MATHEMAGIX
The quest of modularity and efficiency for symbolic and certified numeric computation*

Joris van der Hoeven – vdhoeven@lix.polytechnique.fr†
Grégoire Lecerf – gregoire.lecerf@math.cnrs.fr†
Bernard Mourrain – bernard.mourrain@inria.fr‡
Philippe Trébuchet – philippe.trebuchet@lip6.fr§
Jérémy Berthomieu – berthomieu@lix.polytechnique.fr†
Daouda Niang Diatta – dndiatta@lix.polytechnique.fr†
Angelos Mantzaflaris – angelos.mantzaflaris@inria.fr‡

May 6, 2011

Abstract

We describe the goals and architecture of the software project MATHEMAGIX, the main list of packages it contains, the main characteristics of its programming language, and connections with existing front-ends.

1 Context and objectives

In the recent years, appeared general purpose open source software for symbolic and numeric computations such as AXIOM (www.axiom-developer.org), MAXIMA (maxima.sourceforge.net), OCTAVE (www.gnu.org/software/octave), SAGE (www.sagemath.org), SCILAB (www.scilab.org), etc. featuring an easier integration of different specialized packages. The MATHEMAGIX project follows this mainstream, without compromising modularity for efficiency or vice-versa. It aims at developing a new complete framework including a language well suited to high level mathematical purposes, librairies for low and high level functionalities, and confortable front-ends. MATHEMAGIX is freely distributed mainly under the GPL license. Downloading, installation instructions, documentation, tutorials and examples are available from www.mathemagix.org.

*This work has been partly supported by the French ANR-09-JCJC-0098-01 MAGIX project, and by the DIGITEO 2009-36HD grant of the Région Ile-de-France.

†Laboratoire d'informatique (CNRS, UMR 7161), École polytechnique, Palaiseau, France, 91128.
‡Projet GALAAD, INRIA Sophia Antipolis, France, 06902.
§Laboratoire d'informatique de Paris 6 (CNRS, UMR 7606), équipe APR, Paris, France, 75005.

2 Towards a new high level language

The ultimate design goals of the MATHEMAGIX language are the following:

Strong typedness. The language is strongly typed, with support for overloading, implicit conversions, generic objects, compile-time type checking and, possibly, built-in support for expression types which interact with the type system. One simple, but suggestive example of a strongly typed declaration is the following:

```
forall (R: Ring) cube (x: R): R == x*x*x;
```

High level control structures, like coroutines, generators, exceptions, continuations, etc.

Reusability of external libraries. Besides achieving runtime efficiency of MATHEMAGIX itself, the aim is to achieve runtime efficiency through the extensive reuse of existing dedicated libraries written in other languages. MATHEMAGIX therefore implements transparent mechanisms for reusing external libraries and in particular C++ template libraries.

Currently, a rather slow interpreter of only a part of the full language, the `mmxlight` package, is provided. A compiler and a new faster interpreter, written in the MATHEMAGIX language, are under development in the `mmcompiler` package.

3 Available packages

Most of the packages are written in C++ and in the MATHEMAGIX language. We provide a coherent open source framework for the development and publication of efficient and stand-alone packages, together with tools for their connection to the MATHEMAGIX interpreter. Our central packages contain the following implementations:

basix: symbols, lists, arrays, hash tables, vectors, generic objects, parsers, pretty printers.

numerix: integers, rational numbers based on GMP (`gmplib.org`), floating point numbers based on MPFR (`www.mpfr.org`), complexified version of these types, intervals and balls for certified arithmetic, modular numbers.

algebramix: univariate polynomials, series, p-adic numbers, matrices.

multimix: dense and sparse multivariate polynomials and series.

analyziz: effective analytic computation, including analytic continuation and manipulation of effective analytic functions.

linalg: basic linear algebra tools on dense matrices, with any suitable coefficients (`float`, `double`, `long double`, extended precision, and modular arithmetic). Its interface is fully compliant with the standard BLAS (`www.netlib.org/blas`) and LAPACK (`www.netlib.org/lapack`) libraries one, allowing us to transparently use a specialized implementation such as ATLAS (`math-atlas.sourceforge.net`) for usual machine type coefficients.

newmac: multivariate polynomial system solver using border bases techniques.

realroot: subdivision solvers for polynomials using Bernstein or monomial basis representation, univariate solver using continued fraction expansion, sleeve approximation and preconditioner.

polytopix: manipulation of polytopes based on the CDD library (`www.ifor.math.ethz.ch/~fukuda/cdd_home`), connection with multivariate polynomial supports.

shape: data structures to manipulate algebraic curves and surfaces which are given by equations or by parameterizations, topology of algebraic sets, arrangements, (self-)intersection curves of parameterized surfaces, singularities.

4 Front-ends

A textual shell mode. MATHEMAGIX provides the user with a usual ASCII shell mode and a convenient advanced programmer interface, based on a dedicated EMACS mode.

A graphical interface. MATHEMAGIX can be used within GNU TeXmacs as a primary graphical interface. This offers a unified and user friendly framework for editing structured documents with different types of content: text, graphics, mathematics, session etc. The rendering engine uses high-quality typesetting algorithms so as to produce professionally looking documents, which can either be printed out or presented from a laptop. TeXmacs is also used to write and to generate the top-level documentation of the packages in different format (`html`, `tex`), in combination with DOXYGEN for the C++ code documentation.

A geometric modeler. The packages `realroot` and `shape` of MATHEMAGIX are directly used in the computational kernel of the algebraic-geometric modeler AXEL (`axel.inria.fr`). Other packages (e.g. `polytopix`) produce independent plugins, which extend the interface and the functionalities of AXEL.

5 Example

The following example, run within TeXmacs and exported into LaTeX, shows how to compute asymptotic expansions in x in the neighborhood of infinity, by means of the algorithms designed by VAN DER HOEVEN in *Meta-expansion of transseries.* JSC, to appear:

```
Mmx] use "asymptotix";
Mmx] x == infinity ('x);                    // declare infinitely large variable x
Mmx] 1 / (1 + x + exp x)
```

$$\frac{1}{e^x} - \frac{x}{e^{2x}} - \frac{1}{e^{2x}} + \frac{x^2}{e^{3x}} + \frac{2x}{e^{3x}} + \frac{1}{e^{3x}} - \frac{x^3}{e^{4x}} - \frac{3x^2}{e^{4x}} - \frac{3x}{e^{4x}} - \frac{1}{e^{4x}} + O\left(\frac{x^4}{e^{5x}}\right)$$

```
Mmx] product (x + log x, x)
```

// *asymptotic expansion of* $\prod_{t=c}^{x}(t+\log t)$ for $x \to +\infty$

$$e^{x\log(x)-x+\frac{\log(x)^2}{2}-\frac{\log(x)}{2}} + \frac{\log(x)^2 e^{x\log(x)-x+\frac{\log(x)^2}{2}-\frac{\log(x)}{2}}}{2x} + \frac{\log(x) e^{x\log(x)-x+\frac{\log(x)^2}{2}-\frac{\log(x)}{2}}}{2x} + \frac{13 e^{x\log(x)-x+\frac{\log(x)^2}{2}-\frac{\log(x)}{2}}}{12x} + O\left(\frac{\log(x)^4 e^{x\log(x)-x+\frac{\log(x)^2}{2}-\frac{\log(x)}{2}}}{x^2}\right)$$