A 10 minute tutorial for solving Math problems with Maxima

By Antonio Cangiano. Filed under Essential Math, Software
About 50,000 people read my article 3 awesome free Math programs. Chances are that at least some of them downloaded and installed Maxima. If you are one of them but are not acquainted with CAS (Computer Algebra System) software, Maxima may appear very complicated and difficult to use, even for the resolution of simple high school or calculus problems. This doesn't have to be the case though, whether you are looking for more math resources to use in your career or a student in an online bachelor's degree in math looking for homework help, Maxima is very friendly and this 10 minute tutorial will get you started right away. Once you've got the first steps down, you can always look up the specific function that you need, or learn more from Maxima's official manual. Alternatively, you can use the question mark followed by a string to obtain in-line documentation (e.g. ? integrate). This tutorial takes a practical approach, where simple examples are given to show you how to compute common tasks. Of course this is just the tip of the iceberg. Maxima is so much more than this, but scratching even just the surface should be enough to get you going. In the end you are only investing 10 minutes.

Maxima as a calculator

You can use Maxima as a fast and reliable calculator whose precision is arbitrary within the limits of your PC's hardware. Maxima expects you to enter one or more commands and expressions separated by a semicolon character (;), just like you would do in many programming languages.
`(%i1) 9+7;` `(%o1)` 16 `(%i2) -17*19;` `(%o2)` -323 `(%i3) 10/2;` `(%o3)` 5
Maxima allows you to refer to the latest result through the % character, and to any previous input or output by its respective prompted %i (input) or %o (output). For example:
`(%i4) % - 10;` `(%o4)` -5 `(%i5) %o1 * 3;` `(%o5)` 48
For the sake of simplicity, from now on we will omit the numbered input and output prompts produced by Maxima's console, and indicate the output with a => sign. When the numerator and denominator are both integers, a reduced fraction or an integer value is returned. These can be evaluated in floating point by using the *float* function (or *bfloat* for big floating point numbers):
`8/2; =>` 4 `8/2.0; =>` 4.0 `2/6; =>` $\frac{1}{3}$ `float(1/3); =>` 0.33333333333333 `1/3.0; =>` 0.33333333333333 `26/4; =>` $\frac{13}{2}$ `float(26/4); =>` 6.5
As mentioned above, big numbers are not an issue:
`13^26; =>` $91733330193268616658399616009$ `13.0^26 =>` $9.1733330193268623\ 10^{+28}$ `30!; =>` $265252859812191058636308480000000$ `float((7/3)^35); =>` $7.5715969098311943\ 10^{+12}$

Constants and common functions

Here is a list of common constants in Maxima, which you should be aware of:

- %e – Euler's Number
- %pi – π
- %phi – the golden mean ($\frac{1+\sqrt{5}}{2}$)
- %i – the imaginary unit ($\sqrt{-1}$)
- inf – real positive infinity (∞)
- minf – real minus infinity ($-\infty$)
- infinity – complex infinity

We can use some of these along with common functions:

`sin(%pi/2) + cos(%pi/3); =>` $\frac{3}{2}$ `tan(%pi/3) * cot(%pi/3); =>` 1 `float(sec(%pi/3) + csc(%pi/3)); =>` $3.1547005383792\overline{5}2$ `sqrt(81); =>` 9 `log(%e); =>` 1

Defining functions and variables

Variables can be assigned through a colon ':' and functions through ':='. The following code shows how to use them:

`a:7; b:8; =>` 7 `=>` 8 `sqrt(a^2+b^2); =>` $\sqrt{113}$ `f(x):= x^2 -x + 1; =>` x^2-x+1 `f(3); =>` 7 `f(a); =>` 43 `f(b); =>` 57

Please note that Maxima only offers the natural logarithm function *log.log10* is not available by default but you can define it yourself as shown below:

`log10(x):= log(x)/log(10); =>` $log10(x) := \frac{log(x)}{log(10)};$ `log10(10) =>` 1

Symbolic Calculations

factor enables us to find the prime factorization of a number:

`factor(30!); =>` $2^{26}\,3^{14}\,5^7\,7^4\,11^2\,13^2\,17\,19\,23\,29$

We can also factor polynomials:

`factor(x^2 + x -6); =>` $(x-2)(x+3)$

And expand them:

`expand((x+3)^4); =>` $x^4+12\,x^3+54\,x^2+108\,x+81$

Simplify rational expressions:

`ratsimp((x^2-1)/(x+1)); =>` $x-1$

And simplify trigonometric expressions:

`trigsimp(2*cos(x)^2 + sin(x)^2); =>` $\cos^2 x+1$

Similarly, we can expand trigonometric expressions:

`trigexpand(sin(2*x)+cos(2*x)); =>` $-\sin^2 x+2\,\cos x\,\sin x+\cos^2 x$

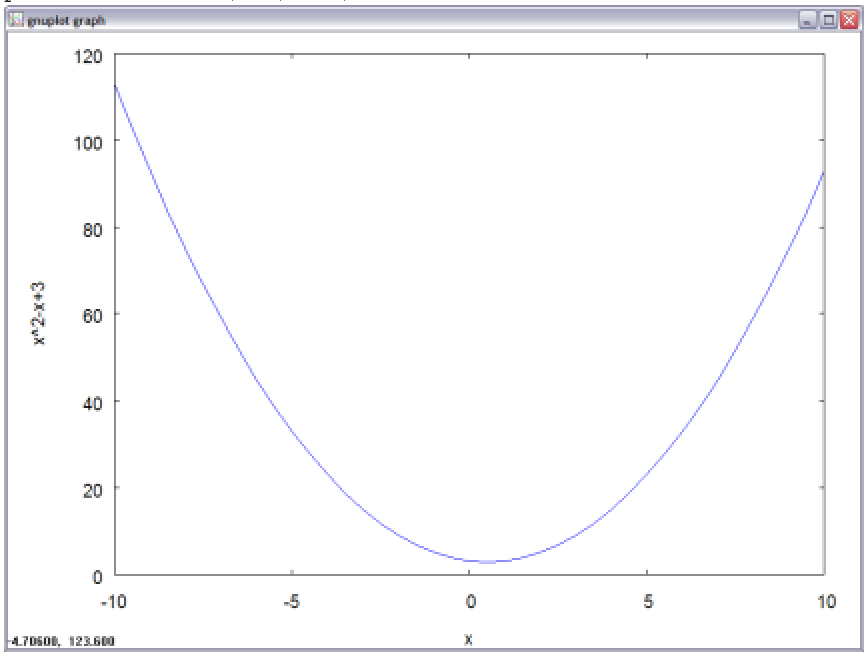
Please note that Maxima won't accept 2x as a product, it requires you to explicitly specify 2*x. If you wish to obtain the TeX representation of a given expression, you can use the *tex* function:

```
tex(%); => $$-\sin ^2x+2\,\cos x\,\sin x+\cos ^2x$$
```

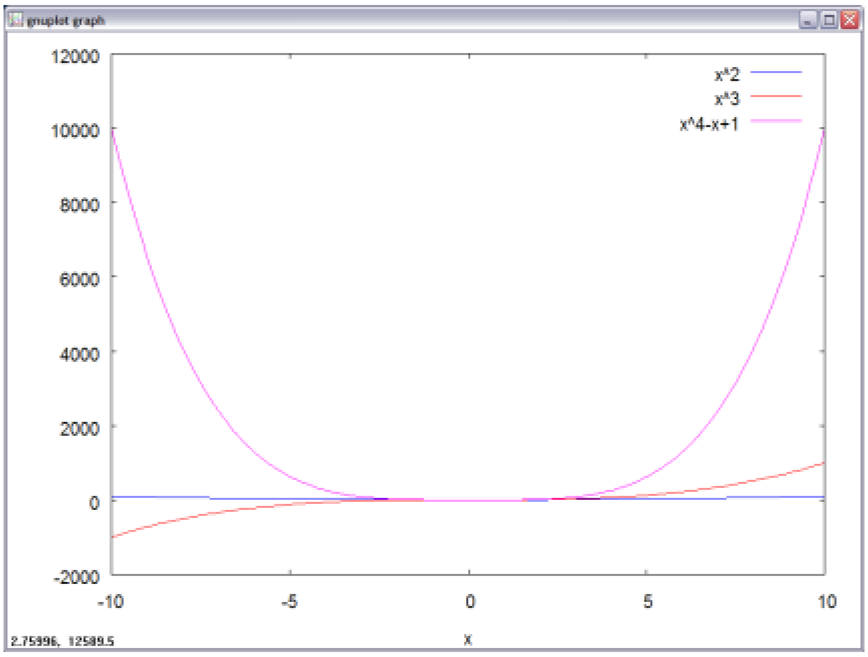
Solving Equations and Systems

We can easily solve equations and systems of equations through the functions*olve*:
`solve(x^2-4,x); =>` $[x=-2,x=2]$ `%[2] =>` $x=2$ `solve(x^3=1,x); =>`
$\left[x=\frac{\sqrt{3}\,i-1}{2},x=-\frac{\sqrt{3}\,i+1}{2},x=1\right]$ `trigsimp(solve([cos(x)^2-x=2-sin(x)^2], [x])); =>` $[x=-1]$ `solve([x - 2*y = 14,  x + 3*y = 9],[x,y]); =>`
$[[x=12,y=-1]]$

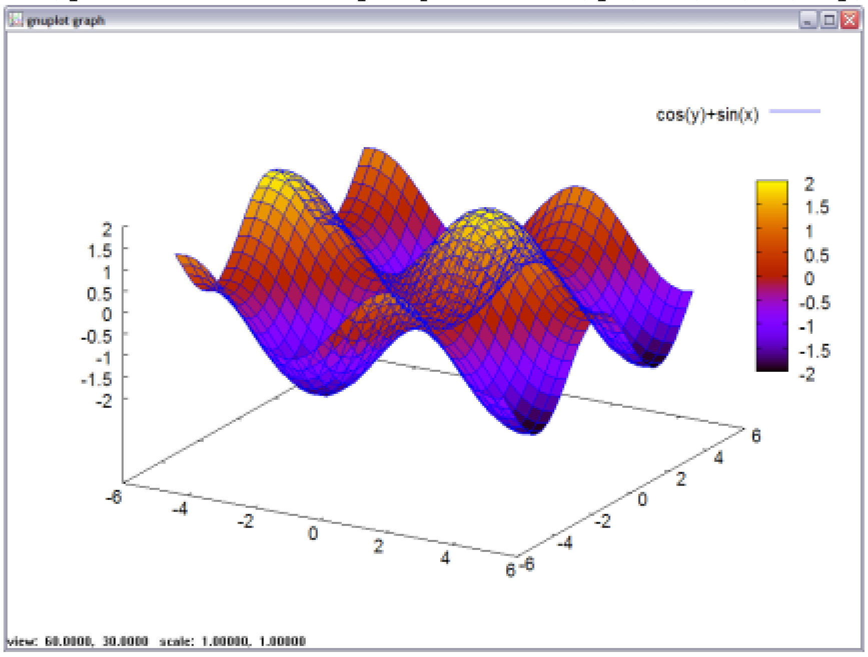
2D and 3D Plotting

Maxima enables us to plot 2D and 3D graphics, and even multiple functions in the same chart. The functions *plot2d* and *plot3d* are quite straightforward as you can see below. The second (and in the case of plot3d, the third) parameter, is just the range of values for x (and y) that define what portion of the chart gets plotted.
```
plot2d(x^2-x+3,[x,-10,10]);
```


```
plot2d([x^2, x^3, x^4 -x +1] ,[x,-10,10]);
```



`f(x,y):= sin(x) + cos(y); plot3d(f(x,y), [x,-5,5], [y,-5,5]);`



Limits

`limit((1+1/x)^x,x,inf);` => $\%e$ `limit(sin(x)/x,x,0);` => 1 `limit(2*(x^2-4)/(x-2),x,2);` => 8 `limit(log(x),x,0,plus);` => $-\infty$ `limit(sqrt(-x)/x,x,0,minus);` => $-\infty$

Differentiation

`diff(sin(x), x);` => $\cos(x)$ `diff(x^x, x);` => $x^x\ (\log x + 1)$
We can calculate higher order derivatives by passing the order as an optional number to the *diff function*:

`diff(tan(x), x, 4); =>` $8\sec^2 x \tan^3 x + 16 \sec^4 x \tan x$

Integration

Maxima offers several types of integration. To symbolically solve indefinite integrals use *integrate*:

`integrate(1/x, x); =>` $log(x)$

For definite integration, just specify the limits of integrations as the two last parameters:

`integrate(x+2/(x -3), x, 0,1); =>` $-2\log 3 + 2\log 2 + \frac{1}{2}$ `integrate(%e^(-x^2),x,minf,inf); =>` $\sqrt{\%pi}$

If the function *integrate* is unable to calculate an integral, you can do a numerical approximation through one of the methods available (e.g.*romberg*):

`romberg(cos(sin(x+1)), x, 0, 1); => 0.57591750059682`

Sums and Products

sum and *product* are two functions for summation and product calculation. The *simpsum* option simplifies the sum whenever possible. Notice how the product can be use to define your own version of the factorial function as well.

`sum(k, k, 1, n); =>` $\sum_{k=1}^{n} k$ `sum(k, k, 1, n), simpsum; =>` $\frac{n^2+n}{2}$ `sum(1/k^4, k, 1, inf), simpsum; =>` $\frac{\%pi^4}{90}$ `fact(n):=product(k, k, 1, n); =>` $fact(n) := product(k, k, 1, n)$ `fact(10); =>` 3628800

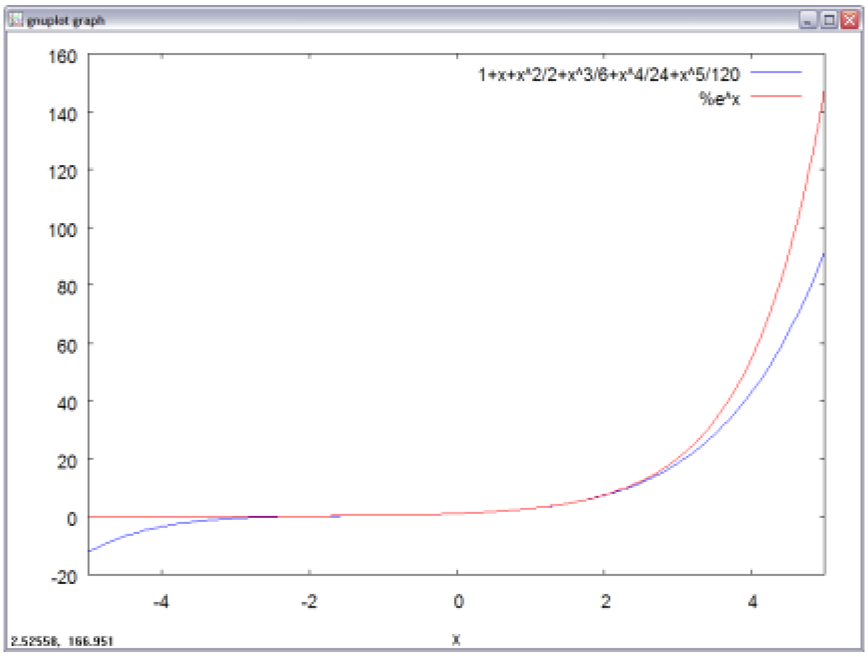
Series Expansions

Series expansions can be calculated through the *taylor* method (the last parameter specifies the depth), or through the method *powerseries*:

`niceindices(powerseries(%e^x, x, 0)); =>` $\sum_{i=0}^{\infty} \frac{x^i}{i!}$ `taylor(%e^x, x, 0, 5); =>`
$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \cdots$

The *trunc* method along with *plot2d* is used when taylor's output needs to be plotted (to deal with the $+\cdots$ in taylor's output):

```
plot2d([trunc(%), %e^x], [x,-5,5]);
```



I hope you'll find this useful and that it will help you get started with Maxima. CAS can be powerful tools and if you are willing to learn how to use them properly, you will soon discover that it was time well invested.