

# Introduction to xcas/giac

B. Parisse

R. De Graeve

[www-fourier.ujf-grenoble.fr/~parisse/giac.html](http://www-fourier.ujf-grenoble.fr/~parisse/giac.html)

May 27, 2002

## Project status :

- GPL license
- general CAS + interactive geometry. First target is educational.
- 2 years old (started as a project over GiNaC), 70 000 lines of C(++) code,
- compiled under Unix/Linux (x86, ARM) and Windows (using GNU autoconfiguration thanks to R. Kreckel)
- Built over GMP and the C++ STL (Standard Template Library); User-interface with FLTK/FLVW or GNU readline, 3-d plotting with gnuplot, latex output.

- Library (libgiac) + user interfaces (GUI xcas, text cas, under texmacs)
- On-line help + Documentation
- Partial internationalization support (J.M.L. de la Fuente)  
`export LANGUAGE=en, fr, es`
- Interfaces in progress with C/C++ libraries (numeric GSL, number theory PARI, 1-d polynomial NTL, realsolving RS (?)...)
- Programmable in C++ (standalone or loadable module) or with restricted maple/mupad-like language with an interactive debugger.
- Scriptable regression tests and benchmarks

Example of xcas session :

- Using menus, on-line help, configuration
- Compute an integral  
`int(1/(x^2+1)^10,x,0,1)`
- Plot a 2-d/3-d function,  
`plotfunc(sin(x),x)`  
`plotfunc(x^2-y^2,[x,y])`
- Eigenvectors of a matrix  
`egv([[1,2,3],[4,5,6],[7,8,9]])`
- Draw 3 points  $A, B, C$  then an ellipsis with focus  $A, B$  containing  $C$   
`ellipse(A,B,C)`  
move the points
- $\text{\LaTeX}$  output of the session

Example of program :

- The integer gcd with xcas language (hello world for a CAS language!)

```
pgcd(a,b):={  
  local r;  
  for (;b!=0;){  
    r:=irem(a,b);  
    a:=b;  
    b:=r;  
  }  
  return(a);  
}
```

- Maple/Mupad-like translation
- step-by-step execution  
debug(pgcd(15,25))

- `#include <giac/giac.h>`  
`using namespace std;`  
`using namespace giac;`

```
gen pgcd(gen a,gen b){  
    gen q,r;  
    for (;b!=0;){  
        r=irem(a,b,q);  
        a=b;  
        b=r;  
    }  
    return a;  
}
```

```
int main(){  
    cout << "Enter 2 integers";  
    gen a,b;  
    cin >> a >> b;  
    cout << pgcd(a,b) << endl;  
    return 0;  
}
```

- Example of loadable module `pgcd.cpp`

## Future directions :

- “Pretty print”, equationwriter (?)
- numbering versions, CVS, rpm/deb packages
- Calculator language support (TI89/92, HP49)
- Improve maths (modular, mrv preprocessing, Risch, ode, Gosper, Zeilberger, ...)
- More Maple instructions support
- Improve spreadsheet, add statistics, units
- More connections to other libraries
- Semi-classical analysis