

LINUX SS7 Signaling Card
SS7 I/O cards, protocol stacks
Blades & Signaling Gateways
www.adax.com/products/hdc.html

Master Fotovoltaico
Iscriviti al Master in Energie
Alternative di eCampus. Chiedi info
www.uniecampus.it/master_universta

Master's Degree in Sweden
Watch film clips and read about
Master's degree students' lives
www.MasterOfScience.se

Ads by Google

[Home](#) [Topics](#) [Community](#) [Forums](#) [Magazine](#) [Shop](#) [Buyer's Guide](#) [Archive CD](#)

[Home](#)

Exploring Advanced Math with Maxima

July 20th, 2009 by Mike Diehl in [Education](#)

Digg

[submit](#)

Average:

Your rating: None Average: 4.8 (12 votes)

When I took Calculus in college, my Professor would give us substantial partial credit for test problems that we got wrong for minor arithmetic errors, and rightfully so, too. Sometimes even simple-sounding problems resulted in a full page, or more, of calculations. Simply changing a -1 to a +1 early on in a problem could be completely devastating. Avoiding simple errors like that is what Computer Algebra Systems (CAS) are all about and I've found Maxima to be an extremely powerful CAS program.

Maxima is licensed under the GPL, so it has to be good! However, Maxima is a text based program, which means it's a bit clumsy by today's standards. Fortunately, there are a few front ends which make the program much easier to use and more polished. For this article, I'm using the wxMaxima front end.

Anyone who's taken a high school Algebra class remembers spending endless hours factoring polynomials, multiplying them together, and evaluating them for given values of x. Factoring polynomials always seemed to have an element of guess work built in, and I found it to be quite frustrating. Of course, the teacher spent weeks teaching various ways to factor these formulas, and then they finally introduced the Quadratic Equation, which made factoring second order polynomials almost trivial. For those of you who don't completely geek out over Math, a second order polynomial is one like this: $ax^2 + bx + c$, where the highest exponent is 2.

Manipulating polynomials with Maxima is very easy, as shown in Figure 1. Here you see 7 steps that I've asked Maxima to perform. My commands are indicated by the "%i" and a step number. The corresponding output is indicated by the "%o" and then the step number. So, (%i90) is my 90th input step and the result is on the line indicated by (%o90). So, at step 90, you see that I entered a simple polynomial equation. Then I asked Maxima to factor it. Here you see that I used a shortcut. The "%" character represents the return value of the previous step. So, when I entered "factor(%);" Maxima retrieved the previous step's value (that polynomial) and factored it. The expected result is on the next line. In step 92, I entered two polynomials multiplied by each other. As you can see, Maxima doesn't immediately do anything special with this formula. But when I asked it to expand the formula, it multiplied the two together, as you can see in step 93. At step 94, I asked Maxima to factor the resulting 4th order polynomial, and behold, we get the original equation back, as we expect. I don't think we even discussed factoring 4th order polynomials in high school Algebra! But we did a lot of evaluating polynomials at a given value of x, which I demonstrate in steps 95 and 96. In step 95, I assign a value to x. In step 96, I re-enter the polynomial, and Maxima calculates it's value at the given value of x.

[Subscribe](#) [Renew](#) [Free Issue](#) [Customer service](#)



The advertisement features a thumbnail of a Linux Journal magazine cover on the left. To its right, the text "15 Years Later" is displayed in a large, bold font. Below the thumbnail is a red banner with the "LINUX JOURNAL" logo in white. On the right side of the red banner, there is a white button with the text "Subscribe Now".

The Latest

- [3 recent OpenOffice.org extensions](#) Nov-02-09
 - [Updating ISOs with zsync](#) Oct-29-09
 - [Linux Journal 2010 Wall Calendars Now Available](#) Oct-28-09
 - [More on Poulsbo \(GMA500\), Intel, and the Community](#) Oct-28-09
 - [Tech Tip: Save an Online Video with your Browser \(no extensions needed\)](#) Oct-28-09
 - [Finally! SecDef signs Clarifying Guidance Regarding Open Source Software](#) Oct-27-09
- [more](#)

Newsletter

Each week *Linux Journal* editors will tell you what's hot in the world of Linux. You will receive late breaking news, technical tips and tricks, and links to in-depth stories featured on www.linuxjournal.com.

Sign up for our Email Newsletter

```
(%i90) x^2+2*x+1;
(%o90) x^2 + 2x + 1

(%i91) factor(%);
(%o91) (x + 1)^2

(%i92) (x^2+2*x-1)*(2*x^2-9*x+1);
(%o92) (x^2 + 2x - 1)(2x^2 - 9x + 1)

(%i93) expand((x^2+2*x-1)*(2*x^2-9*x+1));
(%o93) 2x^4 - 5x^3 - 19x^2 + 11x - 1

(%i94) factor(2*x^4-5*x^3-19*x^2+11*x-1);
(%o94) (x^2 + 2x - 1)(2x^2 - 9x + 1)

(%i95) x:5;
(%o95) 5

(%i96) (x^2+2*x-1)*(2*x^2-9*x+1);
(%o96) 204
```

Figure 1. Manipulating Polynomials

Of course, Maxima can also solve equations, both symbolically, and numerically, as in Figure 2. Here we see that I've asked it to solve a quadratic equation, and it gives me the answer in symbolic form. Then I used the "numer" directive to ask Maxima to give its answers in numerical form, which it does in step 13. Finally, I ask Maxima to use 25 digits of accuracy, instead of the default 16, and recalculate. Maxima will calculate to any degree of accuracy that you want and can handle arbitrarily large numbers. For example, Maxima can easily handle 2^{200} , and can even factor $(2^{200})-1$.

```
(%i11) solve([x^2-5*x-3=0],x);
(%o11) [ x = -\frac{\sqrt{37}-5}{2}, x = \frac{\sqrt{37}+5}{2} ]

(%i12) numer: true;
(%o12) true

(%i13) solve([x^2-5*x-3=0],x);
`rat' replaced 6.082762530298219 by 10657/1752
`rat' replaced 6.082762530298219 by 10657/1752
(%o13) [ x = - 0.54138127853881 , x = 5.54138127853881 ]





(%i14) fpprec: 25;
(%o14) 25

(%i15) bfloat(solve([x^2-5*x-3=0],x));
`rat' replaced 6.082762530298219 by 10657/1752
`rat' replaced 6.082762530298219 by 10657/1752
Warning: Float to bigfloat conversion of -0.54138127853881
Warning: Float to bigfloat conversion of 5.54138127853881
(%o15) [ x = - 5.413812785388127853881279b-1 , x = 5.5413812785388127853881279b+1 ]
```

Figure 2. Symbolic and Numeric Manipulation

Maxima will even handle calculations that you'd never even think about doing manually. For example, you can ask Maxima to solve a 4th order polynomial equation, and it will give you all 4

Tech Tip Videos

- Updating ISOs with zsync Oct-29-09 
- Forgetting Sudo (we've all done it) Oct-26-09 
- Installing Truetype Fonts on Your Linux System Oct-21-09 
- Perl-y Tweets with TTYtter Oct-19-09 

See more Linux Journal Videos

Recently Popular

- Boot with GRUB
- Why Python?
- Tech Tip: Save an Online Video
- Browser (no extensions need)
- Updating ISOs with zsync
- Validate an E-Mail Address with
- Right Way
- How To Kick Your Friends in the Face:
- GMA500

in case of huge numbers, to see all the digits, we have to set dispaly('ascii)

more

From the Magazine

November 2009, #187

It doesn't matter how big your infrastructure is, even if it's planetary sized, Linux can handle it. Got massive amounts of data to analyze? Check out our article on IBMs InfoSphere Streams. Need a SAN on a budget, use Linux to provide it. Messaging problems, try AMQP (Advanced Message Queuing Protocol). In addition to our feature articles, don't miss our articles on RSpec, DEFCON, Ext3 vs XFS, Virtualization, HIPL, Pokerth, X-Moto and more.



Read this issue



- Contribute to the Linux Journal Flickr Pool
- Follow Linux Journal on Twitter
- Add Linux Journal to your RSS reader
- Meet other readers on Facebook
- Get Linux Journal Miis



solutions, both real and complex. However, in symbolic form, these solutions are often so ugly that they scroll off your screen! But, you can then turn around and ask it to give you the same answers in numerical form, to any degree of accuracy that you need.

Maxima uses either GnuPlot or Xmaxima to plot expressions and can plot 2D, 3D and parametric equations. In Figure 3, I entered a fairly complex expression, and asked Maxima to plot it on a 2D plot. In Figure 4, I took an example from Maxima's documentation and plotted a 3D image. You can rotate this image and view it from any angle you like.



```
(%i65) (x-1)*(x+1)*(x+1/2)/((x-2)*(x+2)*(x-1/2))
```

```
(%o65) (x - 1) (x + 1/2) (x + 1) / ((x - 2) (x - 1/2) (x + 2))
```

```
(%i66) plot2d((x-1)*(x+1)*(x+.5)/((x-2)*(x+2)*(x-.5)), [x, -4, 4], [y, -10, 10]);
```

```
(%o66)
```

```
(%i67)
```

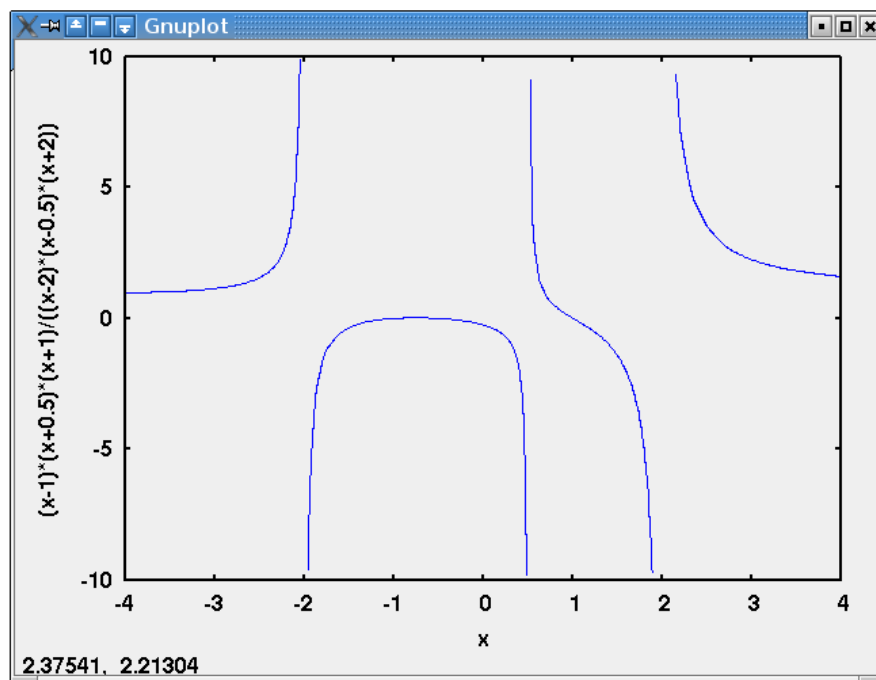


Figure 3. 2D Plot of an Expression

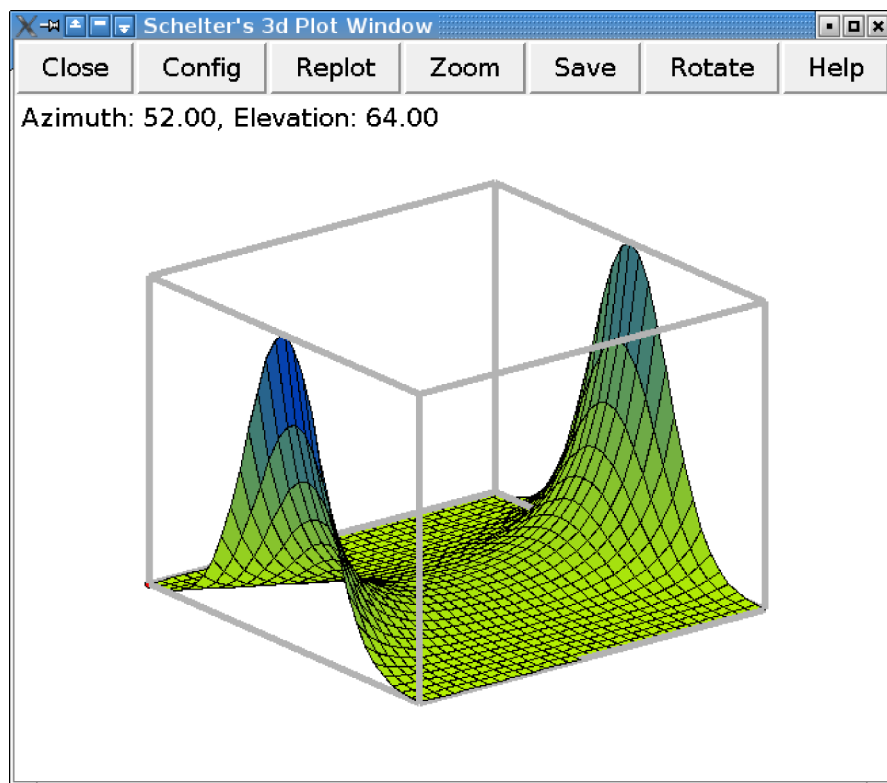


Figure 4. 3D Plot of an Expression

If all Maxima was able to do was manipulate polynomials, it would still be a powerful tool. But Maxima can do much more, as I'll try to demonstrate in Figure 5. Here, in step 16, I asked Maxima to expand a Trigonometric expression using the Double Angle Theorem. Even though I found Trig to be the easiest Math class I ever took, Maxima can manipulate Trigonometric Identities much better than I ever will. In step 17, I asked Maxima to differentiate $\sin(3x)$ with respect to x . This is a fairly simple derivative to calculate. There's nothing simple about the derivative I asked for in step 18. If we were performing this calculation manually, we would probably use a combination of the Product Rule and the Quotient Rule, or multiple instances of the Quotient Rule. Then we would have to simplify the result. Either way, "yuck" is a good way to describe the process. Maxima provides us with a nicely formatted answer. In step 19, I asked Maxima to calculate an indefinite integral with respect to x . We see Maxima's answer on the next line. Ya, that's what I came up with; how about you? In step 20, I asked Maxima to calculate the definite integral of the same function as x goes from 0 to 5. Of course, I could have asked for the answer in numerical format but the symbolic form is so much more geeky. In step 21, we see that Maxima can apply L'Hospital's rule to calculate the limit of an expression as x goes to infinity.

```
(%i16) trigexpand(sin(2*x));
```

```
(%o16) 2 cos(x) sin(x)
```

```
(%i17) diff(sin(3*x), x);
```

```
(%o17) 3 cos(3 x)
```

```
(%i18) diff((x-3)*(x+3)/(x-1));
```

```
(%o18)  $\left( \frac{x+3}{x-1} - \frac{(x-3)(x+3)}{(x-1)^2} + \frac{x-3}{x-1} \right) del(x)$ 
```

```
(%i19) integrate(x*sin(log(x)), x);
```

```
(%o19)  $\frac{x^2(2 \sin(\log(x)) - \cos(\log(x)))}{5}$ 
```

```
(%i20) integrate(x*sin(log(x)), x, 0, 5);
```

```
(%o20) 10 sin(log(5)) - 5 cos(log(5))
```

```
(%i21) limit((x^2+2)/(3*x^2-1), x, inf);
```

```
(%o21)  $\frac{1}{3}$ 
```

```
(%i22) taylor(sin(x), x, 0, 15);
```

```
(%o22)  $x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362880} - \frac{x^{11}}{39916800} + \frac{x^{13}}{6227020800} - \frac{x^{15}}{1307674368000} + \dots$ 
```

Figure 5. More Maxima Manipulations

In step 22, I asked Maxima to calculate the first several terms of the Taylor Series of $\sin(x)$ centered around $x=0$. This is where we start to see a weakness in Maxima. There should have been a factorial in the denominator of each term of that Taylor Series. But Maxima simply gave us the answer in numerical form.

Maxima is also able to manipulate matrices and sets, though I'm not going to demonstrate any of that functionality here. Finally, Maxima has a built-in programming language, though it's not the most elegant language you'll ever see. I show you a short example in Figure 6. Here I assign an initial value to x . Then I calculate the value of an expression 10 times, using the previous result as the next value of x . There are only a few initial values of x for which this expression ever repeats itself. Can you find them?

```
(%i24) x : 2.1
```

```
(%o24) 2.1
```

```
(%i25) for i thru 10 do block(x : 4x - x^2, a : print(### , i, x))
```

```
### 1 3.99
```

```
### 2 0.0398999999999999
```

```
### 3 0.15800799
```

```
### 4 0.60706543509615
```

```
### 5 2.059733297896125
```

```
### 6 3.996431933122453
```

```
### 7 0.014259536408947
```

```
### 8 0.056834811257188
```

```
### 9 0.22410904925811
```

```
### 10 0.84621133107308
```

```
(%o25) done
```

```
(%i26)
```

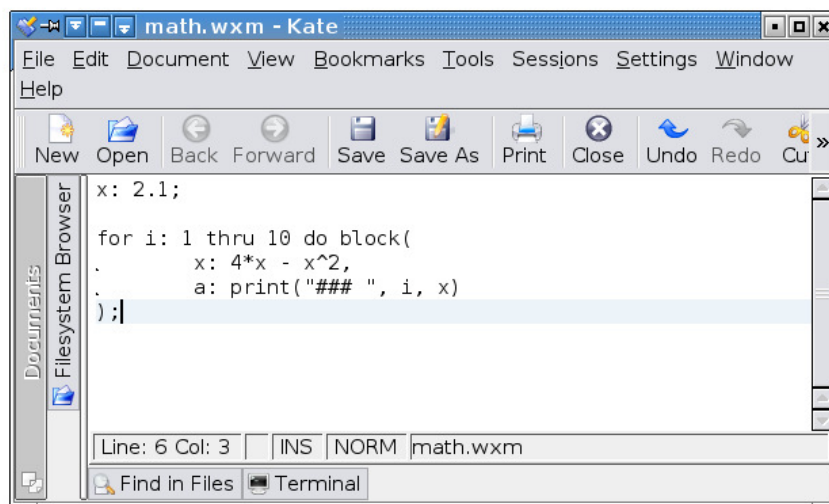


Figure 5. Built-In Programming Language

As you can see, Maxima is a very sophisticated analysis tool and is adept at solving problems in Algebra, Trigonometry, Linear Algebra, Differential Calculus, and Integral Calculus. It can also calculate using arbitrarily large and precise numbers. With all that capability, you might be tempted to ask why we even need to teach Mathematics anymore and that's the real point of this article. For all it can do, Maxima has no understanding of any of the questions you ask it; it only knows how to calculate answers. The value of teaching and learning Mathematics is in understanding concepts and thought processes. The value in using a tool like Maxima is in removing some of the tedium from the process of Math.

Mike Diehl is a freelance Computer Nerd specializing in Linux administration, programming, and VoIP. Mike lives in Albuquerque, NM, with his wife and 3 sons. He can be reached at mdiehl@diehlnet.com

Attachment	Size
Fig1.png	37.23 KB
Fig2.png	58.2 KB
Fig3.png	23.65 KB
Fig4.png	33.23 KB
Fig5.png	48.85 KB
Fig6.png	76.02 KB
cover.png	48.04 KB

Special Magazine Offer -- Free Gift with Subscription

Receive a free digital copy of *Linux Journal's* System Administration Special Edition as well as instant online access to current and past issues. [CLICK HERE for offer](#)

Linux Journal: delivering readers the advice and inspiration they need to get the most out of their Linux systems since 1994.

Related Articles

- [Teaching with Tux](#)
- [Computer Logic Design with KTechLab](#)
- [Teaching Math with the KDE Interactive Geometry Program](#)

Printer-friendly version  Delicious  Digg  StumbleUpon  Reddit
 Facebook  Post to Twitter



**4 OPPORTUNITÀ
PER IL TUO FUTURO
NELLA MODA E NEL DESIGN**

MASTER POST LAUREA

- Fashion Product & Programme Manager
- Luxury Goods Communication Manager
- Luxury Brand Manager
- DELUXE Design Luxury Excellence

ACCADEMIA DEL LUSSO
ISCRIZIONI APERTE A.A. 2009/10
CLICCA X INFO

Comment viewing options

Threaded list - expanded ▾ Date - newest first ▾ 50 comments per page ▾

Select your preferred way to display the comments and click "Save settings" to activate your changes.

Re: Something wrong with accuracy?

On October 9th, 2009 Anonymous says:



No, it's just that Maxima provides a variable, `fpprec`, that controls the precision of its bigfloat computations and a variable, `fpprintprec`, that controls the number of digits printed. Here is what the correct output looks like:

```
(%i18) solve(x^2-5*x-3,x);
(%o18) [x = -(sqrt(37)-5)/2,x = (sqrt(37)+5)/2]
(%i19) %numer;
(%o19) [x = -.5413812651491097,x = 5.54138126514911]
(%i20) (fpprintprec : fpprec : 25, bfloat(%o18));
(%o20) [x = -5.413812651491098444998421b-1,x = 5.541381265149109844499842b0]
```

I believe that %o20 agrees with what you expect.

[reply](#)

You might also want to look

On July 21st, 2009 Anonymous (not verified) says:



You might also want to look into Sage, as it's a single python interface that **combines a number of math packages** including Maxima.

[reply](#)

Something wrong with accuracy?

On July 21st, 2009 Anonymous (not verified) says:



From the logs it looks like `SQRT(13)` was replaced in step 13 with an approximate value that is accurate to 8 digits only, although the precision of numerical values was set to 16.

In step 15 you set `fpprec` to 25 digits, but again, the end results are accurate to 7 digits only. (The solutions should be: `-.5413812651491098444998421` and `5.5413812651491098444998421`). Is there a way to avoid these problems?

[reply](#)

Fantastic Tool

On July 20th, 2009 Salvadesswaran Srinivasan (not verified) says:



I've always loved maths, especially all the algebra and calculus, but I'd love to see others who hate maths just because of this use this fantastic tool, it might just make them love maths. When I teach(maybe 6-7 years on) I'll recommend this to my students :)

[reply](#)

It sure looks very nice and

On July 20th, 2009 Ruben (not verified) says:



It sure looks very nice and sophisticated, but how does it compare to the alternatives? And then I mean of course the commercial stuff they use in colleges and things like Wolfram Alpha, which is supposed to have some impressive calculating power as well? When I start my study of Applied Physics later this year, I can maybe find this all out by myself. But can you give a quick comparison maybe?

Btw, great article, very pleasant to read!

[reply](#)

Post new comment

Please note that comments may not appear immediately, so there is no need to repost your comment.

Your name:

E-mail:

The content of this field is kept private and will not be shown publicly.

Homepage:

Subject:

Comment: *

- Allowed HTML tags: <a> <cite> <code> <pre> <dl> <dt> <dd> <div>
- Lines and paragraphs break automatically.

[More information about formatting options](#)