SigmaCalculator

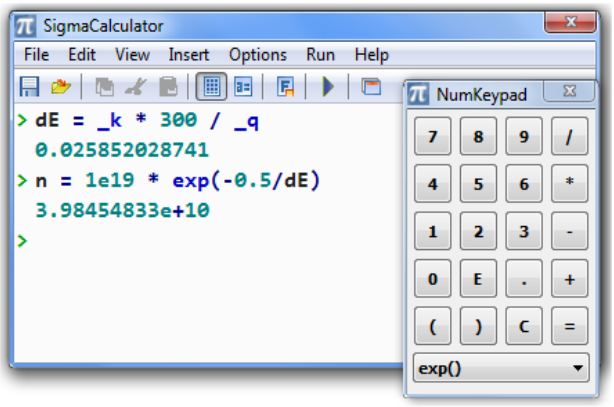
Mathematical Calculator

Sidi HAMADY
sidi@hamady.org

Mathematical Calculator

Contents

Mathematical Calculator



SigmaCalculator is an advanced mathematical expression-based calculator. It supports the most common and useful functions. It's easy to use: to evaluate an expression, simply write it, using operators (+ - * / ^), parenthesis and mathematical functions and press ENTER (or F12). You can also use the numeric keypad to enter numbers and operators. You can set variables (with any non-reserved name), using fundamental constants, etc. The **SigmaCalculator** menu and toolbar give you an easy way to use the software functionality.

To know if a new version is available, click *Menu/Help/Check for Update...* or visit my website: http://www.hamady.org

The following mathematical **functions** are supported:

exp(x)	// exponential
ln(x)	// natural logarithm
log(x)	// decimal logarithm
log2(x)	// base-2 logarithm
pow(x,n)	// x^n
sin(x)	// sine
cos(x)	// cosine
tan(x)	// tangent
asin(x)	// arc sine
acos(x)	// arc cosine
atan(x)	// arc tangent
sinh(x)	// hyperbolic sine
cosh(x)	// hyperbolic cosine
tanh(x)	// hyperbolic tangent
abs(x)	// absolute value
sqrt(x)	// square root
ceil(x)	// ceiling, the smallest integer not less than x
floor(x)	// floor, the largest integer not greater than x
fmod(x,y)	// x modulo y
erf(x)	// error function
j0(x)	// Bessel function of x of the first kind of order 0
j1(x)	// Bessel function of x of the first kind of order 1

jn(n,x)	// Bessel function of x of the first kind of order n
y0(x)	// Bessel function of x of the second kind of order 0
y1(x)	// Bessel function of x of the second kind of order 1
yn(n,x)	// Bessel function of x of the second kind of order n
bern(x)	// Bernoulli function: x / (exp(x) – 1)
gauss(x,μ,σ)	// Gauss function: $\exp((x - \mu)^2 / 2\sigma^2)$
lorentz(x,μ,σ)	// Lorentz function: $\sigma / ((x - \mu)^2 + \sigma^2)$
hypot(x,y)	// hypotenuse, $\sqrt{x^2 + y^2}$
min(x,y)	// smallest value of x and y
max(x,y)	// largest value of x and y
rand(x)	// random number between 0 and 1 (if x ≠ 0 then initialize the generator)
time()	// elapsed time in seconds since January 1, 1970
sign(x)	// sign of x (-1 if x < 0, +1 if x > 0 and 0 if x = 0)
hypot(x,y)	// $\sqrt{x^2+y^2}$
erf(x)	// error function
gammaln(x)	// ln(gamma(x))
beta(x)	// beta(x)
trunc(x)	// nearest integer
round(x)	// nearest integer, rounding
rint(x)	// rounds the floating-point to an integer

Constants:

Pi	// π
_q	// electron charge
_m	// electron mass
_k	// Boltzmann constant
_h	// Planck constant
_c	// speed of light in vacuum
_e	// vacuum permittivity
_n	// Avogadro constant

Commands:

cls	// clear the console
save	// save the current session into a text file
set?	// list the defined variables
format single	// set the numerical format to single float
format double	// set the numerical format to double float
format int	// set the numerical format to integer
format?	// print the current numerical format
help	// show this help
exit	// exit the application

Variables:

You can create variables (with any non-reserved name):

```
a = 6*2
 12
```

The last expression evaluation can be accessed using the internal variable `ans` :

```
6*2
 12
ans
 12
```

Appending a semicolon (`;`) to the expression suppress the output:

```
a = 6*2;
a = 6*2
 12
```

Semicolon (`;`) can also be used to separate multiple commands in a single line:

```
a=12;b=2;c=sqrt(a*b);
```

A comment can be added at the end of an expression, using `#` :

```
y=sin(pi/4)   # comment
 0.707106781187
```

Previous calculated expressions can be reused by pressing `up` or `down` arrows.

Command line:

SigmaCalculator can be executed from the command line:

```
$ sicalc  -run  input  [-out  outfile]  [-show]
```

input may be a filename or a double-quoted expression and **outfile** is the output filename:

```
$ sicalc  -run  "a=1;b=2;c=a*sin(b)"
$ sicalc  -run  calcin.txt  -out  calcout.txt
```

Integer arithmetic in binary, octal and hexadecimal bases:

To perform conversion between binary, octal, decimal and hexadecimal bases in 32 bits unsigned format, you can use the included converter: menu View/Base Converter. You can also perform integer calculations in any of these four bases directly within the console, by prefixing the number with **0b** for binary base, **0o** for octal base and **0x** for hexadecimal base: just type the expression and press enter:

```
0xFFF7 + 12 + 0b111 + 0o547
 65905
```

In menu Options, you can set the base used to print out the calculation result.

Bitwise operations can be performed using `~` (NOT) `&` (AND) `|` (OR) operators:

```
0x7120 & 0x0F10
 0x100
~(0x7120 & 0x0F10)
 0xFFFFFEFF
```

Root of nonlinear function:

To solve a nonlinear equation `f(x) = 0`, you can use one of the following methods:

Method #1: Type `solve(func, xl, xh)` where `func` is the `f(x)` function expression, `xl` the x lower limit of the interval where the solution is to be found and `xh` the x higher limit (`xl` and `xh` are optional). Example: type the following command and press enter:

```
solve(x^2 - 2, 0, 10)
 x = 1.414213216172   y = -9.79204372e-7
```

The solution will be printed out alongside with the corresponding function value y.

Method #2: Firstly, type the interval where the solution is to be found:

```
a = 0;
b = 10;
```

Then type the equation to be solved and press enter:

```
x^2 - 2 -> 0
 x = 1.414213216172   y = -9.79204372e-7
```

Method #3: Use the solver dialog to find the solution: menu *View/Solver*. Just enter the above mentioned parameters and click *'Solve'*.

Function plotting:

To plot a function, just type `plot(func, xl, xh, points)` where `func` is the function expression, `xl` the x lower limit, `xh` the x higher limit and `points` the number of points on the curve (`xl` , `xh` and `points` are optional). Example: type the following command and press enter:

```
plot(1 - exp(-x), 0, 5, 100)
```

The plot window will then be displayed. All the plot properties (curves options, scale, axis, colors …) can be easily modified. You can also add text, lines, rectangles, ellipses … to the plot and save it as PNG or SVG directly from within the plot window.

Specifications

SYSTEM REQUIREMENTS

SigmaCalculator is a **Comet** component and is not available as a standalone application.

Comet runs on:

(i) PC with Windows™ XP, Vista, Windows 7/8/10 installed.

(ii) PC with Linux 32bit or 64bit installed.

(iii) Smartphone or Tablet with Android 2.2 or later installed.

INSTALL

- For **portable version** on **Windows** and **Linux**: Download ***comet_windows_64bit.zip*** (Windows 64bit) or ***comet_windows_32bit.zip*** (Windows 32bit) or ***comet_linux_64bit.tgz*** (Linux 64bit) or ***comet_linux_32bit.tgz*** (Linux 32bit) from http://www.hamady.org, unzip/untar in any location (USB key or memory stick for example) and run *comet.exe* (Windows) or *comet* (Linux) in the bin directory.
- For **setup version** (on **Windows 64bit**): Download ***comet_windows_64bit.msi*** from http://www.hamady.org, double-click on the setup file and it will install Comet.
- For **Android**: Install from Google Play.

Links

To view the Comet online Help and read the latest news, visit my website: **http://www.hamady.org/**

For the language documentation, you can visit the Lua official website:

http://www.lua.org/ and wiki: http://lua-users.org/wiki/

Support

For assistance request, bug report, suggestion, question, you can either:

(i) send an e-mail to sidi@hamady.org.

(ii) visit my website (http://www.hamady.org) and submit a support request.

You will have an answer as soon as possible.

Copyright