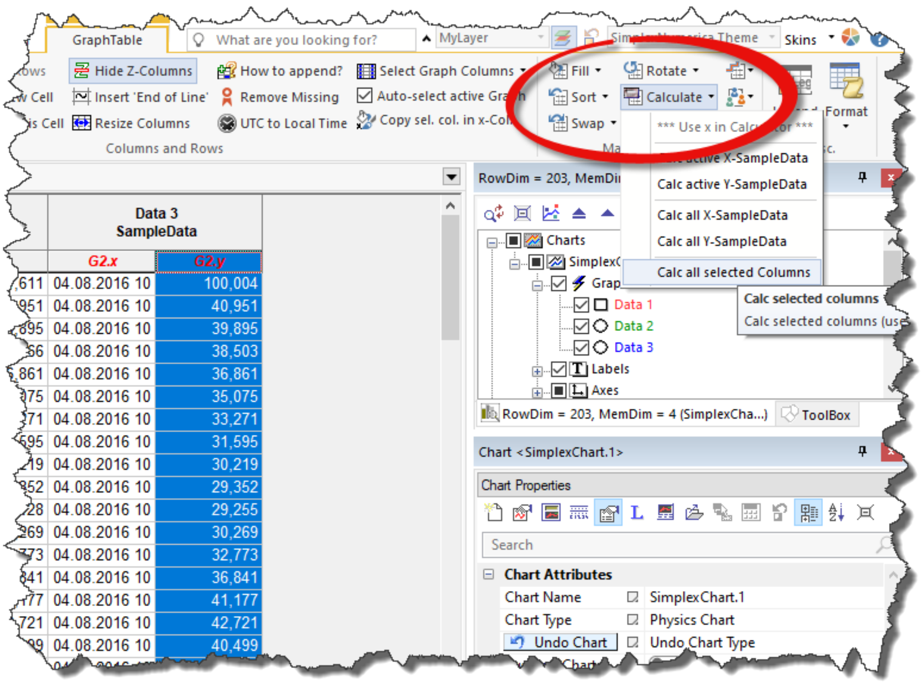


Die Formel kann über die Eigenschaften (Properties) direkt oder mit Hilfe von Simplexety eingegeben werden.

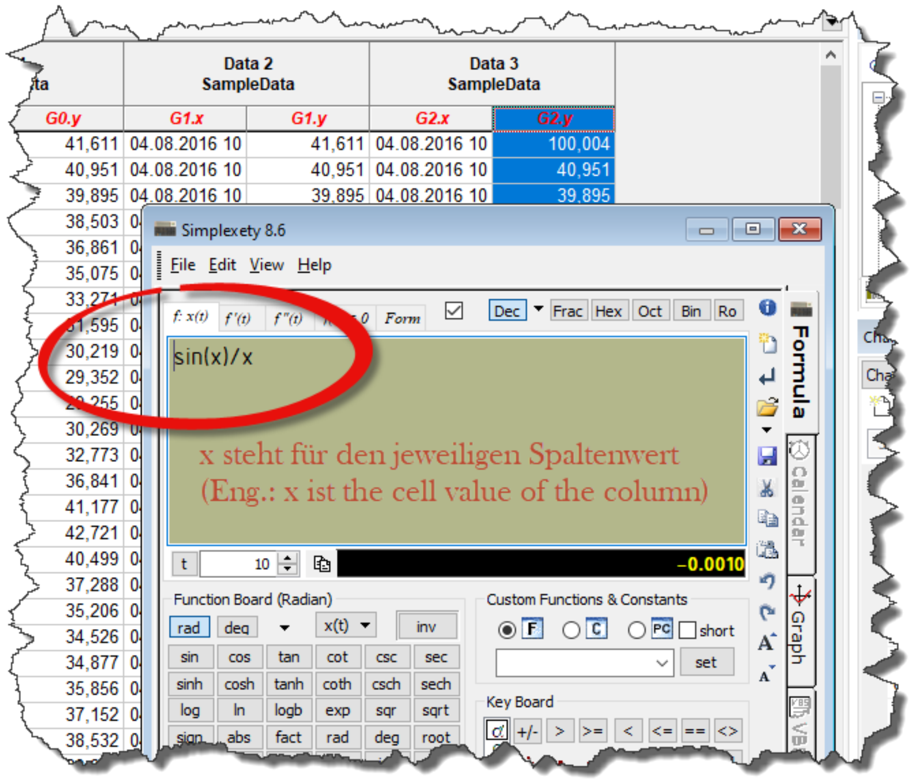


Eine andere Stelle an der man eine Formel benötigt ist die GraphTable.

Liste der Formel-Parser-Funktionen



Hier will man z.B. die Daten von gewissen Spalten mit einer Formel korrigieren.



Drücken Sie den Button

zum Verlassen des Dialogs.

Liste der Formel-Parser-Funktionen

⇨ *Die Funktionen des Formel-Parsers stammen vom bekannten Programm Graph von Ivan Johansen übernommen. Der Autor von SimplexNumerica, Ralf Wirtz, erhielt eine Sonderlizenz von Herrn Johansen.*

Im Folgenden finden Sie eine Liste aller vom Programm unterstützten Variablen, Konstanten, Operatoren und Funktionen. Die Liste der Operatoren zeigt zuerst die Operatoren mit der höchsten Priorität. Die Rangfolge der Operatoren kann durch die Verwendung von Klammern geändert werden. (), {} und [] können alle gleich verwendet werden. Beachten Sie, dass Ausdrücke in *SimplexNumerica* die Groß- und Kleinschreibung nicht berücksichtigen, d.h. es gibt keinen Unterschied zwischen Groß- und Kleinschreibung. Die einzige Ausnahme ist e als Eulerkonstante und E als Exponent in einer Zahl in wissenschaftlicher Notation.

Die folgende Tabelle zeigt alle eingebauten Konstanten:

Konstante	**Beschreibung**
x	Die unabhängige Variable, die in Standardfunktionen verwendet wird.
t	Die unabhängige Variable heißt Parameter für parametrische Funktionen und Polarwinkel für Polar Funktionen.
e	Euler ist konstant. In diesem Programm ist definiert als e = 2.718281828459045235360287
Pi	Die Konstante p, die in diesem Programm als pi = 3.141592653589793238462643 definiert ist
undef	Gibt immer einen Fehler zurück. Wird verwendet, um anzuzeigen, dass ein Teil einer Funktion nicht definiert ist.
ich	Die imaginäre Einheit. Definiert als i2 = -1. Nur nützlich, wenn mit komplexen Zahlen gearbeitet wird.
inf	Die Konstante für die Unendlichkeit. Nur als Argument für die Integrationsfunktion nützlich.
rand	Wertet eine Zufallszahl zwischen 0 und 1 aus.

Die nächste Tabelle zeigt alle eingebauten Operatoren :

Operator	
Potenzierung (^)	Steigern Sie die Macht eines Exponenten. Beispiel: f (x) = 2 ^ x
Verneinung (-)	Der negative Wert eines Faktors. Beispiel: f (x) = - x
Logisch NICHT (nicht)	Nicht a wird zu 1 ausgewertet, wenn a Null ist, andernfalls zu 0.

Liste der Formel-Parser-Funktionen

Multiplikation (*)	Multipliziert zwei Faktoren. Beispiel: f (x) = 2 * x
Einteilung (/)	Trennt zwei Faktoren. Beispiel: f (x) = 2 / x
Zusatz (+)	Fügt zwei Begriffe hinzu. Beispiel: f (x) = 2 + x
Subtraktion (-)	Subtrahiert zwei Terme. Beispiel: f (x) = 2-x
Größer als (>)	Gibt an, ob ein Ausdruck größer als ein anderer Ausdruck ist.
Größer oder gleich (> =)	Gibt an, ob ein Ausdruck größer oder gleich einem anderen Ausdruck ist.
Weniger als (<)	Gibt an, ob ein Ausdruck kleiner als ein anderer Ausdruck ist.
Kleiner oder gleich (<=)	Gibt an, ob ein Ausdruck kleiner oder gleich einem anderen Ausdruck ist.
Gleich (=)	Gibt an, ob zwei Ausdrücke exakt denselben Wert haben.
Nicht gleich (<>)	Gibt an, ob zwei Ausdrücke nicht exakt denselben Wert auswerten.
Logisches UND (und)	a und b werden zu 1 ausgewertet, wenn sowohl a als auch b ungleich Null sind, und ansonsten zu 0.
Logisches ODER (oder)	a oder b ergibt den Wert 1, wenn entweder a oder b ungleich Null ist, und ansonsten den Wert 0.
Logisches XOR (xor)	a xor b wird zu 1 ausgewertet, wenn entweder a oder b, jedoch nicht beide, ungleich Null sind, und ansonsten zu 0.

Die nächste Tabelle zeigt alle eingebauten Funktionen :

Funktion	**Beschreibung**
	Trigometrisch
Sünde	Gibt den Sinus des Arguments zurück, der im Bogenmaß oder Grad sein kann.
cos	Gibt den Cosinus des Arguments zurück, der im Bogenmaß oder Grad sein kann.
bräunen	Gibt den Tangens des Arguments zurück, der im Bogenmaß oder Grad sein kann.

Liste der Formel-Parser-Funktionen

wie in	Gibt den inversen Sinus des Arguments zurück. Der zurückgegebene Wert kann im Bogenmaß oder Grad sein.
acos	Gibt den inversen Cosinus des Arguments zurück. Der zurückgegebene Wert kann im Bogenmaß oder Grad sein.
eine Lohe	Gibt den inversen Tangens des Arguments zurück. Der zurückgegebene Wert kann im Bogenmaß oder Grad sein.
sek	Gibt die Sekante des Arguments zurück, die im Bogenmaß oder Grad sein kann.
csc	Gibt den Cosecans des Arguments zurück, der im Bogenmaß oder in Grad sein kann.
Kinderbett	Gibt den Kotangens des Arguments zurück, der in Radiant oder Grad sein kann.
eine Sekunde	Gibt die inverse Sekante des Arguments zurück. Der zurückgegebene Wert kann im Bogenmaß oder Grad sein.
acsc	Gibt den inversen Cosecant des Arguments zurück. Der zurückgegebene Wert kann im Bogenmaß oder Grad sein.
ein Kinderbett	Gibt den inversen Kotangens des Arguments zurück. Der zurückgegebene Wert kann im Bogenmaß oder Grad sein.
	Hyperbolisch
sinh	Gibt den hyperbolischen Sinus des Arguments zurück.
cosh	Gibt den hyperbolischen Cosinus des Arguments zurück.
Tanh	Gibt den hyperbolischen Tangens des Arguments zurück.
asinh	Gibt den inversen hyperbolischen Sinus des Arguments zurück.
acosh	Gibt den inversen hyperbolischen Kosinus des Arguments zurück.
atanh	Gibt den inversen hyperbolischen Tangens des Arguments zurück.
csch	Gibt den hyperbolischen Cosecant des Arguments zurück.
sech	Gibt die hyperbolische Sekante des Arguments zurück.
Coth	Gibt den hyperbolischen Kotangens des Arguments zurück.
acsch	Gibt die inverse hyperbolische Cosecant des Arguments zurück.
asech	Gibt die inverse hyperbolische Sekante des Arguments zurück.
acoth	Gibt den inversen hyperbolischen Kotangens des Arguments zurück.
	Leistung und Logarithmus
sqr	Gibt das Quadrat des Arguments zurück, dh die Zweierpotenz.
exp	Gibt den Wert e zurück, der dem Argument angehoben wurde.

Liste der Formel-Parser-Funktionen

sqrt	Gibt die Quadratwurzel des Arguments zurück.
Wurzel	Gibt die n-te Wurzel des Arguments zurück.
l	Gibt den Logarithmus mit Basis e an das Argument zurück.
Log	Gibt den Logarithmus mit der Basis 10 an das Argument zurück.
logb	Gibt den Logarithmus mit der Basis n an das Argument zurück.
	Komplex
Abs	Gibt den absoluten Wert des Arguments zurück.
arg	Gibt den Winkel des Arguments in Bogenmaß oder Grad zurück.
Konj	Gibt das Konjugat des Arguments zurück.
Re	Gibt den Realteil des Arguments zurück.
Ich bin	Gibt den Imaginärteil des Arguments zurück.
	Runden
abschneiden	Gibt den ganzzahligen Teil des Arguments zurück.
fraktur	Gibt den gebrochenen Teil des Arguments zurück.
ceiling	Rundet das Argument auf die nächste ganze Zahl auf.
Fußboden	Rundet das Argument auf die nächste Ganzzahl ab.
runden	Rundet das erste Argument auf die Anzahl der Dezimalstellen des zweiten Arguments.
	Stückweise
Zeichen	Gibt das Vorzeichen des Arguments zurück: 1, wenn das Argument größer als 0 ist, und -1, wenn das Argument kleiner als 0 ist.
u	Einheitenschritt: Gibt 1 zurück, wenn das Argument größer oder gleich 0 ist, andernfalls 0.
Mindest	Gibt das kleinste der Argumente zurück.
max	Gibt das größte der Argumente zurück.
Angebot	Gibt das zweite Argument zurück, wenn es im Bereich des ersten und des dritten Arguments liegt.
ob	Gibt das zweite Argument zurück, wenn das erste Argument nicht 0 ergibt. Andernfalls wird das dritte Argument zurückgegeben.
	Besondere
integrieren	Gibt das numerische Integral des ersten Arguments vom zweiten Argument zum dritten Argument zurück.

Liste der Formel-Parser-Funktionen

Summe	Gibt die Summe des ersten Arguments zurück, das für jede ganze Zahl im Bereich vom zweiten bis zum dritten Argument ausgewertet wird.
Produkt	Gibt das Produkt des ersten Arguments zurück, das für jede ganze Zahl im Bereich vom zweiten bis zum dritten Argument ausgewertet wird.
Tatsache	Gibt die Fakultät des Arguments zurück.
Gamma	Gibt die Euler-Gamma-Funktion des Arguments zurück.
Beta	Gibt die für die Argumente ausgewertete Beta-Funktion zurück.
W	Gibt die Lambert-W-Funktion zurück, die für das Argument ausgewertet wurde.
Zeta	Gibt die Riemann-Zeta-Funktion zurück, die für das Argument ausgewertet wurde.
mod	Gibt den Rest des ersten Arguments geteilt durch das zweite Argument zurück.
dnorm	Gibt die Normalverteilung des ersten Arguments mit optionalem Mittelwert und Standardabweichung zurück.

Beachten Sie die folgenden Beziehungen:

sin (x) ^ 2 = (sin (x)) ^ 2
sin 2x = sin (2x)
sin 2 + x = sin (2) + x
sin x ^ 2 = sin (x ^ 2)
2 (x + 3) x = 2 * (x + 3) * x
-x ^ 2 = - (x ^ 2)
2x = 2 * x
e ^ 2x = e ^ (2 * x)
x ^ 2 ^ 3 = x ^ (2 ^ 3)

Constants

rand constant

Returns a random number in the range 0 to 1.

Syntax
rand

Description
rand is used as a constant but returns a new pseudo-random number each time it is evaluated. The value is a

real number in the range [0;1].

Remarks
Because rand returns a new value each time it is evaluated, a graph using rand will not look the same each time it is drawn. A graph using rand will also change when the program is forced to redraw, e.g. because the
coordinate system is moved, resized or zoomed.

Implementation
rand uses a multiplicative congruential random number generator with period 2 to the 32nd power to return
successive pseudo-random numbers in the range from 0 to 1.

See also
Wikipedia [http://en.wikipedia.org/wiki/Random_number_generator#Computational_methods]

Trigonometric

sin function

Returns the sine of the argument.

Syntax
sin(z)

Description
The sin function calculates the sine of an angle z, which may be in radians or degrees depending on the current settings. z may be any numeric expression that evaluates to a real number or a complex number. If z is a real number, the result will be in the range -1 to 1.

Remarks
For arguments with a large magnitude, the function will begin to lose precision.

See also
Wikipedia [http://en.wikipedia.org/wiki/Trigonometric_functions#Sine]

---

cos function

Returns the cosine of the argument.

Syntax
cos(z)

Description

The cos function calculates the cosine of an angle z, which may be in radians or degrees depending on the current settings. z may be any numeric expression that evaluates to a real number or a complex number. If z is a real number, the result will be in the range -1 to 1.

Remarks
For arguments with a large magnitude, the function will begin to lose precision.

See also
Wikipedia [http://en.wikipedia.org/wiki/Trigonometric_functions#Cosine]

---

tan function

Returns the tangent of the argument.

Syntax
tan(z)

Description
The tan function calculates the tangent of an angle z, which may be in radians or degrees depending on the current settings. z may be any numeric expression that evaluates to a real number or a complex number.

Remarks
For arguments with a large magnitude, the function will begin to lose precision. tan is undefined at z = p*p/2, where p is an integer, but the function returns a very large number if z is near the undefined value.

See also
Wikipedia [http://en.wikipedia.org/wiki/Trigonometric_functions#Tangent]

---

asin function

Returns the inverse sine of the argument.

Syntax
asin(z)

Description
The asin function calculates the inverse sine of z. The result may be in radians or degrees depending on the current settings. z may be any numeric expression that evaluates to a real number. This is the reverse of the sin function.

See also
Wikipedia [http://en.wikipedia.org/wiki/Inverse_trigonometric_functions]

Liste der Formel-Parser-Funktionen

acos function

Returns the inverse cosine of the argument.

Syntax
acos(z)

Description
The acos function calculates the inverse cosine of z. The result may be in radians or degrees depending on the current settings. z may be any numeric expression that evaluates to a real number. This is the reverse of the cos function.

See also
Wikipedia [http://en.wikipedia.org/wiki/Inverse_trigonometric_functions]

---

atan function

Returns the inverse tangent of the argument.

Syntax
atan(z)

Description
The atan function calculates the inverse tangent of z. The result may be in radians or degrees depending on the current settings. z may be any numeric expression that evaluates to a real number. This is the reverse of the tan function.

See also
Wikipedia [http://en.wikipedia.org/wiki/Inverse_trigonometric_functions]

---

sec function

Returns the secant of the argument.

Syntax
sec(z)

Description
The sec function calculates the secant of an angle z, which may be in radians or degrees depending on the current settings. sec(z) is the same as 1/cos(z). z may be any numeric expression that evaluates to a real number or a complex number.

Remarks
For arguments with a large magnitude, the function will begin to lose precision.

See also
Wikipedia [http://en.wikipedia.org/wiki/Trigonometric_functions#Reciprocal_functions]

---

csc function

Returns the cosecant of the argument.

Syntax
csc(z)

Description
The csc function calculates the cosecant of an angle z, which may be in radians or degrees depending on the current settings. csc(z) is the same as 1/sin(z). z may be any numeric expression that evaluates to a real number or a complex number.

Remarks
For arguments with a large magnitude, the function will begin to lose precision.

See also
Wikipedia [http://en.wikipedia.org/wiki/Trigonometric_functions#Reciprocal_functions]

---

ot function

Returns the cotangent of the argument.

Syntax
cot(z)

Description
The cot function calculates the cotangent of an angle z, which may be in radians or degrees depending on the current settings. cot(z) is the same as 1/tan(z). z may be any numeric expression that evaluates to a real number or a complex number.

Remarks
For arguments with a large magnitude, the function will begin to lose precision.

See also
Wikipedia [http://en.wikipedia.org/wiki/Trigonometric_functions#Reciprocal_functions]

---

asec function

Returns the inverse secant of the argument.

Syntax
asec(z)

Description
The asec function calculates the inverse secant of z. The result may be in radians or degrees depending on the current settings. asec(z) is the same as acos(1/z). z may be any numeric expression that evaluates to a real
number. This is the reverse of the sec function.

See also
Wikipedia [http://en.wikipedia.org/wiki/Inverse_trigonometric_functions]

---

acsc function

Returns the inverse cosecant of the argument.

Syntax
acsc(z)

Description
The acsc function calculates the inverse cosecant of z. The result may be in radians or degrees depending on the current settings. acsc(z) is the same as asin(1/z). z may be any numeric expression that evaluates to a real number. This is the reverse of the csc function.

See also
Wikipedia [http://en.wikipedia.org/wiki/Inverse_trigonometric_functions]

---

acot function

Returns the inverse cotangent of the argument.

Syntax
acot(z)

Description
The acot function calculates the inverse cotangent of z. The result may be in radians or degrees depending on the current settings. acot(z) is the same as atan(1/z). z may be any numeric expression that evaluates to a real number. This is the reverse of the cot function.

Remarks
The acot function returns a value in the range]-p/2;p/2] (]-90;90] when calculating in degrees), which is the most common definition, though some may define it to be in the range]0;p[.

See also
Wikipedia [http://en.wikipedia.org/wiki/Inverse_trigonometric_functions]

Liste der Formel-Parser-Funktionen

Hyperbolic

sinh function
Returns the hyperbolic sine of the argument.

Syntax
sinh(z)

Description
The sinh function calculates the hyperbolic sine of z. z may be any numeric expression that evaluates to a real number or a complex number.
Hyperbolic sine is defined as: sinh(z) = ½(ez-e-z)

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

---

cosh function
Returns the hyperbolic cosine of the argument.

Syntax
cosh(z)

Description
The cosh function calculates the hyperbolic cosine of z. z may be any numeric expression that evaluates to a real number or a complex number.
Hyperbolic cosine is defined as: cosh(z) = ½(ez+e-z)

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

---

tanh function
Returns the hyperbolic tangent of the argument.

Syntax
tanh(z)

Description
The tanh function calculates the hyperbolic tangent of z. z may be any numeric expression that evaluates to a real number or a complex number.
Hyperbolic tangent is defined as: tanh(z) = sinh(z)/cosh(z)

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

---

asinh function

Returns the inverse hyperbolic sine of the argument.

Syntax
asinh(z)

Description
The asinh function calculates the inverse hyperbolic sine of z. z may be any numeric expression that evaluates to a real number or a complex number. asinh is the reverse of sinh, i.e. asinh(sinh(z)) = z.

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

---

acosh function

Returns the inverse hyperbolic cosine of the argument.

Syntax
acosh(z)

Description
The acosh function calculates the inverse hyperbolic cosine of z. z may be any numeric expression that evaluates to a real number or a complex number. acosh is the reverse of cosh, i.e. acosh(cosh(z)) = z.

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

---

atanh function

Returns the inverse hyperbolic tangent of the argument.

Syntax
atanh(z)

Description
The atanh function calculates the inverse hyperbolic tangent of z. z may be any numeric expression that evaluates to a real number or a complex number. atanh is the reverse of tanh, i.e. atanh(tanh(z)) = z.

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

Liste der Formel-Parser-Funktionen

csch function

Returns the hyperbolic cosecant of the argument.

Syntax
csch(z)

Description
The csch function calculates the hyperbolic cosecant of z. z may be any numeric expression that evaluates to a real number or a complex number.
Hyperbolic cosecant is defined as: csch(z) = 1/sinh(z) = 2/(ez-e-z)

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

sech function

Returns the hyperbolic secant of the argument.

Syntax
sech(z)

Description
The sech function calculates the hyperbolic secant of z. z may be any numeric expression that evaluates to a real number or a complex number.
Hyperbolic secant is defined as: sech(z) = 1/cosh(z) = 2/(ez+e-z)

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

coth function

Returns the hyperbolic cotangent of the argument.

Syntax
coth(z)

Description
The coth function calculates the hyperbolic cotangent of z. z may be any numeric expression that evaluates to a real number or a complex number.
Hyperbolic cotangent is defined as: coth(z) = 1/tanh(z) = cosh(z)/sinh(z) = (ez + e-z)/(ez - e-z)

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

Liste der Formel-Parser-Funktionen

---

acsch function

Returns the inverse hyperbolic cosecant of the argument.

Syntax
acsch(z)

Description
The acsch function calculates the inverse hyperbolic cosecant of z. z may be any numeric expression that evaluates to a real number or a complex number. acsch is the reverse of csch, i.e. acsch(csch(z)) = z.

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

---

asech function

Returns the inverse hyperbolic secant of the argument.

Syntax
asech(z)

Description
The asech function calculates the inverse hyperbolic secant of z. z may be any numeric expression that evaluates to a real number or a complex number. asech is the reverse of sech, i.e. asech(sech(z)) = z.

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

---

acoth function

Returns the inverse hyperbolic cotangent of the argument.

Syntax
acoth(z)

Description
The acoth function calculates the inverse hyperbolic cotangent of z. z may be any numeric expression that evaluates to a real number or a complex number. acoth is the reverse of coth, i.e. acoth(coth(z)) = z. For real numbers acoth is undefined in the interval [-1;1].

See also
Wikipedia [http://en.wikipedia.org/wiki/Hyperbolic_function]

Liste der Formel-Parser-Funktionen

Power and logarithm

sqr function
Returns the square of the argument.

Syntax
sqr(z)

Description
The sqr function calculates the square of z, i.e. z raised to the power of 2. z may be any numeric expression that evaluates to a real number or a complex number.
exp function
Returns e raised to the power of the argument.

Syntax
exp(z)

Description
The exp function is used to raise e, Euler's constant, to the power of z. This is the same as e^z. z may be any numeric expression that evaluates to a real number or a complex number.

See also
Wikipedia [http://en.wikipedia.org/wiki/Exponential_function]

---

sqrt function
Returns the square root of the argument.

Syntax
sqrt(z)

Description
The sqrt function calculates the square root of z, i.e. z raised to the power of ½. z may be any numeric expression that evaluates to a real number or a complex number. If the calculation is done with real numbers,
the argument is only defined for z ³ 0.

See also
Wikipedia [http://en.wikipedia.org/wiki/Square_root]

---

root function
Returns the nth root of the argument.

Syntax
root(n, z)

Description
The root function calculates the nth root of z. n and z may be any numeric expression that evaluates to a real number or a complex number. If the calculation is done with real numbers, the argument is only defined
for z ^{3} 0.

Remarks
When the calculation is done with real numbers, the function is only defined for z<0 if n is an odd integer. For calculations with complex numbers, root is defined for the whole complex plane except at the pole n=0. Notice that for calculations with complex numbers the result will always have an imaginary part when z<0 even though the result is real when calculations are done with real numbers and n is an odd integer.
Example
Instead of x^(1/3), you can use root(3, x).

See also
Wikipedia [http://en.wikipedia.org/wiki/Nth_root]

---

ln function

Returns the natural logarithm of the argument.

Syntax
ln(z)

Description
The ln function calculates the logarithm of z with base e, which is Euler's constant. ln(z) is commonly known as the natural logarithm. z may be any numeric expression that evaluates to a real number or a complex number. If the calculation is done with real numbers, the argument is only defined for z>0. When calculating with complex numbers, z is defined for all numbers except z=0.

See also
Wikipedia [http://en.wikipedia.org/wiki/Natural_logarithm]

---

log function

Returns the base 10 logarithm of the argument.

Syntax
log(z)

Description
The log function calculates the logarithm of z with base 10. z may be any numeric expression that evaluates

to a real number or a complex number. If the calculation is done with real numbers, the argument is only defined for z>0. When calculating with complex numbers, z is defined for all numbers except z=0.

See also
Wikipedia [http://en.wikipedia.org/wiki/Common_logarithm]

---

logb function

Returns the base n logarithm of the argument.

Syntax
logb(z, n)

Description
The logb function calculates the logarithm of z with base n. z may be any numeric expression that evaluates to a real number or a complex number. If the calculation is done with real numbers, the argument is only defined for z>0. When calculating with complex numbers, z is defined for all numbers except z=0. n must evaluate to a positive real number.

See also
Wikipedia [http://en.wikipedia.org/wiki/Logarithm]

Complex

abs function

Returns the absolute value of the argument.

Syntax
abs(z)

Description
The abs function returns the absolute or numeric value of z, commonly written as |z|. z may be any numeric expression that evaluates to a real number or a complex number. abs(z) always returns a positive real value.

See also
Wikipedia [http://en.wikipedia.org/wiki/Absolute_value]

---

arg function

Returns the argument of the parameter.

Syntax
arg(z)

Description
The arg function returns the argument or angle of z. z may be any numeric expression that evaluates to a real number or a complex number. arg(z) always returns a real number. The result may be in radians or degrees depending on the current settings. The angle is always between -p and p. If z is a real number, arg(z)
is 0 for positive numbers and p for negative numbers. arg(0) is undefined.

See also
Wikipedia [http://en.wikipedia.org/wiki/Arg_(mathematics)]

---

conj function

Returns the conjugate of the argument.

Syntax
conj(z)

Description
The conj function returns the conjugate of z. z may be any numeric expression that evaluates to a real number or a complex number. The function is defined as: conj(z) = re(z) - i*im(z).

See also
Wikipedia [http://en.wikipedia.org/wiki/Complex_conjugation]

---

re function

Returns the real part of the argument.

Syntax
re(z)

Description
The re function returns the real part of z. z may be any numeric expression that evaluates to a real number or a complex number.

See also
Wikipedia [http://en.wikipedia.org/wiki/Real_part]

---

im function

Returns the imaginary part of the argument.

Syntax
im(z)

Description
The im function returns the imaginary part of z. z may be any numeric expression that evaluates to a real number or a complex number.

See also
Wikipedia [http://en.wikipedia.org/wiki/Imaginary_part]

Rounding

trunc function
Removes the fractional part of the argument.

Syntax
trunc(z)

Description
The trunc function returns the integer part of z. The function removes the decimal part of z, i.e. rounds against zero. z may be any numeric expression that evaluates to a real number or a complex number. If z is a
complex number, the function returns trunc(re(z))+trunc(im(z))i.

See also
Wikipedia [http://en.wikipedia.org/wiki/Truncate]

---

fract function
Returns the fractional part of the argument.

Syntax
fract(z)

Description
The fract function returns the fractional part of z. The function removes the integer part of z, i.e. fract(z) = z - trunc(z). z may be any numeric expression that evaluates to a real number or a complex number. If z is a complex number, the function returns fract(re(z))+fract(im(z))i.

See also
Wikipedia [http://en.wikipedia.org/wiki/Floor_and_ceiling_functions#Fractional_part]

---

ceil function
Rounds the argument up.

Syntax
ceil(z)

Description
The ceil function finds the smallest integer not less than z. z may be any numeric expression that evaluates to a real number or a complex number. If z is a complex number, the function returns ceil(re(z))+ceil(im(z))i.

See also
Wikipedia [http://en.wikipedia.org/wiki/Floor_and_ceiling_functions]

---

floor function
Rounds the argument down.

Syntax
floor(z)

Description
The floor function, which is also called the greatest integer function, gives the largest integer not greater than z. z may be any numeric expression that evaluates to a real number or a complex number. If z is a complex number, the function returns floor(re(z))+floor(im(z))i.

See also
Wikipedia [http://en.wikipedia.org/wiki/Floor_and_ceiling_functions]

---

round function
Rounds a number to the specified number of decimals.

Syntax
round(z,n)

Description
The round function rounds z to the number of decimals given by n. z may be any numeric expression that evaluates to a real number or a complex number. If z is a complex number, the function returns round(re(z),n)+round(im(z),n)i. n may be any numeric expression that evaluates to an integer. If n<0, z is rounded to n places to the left of the decimal point.
Examples
round(412.4572,3) = 412.457
round(412.4572,2) = 412.46
round(412.4572,1) = 412.5
round(412.4572,0) = 412
round(412.4572,-2) = 400

See also
Wikipedia [http://en.wikipedia.org/wiki/Rounding]

Piecewise

sign function

Returns the sign of the argument.

Syntax
sign(z)

Description
The sign function, which is also called signum, returns the sign of z. z may be any numeric expression that evaluates to a real number or a complex number. When z is a real number, sign(z) returns 1 for z>0 and -1 for z<0. sign(z) returns 0 for z=0. When z evaluates to a complex number, sign(z) returns z/abs(z).

See also
Wikipedia [http://en.wikipedia.org/wiki/Sign_function]

---

u function

The unit step function.

Syntax
u(z)

Description
u(z) is commonly known as the unit step function. z may be any numeric expression that evaluates to a real number. The function is undefined when z has an imaginary part. u(z) returns 1 for $z^3$0 and 0 for z<0.

See also
Wikipedia [http://en.wikipedia.org/wiki/Unit_step#Discrete_form]

---

min function

Finds and returns the minimum of the values passed as arguments.

Syntax
min(A,B,...)

Description
The min function returns the minimum value of its arguments. min can take any number of arguments not less than 2. The arguments may be any numeric expressions that evaluate to real numbers or complex

numbers. If the arguments are complex numbers, the function returns min(re(A), re(B), ...) + min(im(A), im(B), ...)i.

---

max function

Finds and returns the maximum of the values passed as arguments.

Syntax
max(A,B,...)

Description
The max function returns the maximum value of its arguments. max can take any number of arguments not less than 2. The arguments may be any numeric expressions that evaluate to real numbers or complex numbers. If the arguments are complex numbers, the function returns max(re(A), re(B), ...) + max(im(A), im(B), ...)i.

---

range function

Returns the second argument if it is in the range between the first argument and the third argument.

Syntax
range(A,z,B)

Description
The range function returns z, if z is greater than A and less than B. If $z < A$ then A is returned. If $z > B$ then B is returned. The arguments may be any numeric expressions that evaluate to real numbers or complex
numbers. The function has the same effect as max(A, min(z, B)).

---

if function

Evaluates one or more conditions and returns a different result based on them.

Syntax
if(cond1, f1, cond2, f2, ... , condn, fn [,fz])

Description
The if function evaluates cond1 and if it is different from 0 then f1 is evaluated and returned. Else cond2 is evaluated and if it is different from 0 then f2 is returned and so forth. If none of the conditions are true fz is returned. fz is optional and if not specified if returns an error if none of the conditions are true. The arguments may be any numeric expressions that evaluate to real numbers or complex numbers.

Liste der Formel-Parser-Funktionen

Special

integrate function

Returns an approximation for the numerical integral of the given expression over the given range.

Syntax
integrate(f,var,a,b)

Description
The integrate function returns an approximation for the numerical integral of f with the variable var from a to b. This is mathematically written as:

$$\int_a^b f(x)\ dx$$

This integral is the same as the area between the function f and the x-axis from a to b where the area under the axis is counted negative. f may be any function with the variable indicated as the second argument var. a and b may be any numeric expressions that evaluate to real numbers or they can be -INF or INF to indicate negative or positive infinity. integrate does not calculate the integral exactly. Instead the calculation is done using the Gauss-Kronrod 21-point integration rule adaptively to an estimated relative error
less than 10-3.

Examples
f(x)=integrate(t^2-7t+1, t, -3, 15) will integrate f(t)=t^2-7t+1 from -3 to 15 and evaluate to 396. More useful
is f(x)=integrate(s*sin(s), s, 0, x). This will plot the integral of f(s)=s*sin(s) from 0 to x, which is the same as the definite integral of f(x)=x*sin(x).

See also
Wikipedia [http://en.wikipedia.org/wiki/Integral]

---

sum function

Returns the summation of an expression evaluated over a range of integers.

Syntax
sum(f,var,a,b)

Description
The sum function returns the summation of f where var is evaluated for all integers from a to b. This is mathematically written as:

$$\sum_{x=a}^{b} f(x)$$

f may be any function with the variable indicated as the second argument var. a and b may be any numeric expressions that evaluate to integers.

See also
Wikipedia [http://en.wikipedia.org/wiki/Summation]

---

product function

Returns the product of an expression evaluated over a range of integers.

Syntax
product(f,var,a,b)

Description
The product function returns the product of f where var is evaluated for all integers from a to b. This is mathematically written as:

$$\prod_{x=a}^{b} f(x)$$

f may be any function with the variable indicated as the second argument var. a and b may be any numeric expressions that evaluate to integers.

See also
Wikipedia [http://en.wikipedia.org/wiki/Multiplication#Capital_pi_notation]

---

fact function

Returns the factorial of the argument.

Syntax
fact(n)

Description
The fact function returns the factorial of n, commonly written as n!. n may be any numeric expression that evaluates to a positive integer. The function is defined as fact(n)=n(n-1)(n-2)...1, and relates to the gamma function as fact(n)=gamma(n+1).

See also
Wikipedia [http://en.wikipedia.org/wiki/Factorial]

Liste der Formel-Parser-Funktionen

gamma function

Returns the value of the Euler gamma function of the argument.

Syntax
gamma(z)

Description
The gamma function returns the result of the Euler gamma function of z, commonly written as G(z). z may be any numeric expression that evaluates to a real number or a complex number. The gamma function relates
to the factorial function as fact(n)=gamma(n+1). The mathematical definition of the gamma function is:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$$

This cannot be calculated precisely, so Graph is using the Lanczos approximation to calculate the gamma function.

See also
Wikipedia [http://en.wikipedia.org/wiki/Gamma_function]

---

beta function

Returns the value of the Euler beta function evaluated for the arguments.

Syntax
beta(m, n)

Description
The beta function returns the result of the Euler beta function evaluated for m and n. m and n may be any numeric expressions that evaluate to real numbers or complex numbers. The beta function relates to the gamma function as beta(m, n) = gamma(m) * gamma(n) / gamma(m+n).

See also
Wikipedia [http://en.wikipedia.org/wiki/Beta_function]

W function

Returns the value of the Lambert W-function evaluated for the argument.

Syntax
W(z)

Description
The W function returns the result of the Lambert W-function, also known as the omega function, evaluated

for
z. z may be any numeric expression that evaluates to a real number or a complex number. The inverse of the
W function is given by f(W)=W*eW.

Remarks
For real values of z when z < -1/e, the W function will evaluate to values with an imaginary part.

See also
Wikipedia [http://en.wikipedia.org/wiki/Lambert_w_function]

---

zeta function

Returns the value of the Riemann Zeta function evaluated for the argument.

Syntax
zeta(z)

Description
The zeta function returns the result of the Riemann Zeta function, commonly written as z(s). z may be any numeric expression that evaluates to a real number or a complex number.

Remarks
The zeta function is defined for the whole complex plane except for the pole at z=1.

See also
Wikipedia [http://en.wikipedia.org/wiki/Riemann_zeta_function]

---

mod function

Returns the remainder of the first argument divided by the second argument.

Syntax
mod(m,n)

Description
Calculates m modulo n, the remainder of m/n. mod calculates the remainder f, where m = a*n + f for some integer a. The sign of f is always the same as the sign of n. When n=0, mod returns 0. m and n may be any numeric expressions that evaluate to real numbers.

See also
Wikipedia [http://en.wikipedia.org/wiki/Modular_arithmetic]

Liste der Formel-Parser-Funktionen

dnorm function

Returns the normal distribution of the first argument with optional mean value and standard deviation.

Syntax
dnorm(x, [μ,s])

Description
The dnorm function is the probability density of the normal distribution, also called Gaussian distribution. x is the variate, also known as the random variable, μ is the mean value and s is the standard deviation. μ and s are optional and if left out the standard normal distribution is used where μ=0 and s=1. x, μ and s may be any numeric expressions that evaluate to real numbers where s > 0. The normal distribution is defined as:

$$\mathrm{dnorm}(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

See also
Wikipedia [http://en.wikipedia.org/wiki/Normal_distribution]