Giac and its interfaces

Frederic Han

Université Paris 7, IMJ-PRG

September 3, 2015
Giac is the main C++ library used in the CAS **Xcas**

- **author:** Bernard Parisse (Institut Fourier Grenoble, France)  
  [www-fourier.ujf-grenoble.fr/~parisse/giac.html](http://www-fourier.ujf-grenoble.fr/~parisse/giac.html)
- **2000-**
- **GPLv3+**
- formal calculus by GIAC helped by GMP, MPFR, PARI, NTL ...  
- numeric computations, plots, stats ... uses GSL, MPFI ...
- Native binaries provided for linux, macos, windows 32 and 64. (Fastest is linux 64)
- For low configurations (ex Pocket Calculator HP Prime), most functions stay available without PARI, NTL, GSL, MPFI...
Xcas
The FLTK based GUI by B. Parisse

- FLTK based GUI. (well documented)
- Symbolic style similar to maple. (no ring, no conversions, expressions are not automatically expanded..., rootof...)
- Educational software mixing symbolic calculus and interactive geometry 2D and 3D OpenGL).
- Symbolic and numeric solver for equations and some inequations, integration ...
- Interesting implementation of products and groebner basis.
Other interfaces with giac syntax

- the console interface provided by B. Parisse is named **icas** or **giac** (linux, macos, windows)
- Since 2012 there is a Qt interface: **Qcas** by Loic Lecoq and FH.
  - [http://webusers.imj-prg.fr/~frederic.han/qcas](http://webusers.imj-prg.fr/~frederic.han/qcas)
  - Calculus is implemented with mathml output. Most of the 2D output and interactive geometry is implemented but 3D output is not implemented nor the spreadsheet.
Giacpy: a Cython interface to giac
http://webusers.imj-prg.fr/~frederic.han/xcas/giacpy

- One version for python 2 or 3.
  - debian packages available in giac’s debian repo.
  - windows binaries for some python versions are available on my page.

- A fork for sage.
  - use of gmp integers between sage/giac instead of strings.
  - optional spkg of giac and giacpy are available since sage 6.8.
Building interactive html pages from latex files

(Develloppement feature. giac.js is built from giac with **emscript**)  

- giac.tex + giac.js + hevea.sty ⇒ interactive html pages.

Online example to test yourself at:  
www-fourier.ujf-grenoble.fr/~parisse/giac/xcasen.html
On my 2012 notebook, computing offline in firefox:

- $\text{ifactor}(2^{128}+1)$ in 0.15s, $\text{ifactor}(3^{128}+1)$ in 2.4s.
  (2 times slower than with the C++ giac library)

- with $f := \text{normal}((x+y+z+1)^{32}+1); (6545 \text{ terms})$ then expanding the product with $\text{normal}(f*(f+1));$ takes 3.5s
  (10 times slower than with the C++ giac library)
Thank You