

Re: Computer Algebra Algorithms

Source: <http://sci.tech-archive.net/Archive/sci.math.symbolic/2005-04/msg00037.html>

- *From:* Bernard Parisse <parisse@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sat, 09 Apr 2005 08:18:50 +0200
-

Actually, there are C++ programmed CAS, like my giac/xcas
www-fourier.ujf-grenoble.fr/~parisse/giac.html
They are more flexible for programmers because they are available
as C++ libraries, not only as a binary.

You're link doesn't work. Do I get from what you are saying
you
code is not open source?

It was probably a temporary failure, at this time it works. I rewrite
it to be sure it is the correct link:

<http://www-fourier.ujf-grenoble.fr/~parisse/giac.html>

And giac/xcas is open-source (GPL)

If it is not open source how are people going
to learn the algorithms by exploring the code? How does your
system
compare to Maxima, Axiom or Derive?

That is not something you can say in a few words, it would require
an extensive study. I would say that it is comparable to maxima for
CAS functions. I don't know enough Axiom to compare.

One point for xcas is that maple users
can choose maple compatibility mode, in that mode
the syntax is the same as in maple, and there is even a chance
that they can run their maple worksheet without modification
inside xcas.

The best way to compare it is to test it!

Personally I find C code
aesthetically unpleasing. If I was to extensively study computer
algebra I would not want to spend my time reading such a low level
language as C. But to each there own.

Re: Computer Algebra Algorithms

Giac/xcas is written in C++, not C, some low level code is therefore handled directly by the language, e.g. memory handling using constructors and destructors.

Anyway, a computer algebra system is another language in itself. It has its own rules and syntax. People generally prefer righting there numeric algorithms in Maple and MATLAB then they do in C. They will only resort to a low level language like C for efficiency or interfacing reasons.

This is one reason I choosed C++ as language for CAS development, because it should not be too hard to add a giac/xcas->C++ translator, so that once the code works in xcas user language, it can be compiled (after the user make some changes if necessary).
One other reason is that there are many scientific C libraries.

I am focusing on learning Maple and Hakell. Maple because it is used via the MATLAB symbolic package and Haskell because it is what a bunch of experts thought lazy pure functional programming language should be. I think learning a computer language is easy and once I learn Haskell the other functional languages (example ML (metta language)) should come quickly. Logical programming might take some time.

I believe that you can have an idea of a language relatively fast, but it requires much more time to really master it.

.