IN PARTNERSHIP WITH:
CNRS

Université Claude Bernard (Lyon 1)

Ecole normale supérieure de Lyon

Activity Report 2012

Team ARIC

Arithmetic and Computing

IN COLLABORATION WITH: Laboratoire de l'Informatique du Parallélisme (LIP)

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Algorithms, Certification, and Cryptography

Table of contents

Team ARIC

Keywords: Computer Arithmetic, Numerical Methods, Hardware Accelerators, Computer Algebra, Floating-point Numbers, Interval Analysis

AriC succeeds to the Arenaire team. It is located in Lyon.

Creation of the Team: January 01, 2012 *, Updated into Project-Team:* January 01, 2013 .

1. Members

Research Scientists

Nicolas Brisebarre [Researcher CNRS]
Laurent-Stéphane Didier [Associate Professor (Univ. Paris 6), sabbatical, until February 2012, HdR]
Claude-Pierre Jeannerod [Researcher Inria]
Vincent Lefèvre [Researcher Inria]
Micaela Mayero [Researcher Inria (on partial secondment, on leave from U. Paris Nord), until August 2012]
Jean-Michel Muller [Senior Researcher CNRS, HdR]
Nathalie Revol [Researcher Inria]
Bruno Salvy [Senior Researcher Inria, since September 2012]
Gilles Villard [Senior Researcher CNRS, HdR]

Faculty Members

Florent de Dinechin [Team leader, Associate Professor ENS de Lyon, HdR]
Nicolas Estibals [ATER ENS de Lyon, since September 2012]
Eleonora Guerrini [ATER ENS de Lyon, until August 2012]
Guillaume Hanrot [Professor ENS de Lyon, HdR]
Fabien Laguillaumie [Professor UCBL, ISFA since September 2012 - Junior Researcher CNRS (on partial secondment, on leave from U. Caen) until August 2012, HdR]
Nicolas Louvet [Associate Professor UCBL]
Ioana Pasca [ATER ENS de Lyon, until August 2012]
Damien Stehlé [Professor ENS de Lyon since September 2012 - Junior Researcher CNRS until August 2012, HdR]

Engineer

Serge Torres [Technical Staff, ENS de Lyon, 40% on the project]

PhD Students

Nicolas Brunie [CIFRE grant (Kalray), 3rd year]
Jingwei Chen [Joint fellowship from CNRS and Chinese Academy of Sciences]
Matei Istoan [ACET engineer since September 2012]
Jingyan Jourdan-Lu [CIFRE grant (STMicroelectronics Compilation Expertise Center, Grenoble), until November 2012]
Adeline Langlois [*Élève Normalienne*, ENS Cachan until August 2012, ENS de Lyon since September 2012, 2nd year]
Érik Martin-Dorel [MESR grant, until September 2012]
Adrien Panhaleux [*Allocataire-moniteur*, ENS grant, until August 2012]
Xavier Pujol [*Allocataire-moniteur*, ENS grant, until August 2012]
Philippe Théveny [MESR grant, 2nd year]

Post-Doctoral Fellows

Rishiraj Bhattacharyya [ENS de Lyon, since June 2012]
Marc Mezzarobba [Inria]

Administrative Assistant

Damien Séon [ENS de Lyon, TR Inria, 50% on the project]

2. Overall Objectives

2.1. Overview

Computer Arithmetic studies how a machine may deal with numbers. This is a wide field with many aspects: from the *mathematics* related to numbers, their representation, and operations on them, to the *technologies* used to build the machine, and through the *algorithms* related to number processing. In addition, there are many different *types* of numbers (mostly integers, reals, complex numbers, and finite fields), many *operations* defined by algebra over these number sets, and many possible *machine representations* of these numbers. Some of these representations are only approximate, which raises *safety* issues. Finally, number processing takes place in the context of *applications* which define constraints or costs that have to be optimized.

The overall objective of AriC is, through computer arithmetic, to improve computing at large, in terms of performance, efficiency, and reliability.

This requires to master the broad range of expertises listed above. The AriC project addresses this challenge in breadth, spanning computer arithmetic along three structural axes:

1. from the high-level specification of a computation to the lower-level details of its implementation,
2. reconciling performance and numerical quality, both when building operators and when using existing operators,
3. developing the mathematical and algorithmic foundations of computing.

More than being research directions themselves, these three axes structure the links between our individual research directions.

This in-breadth approach to computer arithmetic is the very specificity of the AriC project, and its main strength. Other computer arithmetic teams have a much narrower focus (e.g., hardware arithmetic, or floating-point algorithms, or arithmetic for cryptography, or formal proof of computer arithmetic, etc.). Actually, most members of the computer arithmetic community belong to teams that do not focus on computer arithmetic.

With respect to computing at large, our originality is the computer arithmetic focus. We believe that a deep understanding of the arithmetic of machine numbers (taken for what they are, not only as approximate integers or real numbers), is critical to address many challenges of numerics and computing, from reliability (e.g., avoiding overflow or critical loss of precision) to performance.

2.2. Highlights of the Year

Damien Stehlé received the CNRS-INS2I bronze medal.

3. Scientific Foundations

3.1. Applications

Whether its purpose is to design better operators or to make the best use of existing ones, computer arithmetic is strongly connected to applications. Some application domains are particularly in demand for high-quality arithmetic: high-performance computing (HPC) for floating-point, accounting for decimal, digital signal processing (DSP) for fixed-point, embedded systems for application-specific operators, cryptography for finite fields. Each domain comes with its specific constraints and quality metric. For example, cryptography has a specific need of resistance to attacks that impact the design of the operators themselves: a good operator for cryptography should have electromagnetic emissions and power consumption patterns independent of the data it manipulates. Another example is very large-scale HPC, which in some cases is reaching the limits of the accuracy provided by the prevalent double-precision floating-point arithmetic.

The regional (Rhône-Alpes) context is especially strong in embedded systems, with the Minalogic Competitivity Centre, major players such as STMicroelectronics, CEA and Inria, and strong startups such as Kalray. This is also true at the European level, with the HiPEAC European network of excellence. This network addresses hardware issues, but also software and compiler issues.

Indeed, the bridge between the application and the underlying hardware arithmetic is usually the compiler. Therefore, more and more arithmetic expertise should be integrated within the compiler. This goes on par with the current trend to automate arithmetic core generation. In the long term, working at the compiler level opens optimization perspective beyond what compilers traditionally perform, for instance ad-hoc generation and optimization in context of application-specific functional cores.

However, much of computer arithmetic research still focuses on the implementation of standard computing cores (such as elementary functions, linear algebra operators, or DSP filters), although this implementation is more and more automated as illustrated by projects such as ATLAS, Spiral, FFTW, and others.

Cryptography is an active field of research where there is a strong demand for efficient arithmetic operators. Practical schemes such as hash functions, public-key encryption and digital signatures may be used in constrained environments, leading to interesting arithmetic problems. Common examples are long integer arithmetic (RSA) and arithmetic of algebraic curves and finite fields of medium sizes (elliptic curve cryptography, including pairing-based cryptography), and small finite fields (code-based cryptography and lattice-based cryptography).

3.2. Technology

The traditional arithmetic operators are small, low-level, close-to-the-silicon hardware building bricks, and it is therefore important to anticipate the evolutions of the technology to address the new challenges these evolutions will bring.

It is well known now that Moore's law is no longer what it used to be. It continues to bring more transistors on a chip with each new generation, but the speed of these transistors no longer increases, and their power consumption no longer decreases. With more integration come also more reliability issues.

These are the driving forces behind the shift to multicore processors, and to coarser and more complex processing units in these processors: single-instruction, multiple data (SIMD) instructions, fused multiply-and-add, and soon dot-product operations. It also led to the emergence of new massively parallel computing devices such as graphical processing units (GPU) and field-programmable gate arrays (FPGAs). Both are increasingly being used for general purpose computing.

In the shift to massively parallel multicores and GPUs, the real challenge is how to program them. With respect to computer arithmetic, the main problem is the control of numerical precision: the order of the elementary operations is changed in a parallel execution, and will very often not even be deterministic if the main objective is performance. Assessing or guaranteeing numerical quality in the face of this uncertainty is an open problem, all the more as SIMD units and limited data bandwidth encourage the use of mixed precision where possible.

Concerning FPGAs, their programming model is that of a digital circuit which may be application-specific, and even change in the lifetime of an application. The challenge here is to design arithmetic operators that exploit this reconfigurability, which is their main strength. Whereas processor operators have to be as general-purpose as possible, in an FPGA an operator can be designed specifically for a given application's context. A related challenge is to convince application designers that they should use these operators, which may be radically different from those they are used to see in processors. The C-to-hardware community addresses this challenge by hiding the FPGA behind a classical C programming model. This raises the arithmetic problem of automatically extracting from a piece of C code a fragment that is suitable for implementation as an application-specific operator in an FPGA.

In traditional circuit design, power consumption is no longer a concern only for embedded, battery-powered applications: heat dissipation is now the main issue limiting the frequency of high-performance processors. The nature of power consumption is also changing: it used to be caused mostly by the active switching

transistors, but leakage power is now as much of a concern. All this impacts the design of operators, but also their use: the energy-per-computation metric will become more and more important and will orient algorithmic choices, for instance inviting us to reassess the benefits of pre-computing values.

Finally, the industry is preparing to address, within a decade or two, the end of silicon-based Moore's law. In addition to the physical limits (it is believed so far that we need at least one atom to build a transistor), the raising cost of fabrication plants at each generation has led to increasing concentration in fewer and fewer foundries. There will therefore be an economic limit when the number of foundries is down to one. Silicon replacement alternative are emerging in laboratories, without a clear winner yet. When these alternatives reach the integrated circuit, they may be expected to drastically change the rules by which arithmetic operators are designed.

3.3. Numbers and Number Representation

The first issue addressed by computer arithmetic is the representation of numbers in the computer. There are many possible representations, and a representation typically has many parameters. For instance, for integers, the decimal representation and the binary representation belong to the same family, only differing by the radix, 10 or 2. Another parameter of this representation is the number of digits considered.

A good representation is one that enables good computing. Here the measures of quality are numerous, sometimes conflicting, and application-dependent. For instance, the classical representation of integers is compact, but addition involves a carry propagation. There exists another classical family of integer representations which are redundant, therefore less compact, but allow for carry-free, thus faster, addition. Many other quality measures are possible, for instance power consumption, or silicon area.

Research on number representation for integers and reals is no longer very active, and it may be that there is little left to find in this field. The corresponding expertise now belongs to the common culture of the computer arithmetic community. For the integers, from time to time, a new context revives interest in an exotic number representation. For the reals, the indisputable advantages of a widespread and shared standard (the IEEE 754 floating-point standard) weigh strongly against innovation. However, for barely more complex datatypes, such as complex numbers or real intervals (each of which can be represented by a pair of reals), there is no such consensus yet.

Finally, research on number representation is still very active for datatypes related to more recent application fields, most notably in cryptography. For instance, the elliptic curve number system has been introduced because it allowed to use smaller keys for similar security, and research is still active to find representations of elliptic curves that enable efficient computation on this number system. This research tries to improve on the usual quality metrics (performance, resource consumption, power), and in addition we have two more context-specific metrics: the key size, and the security level.

3.4. Arithmetic Algorithms

Each year, new algorithms are still published for basic operations (from addition to division), but the main focus of the computer arithmetic community has long shifted to more complex objects: examples are sums of many numbers, arithmetic on complex numbers, and evaluation of algebraic and transcendental functions.

The latter typically reduces to polynomial evaluation, with two sub-problems: firstly, one must find a good approximation polynomial. Secondly, one must evaluate it as fast as possible under some accuracy constraint.

When looking for good approximation polynomials, "good" has various possible meanings. For arbitrary precision implementations, polynomials must be built at runtime, so "good" means "simple" (for both the polynomial and the error term). Typical techniques in this case are based on Taylor or Chebyshev formulae. For fixed-precision implementations (for instance for the functions of the standard floating-point mathematical library), the polynomial is static, and we may afford to spend much more effort to build it. In this case, we may aim for better polynomials, in the sense that they minimize the approximation error over a complete interval: such polynomials are given by Remez' algorithm [56]. However, the coefficients of Remez polynomials

will be arbitrary reals, and for implementation purpose we are more interested in the class of polynomials with machine-representable coefficients. An even better polynomial is therefore one that minimizes the approximation error among this class, a problem addressed in the Sollya toolbox developed in Arénaire (http://sollya.gforge.inria.fr/). In some cases it is useful to impose even more constraints on the polynomial. For instance, if the function is even, one classically wants to force to zero the coefficients of the odd powers in its polynomial approximation. Although this may require a higher degree approximation for the same accuracy, it reduces operation counts, and also increases the numerical stability of the evaluation.

Then, there are many ways to evaluate a polynomial, corresponding to many ways to rewrite it. The Horner scheme minimizes operation count and, in most practical cases, rounding errors, but it is a sequential scheme entailing a long execution time on modern processors. There exists parallel evaluation schemes that improve this latency, but degrade operation count and accuracy. The optimal scheme depends on details of the target architecture, and is best found by programmed exploration, as demonstrated by Intel on Itanium, and by Arénaire on the ST200 processor.

Thus, both polynomial approximation and polynomial evaluation illustrate the need for "meta-algorithms": i.e., algorithms designed to build arithmetic algorithms. In our example, the meta-algorithms in turn rely on linear algebra, integer linear programming, and Euclidean lattices. Other approaches may also lead to successful meta-algorithms, for instance the SPIRAL project (http://www.spiral.net/) uses algebraic rewriting to implement and optimize linear transforms. This approach has potential in arithmetic design, too.

3.5. Euclidean Lattice Reduction and Applications

A Euclidean lattice is the set of *integer* linear combinations of a finite set of real vectors. Typically, lattices occur when linear algebra questions are asked with discreteness constraints. In the last decade, they have become a classical ingredient in the computer arithmetic toolbox, along with other number-theoretic techniques (continued fractions, diophantine approximation, etc.). Indeed, integers (scaled by powers of the radix) are the essence of the fixed-point and floating-point representations of the real numbers. If the macroscopic properties of floating-point numbers are close to those of the real numbers, the finer properties are definitely related to questions over the integers. Thus, lattices have been successfully used in computer arithmetic to find constrained polynomial approximations to functions, and to attack the Table Maker's Dilemma. They have a potential for further arithmetic applications, for instance the design of digital filters.

Besides, the algorithms on Euclidean lattices are a rich experimentation laboratory for different types of arithmetics. The basis vectors are often represented exactly with long integer arithmetic. Furthermore, the fastest algorithms find the operations to be performed on the basis vectors via approximate computations, typically an approximate Gram-Schmidt orthogonalisation. These approximate computations may be performed with fixed-precision or arbitrary precision floating-point arithmetics. In some time-consuming applications of lattice algorithms, such as cryptanalyses of variants of RSA or lattice-based cryptosystems, integer linear programming, or even for solving the Table Maker's Dilemma, the practical run-time is of utmost importance. This motivates strong optimizations for the underlying arithmetics.

Further, aside from this strong relationship between lattices and arithmetics, the understanding of lattice-based cryptology is developing at a quick pace; making it efficient while remaining secure will require a thorough study, which must involve experts in both arithmetics and cryptology.

3.6. Reliability and Accuracy

Having basic arithmetic operators that are well-specified by standards leads to two directions. The first is to provide a guarantee that the implementations of these operators match their specification. The second is to use these operators as building blocks of well-specified computations, in other words to build upon these operators to obtain guarantees on the results of larger computing cores.

The approaches used to get such a guarantee vary greatly. Some computations are performed exactly, and in this case the results are considered to be intrinsically correct. However, exact values may not be finitely representable in the chosen number system and format: they must then be approximated. When an approximate value is computed using floating-point arithmetic, the specification of this arithmetic is employed to establish a bound on the roundoff errors, or to check that no exceptional situation occurred. For instance, the IEEE-754 standard for floating-point arithmetic implies useful properties, e.g., Dekker's error-free multiplication for various radices and precisions, the faithfulness of Horner's polynomial evaluation, etc.

Another possibility is that a simple final computation, still performed using floating-point arithmetic, enables to check whether a computed result is a reasonable approximation of the exact (unknown) result. Typically, to check that, for instance, a computed matrix R is close to the inverse of the initial matrix A, it suffices to check whether the product RA is close enough to the identity matrix. Such a simple, a posteriori, computation is called a *certificate*.

When considering more complicated functions, e.g., elementary functions, another issue arises. These functions have to be approximated, in general by polynomials. It no longer suffices to bound the rounding errors of the computations and check that no underflow/overflow may occur. One also has to take into account the approximation errors: certifying tight error bounds is quite a challenge. One usually talks of *verified computations* in this case.

Safety is typically based on interval arithmetic: what is computed is an interval which provably encloses the sought values. Naive interval arithmetic evaluates an expression as it is written, which does not take into account the dependencies between variables. This leads to irrelevant interval bloat. To address this problem, a solution is sometimes to rewrite the expression, a technique used for instance by the Gappa tool (http://gappa.gforge.inria.fr/) initially developed in Arénaire. Another systematic method is to use extensions to interval arithmetic. For instance, affine arithmetic has been used to optimize the data-path width of FPGA computing cores, and is also used in the Fluctuat tool to diagnose numerical instabilities in programs. When working with functions, Taylor models are a relevant extension: they represent a function as the sum of a polynomial of fixed degree and of an interval enclosing all errors (approximation as well rounding errors). This approach is very useful for computations involving function approximations, and has for instance been used successfully for the computation of the supremum norm of a function in one variable. The issue here is to devise algorithms that do not overestimate too much the result. It may be necessary to mix interval arithmetic and variable precision to reach the required level of guarantee and accuracy. In general, determining the right precision is difficult: the precision must be high enough to yield accurate results, but not too high since the computing time increases with the computing precision.

The complexity of some computer arithmetic algorithms, the intrinsic complexity of the floating-point model, the use of floating-point for critical applications, strongly advocate for the use of *formal proof* in computer arithmetic: a proof checker checks every step of the proof obtained by any means mentioned above. Even circuit manufacturers often provide a formal proof of the critical parts of their floating-point algorithms. For instance, the Intel divide and square root algorithms for the Itanium were formally proven. The expertise of the French community (which includes several ex-Arénaire members) in proving floating-point algorithms is well recognized. However, even the lower properties of the arithmetic are still challenging. For instance, with the specification of decimal arithmetic in the new version of the IEEE 754 standard, many theorems established in radix two have to be generalized to other radices.

4. Application Domains

4.1. Hardware Arithmetic

The application domains of hardware arithmetic operators are digital signal processing, image processing, embedded applications, reconfigurable computing, and cryptography.

4.2. Floating-point and Validated Numerics

Our expertise on validated numerics is useful to analyse and improve, and guarantee the quality of numerical results in a wide range of applications, from scientific simulation to global optimization or control theory. Much of our work, in particular the development of correctly rounded elementary functions, is critical to the reproducibility of floating-point computations.

4.3. Cryptography, Cryptology, Communication Theory

Lattice reduction algorithms have direct applications in public-key cryptography. A new and promising field of applications is communications theory.

5. Software

5.1. Overview

AriC software and hardware realizations are accessible from the web page http://www.ens-lyon.fr/LIP/AriC/ware.html. We describe below only those which progressed in 2012.

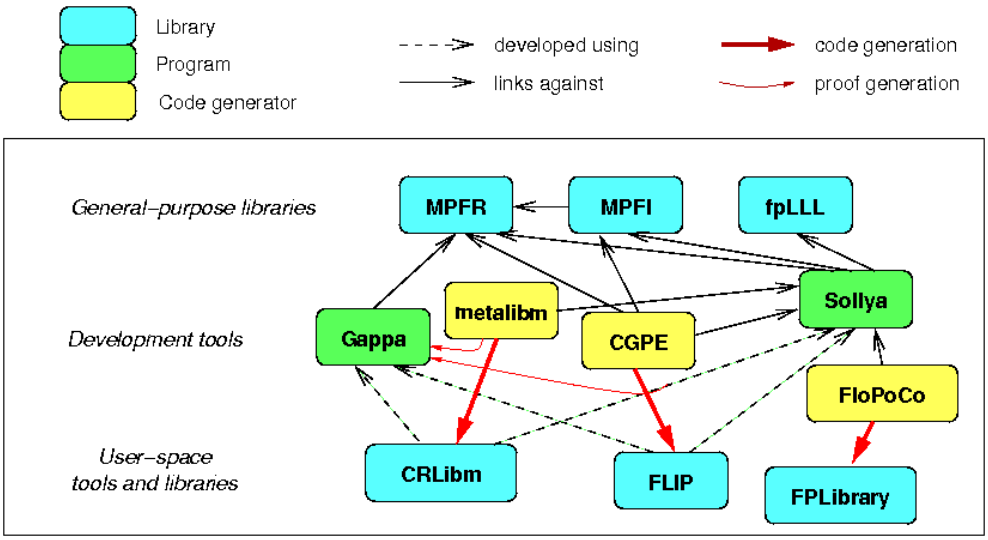

Figure 1. Relationships between some AriC developments.

5.2. FloPoCo

Participants: Florent de Dinechin [correspondant], Matei Istoan.

The purpose of the FloPoCo project is to explore the many ways in which the flexibility of the FPGA target can be exploited in the arithmetic realm. FloPoCo is a generator of operators written in C++ and outputting synthesizable VHDL automatically pipelined to an arbitrary frequency.

In 2012, the diverging multiplier implementations in FloPoCo were unified using a common *bit-heap* framework. In addition, several new operators were added.

FloPoCo also now offers state-of-the-art random generators written by David Thomas at Imperial College.

Versions 2.3.1 and 2.4.0 were released in 2012.

Among the known users of FloPoCo are U. Cape Town, U.T. Cluj-Napoca, Imperial College, U. Essex, U. Madrid, U. P. Milano, T.U. Muenchen, T. U. Kaiserslautern, U. Paderborn, CalTech, U. Pernambuco, U. Perpignan, U. Tokyo, Virginia Tech U. and several companies.

URL: http://flopoco.gforge.inria.fr/

- Version: 2.3.0 (december 2011)
- APP: IDDN.FR.001.400014.000.S.C.2010.000.20600 (version 2.0.0)
- License: specific, GPL-like.
- Type of human computer interaction: command-line interface, synthesizable VHDL output.
- OS/Middleware: Linux, Windows/Cygwin.
- Required library or software: MPFR, flex, Sollya.
- Programming language: C++.
- Documentation: online and command-line help, API in doxygen format, articles.

5.3. GNU MPFR

Participants: Vincent Lefèvre [correspondant], Paul Zimmermann [Caramel, Inria Nancy - Grand Est].

GNU MPFR is an efficient multiple-precision floating-point library with well-defined semantics (copying the good ideas from the IEEE-754 standard), in particular correct rounding in 5 rounding modes. GNU MPFR provides about 80 mathematical functions, in addition to utility functions (assignments, conversions...). Special data (*Not a Number*, infinities, signed zeros) are handled like in the IEEE-754 standard.

MPFR was one of the main pieces of software developed by the old SPACES team at Loria. Since late 2006, with the departure of Vincent Lefèvre to Lyon, it has become a joint project between the Caramel (formerly SPACES then CACAO) and the AriC (formerly Arénaire) project-teams. MPFR has been a GNU package since 26 January 2009.

An MPFR-MPC developers meeting took place from 25 to 27 June 2012 in Bordeaux. GNU MPFR 3.1.1 was released on 3 July 2012.

The main changes done in the AriC project-team for the future versions are tcc support, more automation for the releases, new functions to operate on groups of flags, and bug fixes.

URL: http://www.mpfr.org/

GNU MPFR is now on the Ohloh community platform for free and open source software: https://www.ohloh.net/p/gnu-mpfr

- ACM: D.2.2 (Software libraries), G.1.0 (Multiple precision arithmetic), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- APP: no longer applicable (copyright transferred to the Free Software Foundation).
- License: LGPL version 3 or later.
- Type of human computer interaction: C library, callable from C or other languages via third-party interfaces.
- OS/Middleware: any OS, as long as a C compiler is available.
- Required library or software: GMP.
- Programming language: C.
- Documentation: API in texinfo format (and other formats via conversion); algorithms are also described in a separate document.

5.4. Exhaustive Tests for the Correct Rounding of Mathematical Functions

Participant: Vincent Lefèvre.

The search for the worst cases for the correct rounding (hardest-to-round cases) of mathematical functions (exp, log, sin, cos, etc.) in a fixed precision (mainly double precision) using Lefèvre's algorithm is implemented by a set of utilities written in Perl, with calls to Maple/intpakX for computations on intervals and with C code generation for fast computations. It also includes a client-server system for the distribution of intervals to be tested and for tracking the status of intervals (fully tested, being tested, aborted).

The Perl scripts have been improved to detect various errors from Maple and in particular, restart Maple automatically when the license server is not reachable.

5.5. FLIP: Floating-point Library for Integer Processors

Participants: Claude-Pierre Jeannerod [correspondant], Jingyan Jourdan-Lu.

FLIP is a C library for the efficient software support of binary32 IEEE 754-2008 floating-point arithmetic on processors without floating-point hardware units, such as VLIW or DSP processors for embedded applications. The current target architecture is the VLIW ST200 family from STMicroelectronics (especially the ST231 cores). This year, we have extended the DP2 operator (fused dot product in dimension two) and its specializations, initially designed for rounding to nearest, to directed rounding modes. We have also worked on the implementation of the simultaneous computation of sine and cosine, with proven 1-ulp accuracy and in the same latency as the evaluation of sine alone.

URL: http://flip.gforge.inria.fr/

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- APP: IDDN.FR.001.230018.S.A.2010.000.10000
- License: CeCILL v2
- Type of human computer interaction: C library callable, from any C program.
- OS/Middleware: any, as long as a C compiler is available.
- Required library or software: none.
- Programming language: C

5.6. FPLLL: A Lattice Reduction Library

Participants: Xavier Pujol, Damien Stehlé [correspondant].

fplll contains several algorithms on lattices that rely on floating-point computations. This includes implementations of the floating-point LLL reduction algorithm, offering different speed/guarantees ratios. It contains a "wrapper" choosing the estimated best sequence of variants in order to provide a guaranteed output as fast as possible. In the case of the wrapper, the succession of variants is oblivious to the user. It also includes a rigorous floating-point implementation of the Kannan-Fincke-Pohst algorithm that finds a shortest non-zero lattice vector, and the BKZ reduction algorithm.

The fplll library is used or has been adapted to be integrated within several mathematical computation systems such as Magma, Sage and PariGP. It is also used for cryptanalytic purposes, to test the resistance of cryptographic primitives.

Versions 4.0.0 and 4.0.1 were released in 2012, implementing the BKZ reduction algorithm.

URL: http://xpujol.net/fplll/

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- APP: Procedure started
- License: LGPL v2.1
- Type of human computer interaction: C++ library callable, from any C++ program.
- OS/Middleware: any, as long as a C++ compiler is available.
- Required library or software: MPFR and GMP.
- Programming language: C++.
- Documentation: available in html format on **URL:** http://xpujol.net/fplll/fplll-doc.html

5.7. Symbolic-numeric Computations with Linear ODEs

Participant: Marc Mezzarobba.

NumGfun is a Maple package for performing numerical and "analytic" computations with the solutions of linear ordinary differential equations with polynomial coefficients. Its main features include the numerical evaluation of these functions with rigorous error bounds and the computation of symbolic bounds on solutions of certain recurrences. NumGfun is distributed as part of gfun, itself part of the Algolib bundle. It is used by the Dynamic Dictionary of Mathematical Functions to provide its numerical evaluation features. NumGfun 0.6, released in 2012, provides new feature for the numerical solution of so-called regular singular connection problems, and many small improvements.

URL: http://marc.mezzarobba.net/#code-NumGfun

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- APP: cf. gfun
- License: LGPL v2.1
- Type of human computer interaction: Maple library, usable interactively or from Maple code.
- OS/Middleware: any platform supporting Maple.
- Required library or software: Maple, gfun.
- Programming language: Maple
- Documentation: available as Maple help pages and in pdf format.

5.8. SIPE: Small Integer Plus Exponent

Participant: Vincent Lefèvre.

SIPE (Small Integer Plus Exponent) is a mini-library in the form of a C header file, to perform computations in very low precisions with correct rounding to nearest in radix 2. The goal of such a tool is to do proofs of algorithms/properties or computations of tight error bounds in these precisions by exhaustive tests, in order to try to generalize them to higher precisions. The currently supported operations are the addition, subtraction, multiplication, FMA, minimum/maximum/comparison functions (of the signed numbers or in magnitude), and conversions.

A new macro `SIPE_2MUL`, returning the rounded result and the error of a multiplication, has been added.

A test program and scripts to perform timing comparisons with hardware IEEE-754 floating-point and with GNU MPFR are available, together with a discussion on the technical and algorithmic choices behind SIPE and timing results. [39]

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- License: LGPL version 2.1 or later.
- Type of human computer interaction: C header file.
- OS/Middleware: any OS.
- Required library or software: GCC compiler.
- Programming language: C.
- Documentation: Research report Inria RR-7832.
- URL: http://www.vinc17.net/software/sipe.h

6. New Results

6.1. Applications

Florent de Dinechin contributed high-performance signal processing on an FPGA to a prototype of high-throughput receiver for optical fiber transmission developed by Alcatel [33]. He also wrote a book chapter exposing the potential of FPGA-specific arithmetic for high-performance computing [49].

6.2. Hardware and FPGA Arithmetic

6.2.1. Mixed-precision fused multiply-and-add

With B. de Dinechin, from Kalray, N. Brunie and F. de Dinechin proposed to extend the classical fused-multiply-and-add operator with a larger addend and result. This enables higher-precision computation of sums of products at a cost that remains close to that of the classical FMA [29].

6.2.2. Multiplication by rational constants versus division by a constant

Motivated by the division by 3 or by 9 appearing in some stencil kernels, F. de Dinechin investigated how the periodicity of the binary representation of a rational constant could be exploited to design an architecture multiplying by this constant [18]. With L. S. Didier, this approach was then compared to a specialisation of divider architectures to the division by small integer constants, which is shown to match well the fine structure of FPGAs [32].

6.2.3. Floating-point exponentiation on FPGA

F. de Dinechin, with P. Echeverria and M. Lopez-Vallejo (U. Madrid) and B. Pasca (Altera), implemented the first floating-point unit for the pow and powr functions of the IEEE-754-2008 standard [50]. These functions compute x^y, and differ only in the specification of special cases. The implementation, parameterized in exponent and significand size, combines suitably modified exponential and logarithm units.

6.2.4. Arithmetic around the bit heap

F. de Dinechin, M. Istoan, G. Sergent, K. Illyes, B. Popa, and N. Brunie extended FloPoCo with a versatile framework for manipulating sums of weighted bits [51], [44]. This is a relevant way of implementing polynomials, filters and other coarse arithmetic cores.

6.2.5. Improving computing architectures

To improve High-Level Synthesis (HLS) for FPGAs, B. Pasca (former PhD student in AriC), with Ch. Alias (Inria Compsys) and A. Plesco (Zettice) developed tiling and scheduling algorithms that exploit the deeply pipelined operator at the core of a computing kernel [14].

With S. Collange and G. Diamos, N. Brunie proposed improvements in the architecture of general-purpose graphical processing units [28].

N. Brunie and F. de Dinechin, with Kalray's B. de Dinechin, are investigating embedding a reconfigurable core in the Kalray MPPA architecture. For this purpose, N. Brunie developed an environment for the design exploration of such an accelerator. This environment produces the hardware on one side, and its programming tools on the other side [43].

6.3. Elementary Functions

6.3.1. (M,p,k)-friendly points: a table-based method for trigonometric function evaluation

N. Brisebarre, M. Ercegovac (U. California at Los Angeles) and J.-M. Muller [25] present a new way of approximating the sine and cosine functions by a few table look-ups and additions. It consists in first reducing the input range to a very small interval by using rotations with "(M, p, k) friendly angles", proposed in this work, and then by using a bipartite table method in a small interval. An implementation of the method for 24-bit case is described and compared with CORDIC. Roughly, the proposed scheme offers a speedup of 2 compared with an unfolded double-rotation radix-2 CORDIC.

6.3.2. On Ziv's rounding test

With Ch. Lauter (LIP6), F. de Dinechin, J.-M. Muller and S. Torres proved and generalized a code sequence due to Ziv, which is used to round correctly a real value approximated (with a known error bound) as the unevaluated sum of two floating-point numbers [52].

6.4. Arithmetic Algorithms

6.4.1. Binary floating-point operators for VLIW integer processors

C.-P. Jeannerod and J. Jourdan-Lu [35] proposed software implementations of sinf, cosf and sincosf over [-pi/4, pi/4] that have proven 1-ulp accuracy and whose respective latencies on STMicroelectronics' ST231 VLIW integer processor are 19, 18 and 19 cycles. To get such performances they introduced a novel algorithm for simultaneous sine and cosine that combines univariate and bivariate polynomial evaluation schemes.

In the same context, C.-P. Jeannerod, J. Jourdan-Lu and C. Monat (STMicroelectronics Compilation Expertise Center, Grenoble) [36] studied the implementation of *custom* (i.e., specialized, fused, or simultaneous) operators, and provided qualitative evidence of the benefits of supporting such operators in addition to the five basic ones: this allows to be up to 4.2x faster on individual calls, and up to 1.59x faster on DSP kernels and benchmarks.

6.4.2. Error bounds for complex floating-point division with an FMA

Assuming that a fused multiply-add (FMA) instruction is available, C.-P. Jeannerod, N. Louvet and J.-M. Muller [37] obtained sharp error bounds for various alternatives to Kahan's 2 by 2 determinant algorithm. Combining such alternatives with Kahan's original scheme leads to componentwise-accurate algorithms for complex floating-point division, and for these algorithms sharp or reasonably sharp error bounds were also obtained.

6.4.3. Computation of correctly-rounded sums

P. Kornerup (U. of Southern Denmark), V. Lefèvre and J.-M. Muller [19] have shown that among the set of the algorithms with no comparisons performing only floating-point additions/subtractions, the 2Sum algorithm introduced by Knuth is minimal, both in terms of number of operations and depth of the dependency graph. They also prove that under reasonable conditions, an algorithm performing only round-to-nearest additions/subtractions cannot compute the round-to-nearest sum of at least three floating-point numbers. They also present new results about the computation of the correctly-rounded sum of three floating-point numbers.

6.4.4. Comparison between binary64 and decimal64 floating-point numbers

N. Brisebarre, C. Lauter (U. Paris 6), M. Mezzarobba and J.-M. Muller [27] introduce an algorithm that allows one to quickly compare a binary64 floating-point (FP) number and a decimal64 FP number, assuming the "binary encoding" of the decimal formats specified by the IEEE 754-2008 standard for FP arithmetic is used. It is a two-step algorithm: a first pass, based on the exponents only, makes it possible to quickly eliminate most cases, then when the first pass does not suffice, a more accurate second pass is required. They provide an implementation of several variants of their algorithm, and compare them.

6.5. Computer Algebra

6.5.1. Faster multivariate interpolation with multiplicities

M. Chowdhury (U. Western Ontario), C.-P. Jeannerod, V. Neiger (ENS de Lyon), É. Schost (U. Western Ontario) and G. Villard proposed fast randomized algorithms for interpolating multivariate polynomials with multiplicities. In the special bivariate case, this allows to accelerate the interpolation step of Guruswami and Sudan's list-decoding by a factor (list size)/(multiplicity).

6.5.2. On the complexity of solving quadratic boolean systems

M. Bardet (U. Rouen), J.-Ch. Faugère (PolSys), B. Salvy, and P.-J. Spaenlehauer (PolSys) [16] dealt with the fundamental problem in computer science of finding all the common zeroes of polynomials systems of quadratic polynomials over the field with 2 elements. The cryptanalysis of several modern ciphers reduces to this problem. Up to now, the best complexity bound was reached by an exhaustive search. They gave an algorithm that reduces the problem to a combination of exhaustive search and sparse linear algebra. This algorithm has several variants depending on the method used for the linear algebra step. Under precise algebraic assumptions, their complexity breaks the 2^n barrier. Experiments on random systems show that the algebraic assumptions are satisfied with probability very close to 1.

6.5.3. Power series solutions of singular (q)-differential equations

A. Bostan (Algorithms), M. F. I. Chowdhury (U. Western Ontario), R. Lebreton (Lix), B. Salvy, and É. Schost (U. Western Ontario) provided in [23] algorithms computing power series solutions of a large class of differential or q-differential equations or systems. Their number of arithmetic operations grows linearly with the precision, up to logarithmic terms.

6.5.4. Fast computation of common left multiples of linear ordinary differential operators

A. Bostan (Algorithms), F. Chyzak (Algorithms), Ziming Li (Chinese Academy of Sciences), and B. Salvy studied in [24] tight bounds and fast algorithms for LCLMs of several linear differential operators with polynomial coefficients. They analyzed the arithmetic complexity of existing algorithms for LCLMs, as well as the size of their outputs. They proposed a new algorithm that recasts the LCLM computation in a linear algebra problem on a polynomial matrix. This algorithm yields sharp bounds on the coefficient degrees of the LCLM, improving by one order of magnitude the best bounds obtained using previous algorithms. The complexity of the new algorithm is almost optimal, in the sense that it nearly matches the arithmetic size of the output.

6.5.5. Space complexity of fast D-finite function evaluation

M. Mezzarobba [41] showed that D-finite functions, i.e., solutions of linear differential equations with polynomial coefficients, can be evaluated in quasi-linear time and linear space with respect to the precision. In comparison, existing fast algorithms due to Chudnovsky and Chudnovsky and to van der Hoeven achieved the same time complexity with an overhead of a logarithmic factor in terms of memory usage.

6.5.6. Multiple precision evaluation of the Airy function with reduced cancellation

The series expansion at the origin of the Airy function $\mathrm{Ai}(x)$ is alternating and hence problematic to evaluate for $x > 0$ due to cancellation. Based on a method recently proposed by Gawronski, Müller, and Reinhard, Sylvain Chevillard and Marc Mezzarobba [31] exhibit two functions F and G, both with nonnegative Taylor expansions at the origin, such that $\mathrm{Ai}(x) = G(x)/F(x)$. The sums are now well-conditioned, but the Taylor coefficients of G turn out to obey an ill-conditioned three-term recurrence. They use the classical Miller algorithm to overcome this issue. They bound all errors and their implementation allows an arbitrary and certified accuracy, that can be used, e.g., for providing correct rounding in arbitrary precision.

6.5.7. Algorithms for combinatorial structures: well-founded systems and Newton iterations

C. Pivoteau (U. Marne-la-Vallée), B. Salvy, and M. Soria (UPMC) [21] considered systems of recursively defined combinatorial structures. They gave algorithms checking that these systems are well founded, computing generating series and providing numerical values. Their framework is an articulation of the constructible classes of Flajolet and Sedgewick with Joyal's species theory. They extend the implicit species theorem to structures of size zero. A quadratic iterative Newton method was shown to solve well-founded systems combinatorially. From there, truncations of the corresponding generating series were obtained in quasi-optimal complexity. This iteration transfers to a numerical scheme that converges unconditionally to the values of the generating series inside their disk of convergence. These results provide important subroutines in random generation. Finally, the approach was extended to combinatorial differential systems.

6.6. Euclidean Lattice Reduction and Applications

6.6.1. Lattice algorithms and hardness proofs

X.-W. Chang (McGill), D. Stehlé and G. Villard [17] proposed the first fully rigorous perturbation analysis of the R-factor of LLL-reduced matrices under column-wise perturbations. This study is very useful to devise LLL-type algorithms relying on floating-point approximations.

L. Luzzi (ENSEA), C. Ling (Imperial College) and D. Stehlé improved [20] the analyses of efficient Bounded Distance Decoding algorithms for lattices, and investigated the consequences for lattice-coded multiple-input multiple-output (MIMO) systems.

A. Langlois and D. Stehlé [54] introduced the Module-SIS and Module-LWE average-case lattice problems and reduced worst-case lattice problems to them. This provides a progressive transformation from the non-structured average-case lattices problems SIS and LWE, to the quite restricted but efficient average-case lattices problems Ring-SIS and Ring-LWE.

6.6.2. Cryptography

S. Ling (Nanyang Technological University, Singapore) and D. Stehlé [55] described the first public-key traitor tracing encryption scheme with security relying on the hardness of standard worst-case problems on Euclidean lattices.

J.-C. Belfiore (Telecom Paritech), L. Luzzi (ENSEA), C. Ling (Imperial College) and D. Stehlé [53] proved that nested lattice codes can achieve semantic security and strong secrecy over the Gaussian wiretap channel.

S. Ling (Nanyang Technological University, Singapore), K. Nguyen (NTU), H. Wang (NTU) and D. Stehlé [40] generalized Stern's zero-knowledge proof of knowledge protocol to obtain a statistical zero-knowledge proof of knowledge for the Inhomogeneous Small Integer Solution ISIS problem (in the infinity norm). This scheme is the first one that comes with no norm loss in the knowledge extraction procedure, leading to cryptographic constructions with tighter security proofs.

N. Attrapadung (AIST, Japan), J. Herranz (UPC, Spain), F. Laguillaumie, B. Libert (UCL, Belgium), E. de Panafieu (ENS Cachan), C. Ràfols (UPC, Spain) [15] proposed the first attribute-based encryption (ABE) schemes allowing for truly expressive access structures and with constant ciphertext size.

G. Castagnos (IMB) and F. Laguillaumie [38] gave a generic approach to design homomorphic encryption schemes, which extends Gjosteen's framework. A specific scheme allows an arbitrary number of multiplications in the groups, as well as a pairing evaluation on the underlying plaintexts.

J. Herranz (UPC, Spain), F. Laguillaumie, B. Libert (UCL, Belgium) and C. Ràfols (URV, Catalonia) [34] proposed the first two attribute-based (for threshold predicates) signature schemes with constant size signatures. Their security is proven in the selective-predicate and adaptive-message setting, in the standard model, under chosen message attacks.

S. Canard (Orange Labs), G. Fuchsbauer (University of Bristol, UK), A. Gouget (Gemalto), F. Laguillaumie [30] defined a new cryptographic primitive called plaintext-checkable encryption, which extends public-key encryption by the following functionality: given a plaintext, a ciphertext and a public key, it is universally possible to check whether the ciphertext encrypts the plaintext under the key. They provide efficient generic random-oracle constructions based on any probabilistic or deterministic encryption scheme as well as a practical construction in the standard model.

6.7. Reliability and Accuracy

6.7.1. Standardization of interval arithmetic

We contributed to the creation in 2008 and N. Revol chairs the IEEE 1788 working group on the standardization of interval arithmetic http://grouper.ieee.org/groups/1788/. More than 140 persons from over 20 countries take part in the discussions, around 1500 messages were exchanged in 2012. We are currently voting on portions of the text of the standard and have good hope that the group will reach a final version of the standard within the allotted time. An extension has been granted for 2 more years, until December 2014.

The annual in-person meeting, chaired by N. Revol, took place at the end of the SCAN 2012 conference in Novosibirsk, Russia, the 28th of September. It was broadcasted via the Web and feedback was possible through e-mails. More than 20 persons attended the meeting.

V. Lefèvre participated in various discussions, either in the mailing-list or in small subgroups (he sent around 390 mail messages in 2012). He proposed a motion, which passed, on properties needed by number formats for operations between intervals and numbers (constructors, midpoint, etc.).

The latest discussions dealt with:

- flavors: even if there continues to be a give-and-take between proponents of a "small" standard involving just basic interval arithmetic and those who also want to also include the less common "modal arithmetic", this motion about "flavors" intends to allow inclusion of modal interval arithmetic consistently and simply, possibly at a later stage or revision of the standard;
- expressions: what is regarded as an expression by P1788, the relation with the programming languages, what this implies concerning the allowed optimizations, etc.;
- decorations: what are the properties of functions we want to track along a computation, how the empty interval is handled, etc.;
- reproducibility: across several runs of a translated (e.g., compiled) program or across platforms, representation-independent behavior, reproducibility for parallel programs, etc.

A personal view of the current status of the work of the IEEE P1788 group and of directions for future work has been presented in [46], [45].

6.7.2. Interval matrix multiplication

Several formulas exist for the product of two intervals using the midpoint-radius representation: they trade off accuracy for efficiency. The use of these formulas for the product of matrices with interval coefficients allows to use BLAS3 routines and to benefit from their performances in terms of execution time [48]. The accuracy of these methods are studied in [42]. As it can be difficult to ensure that a prescribed rounding mode is actually in use, formulas that are oblivious to the rounding mode are developed [22]. The implementations of these variants on multicores are compared in [47].

6.7.3. Rigorous polynomial approximation using Taylor models in Coq

One of the most common and practical ways of representing a real function on machines is by using a polynomial approximation. It is then important to properly handle the error introduced by such an approximation. N. Brisebarre, M. Joldes (Uppsala Univ., Sweden), E. Martin-Dorel, M. Mayero, J.-M. Muller, I. Pasca, L. Rideau (Marelle), and L. Théry (Marelle) have worked on the problem of offering guaranteed error bounds for a specific kind of rigorous polynomial approximation called Taylor model [26]. They carry out this work in the Coq proof assistant, with a special focus on genericity and efficiency for our implementation. They give an abstract interface for rigorous polynomial approximations, parameterized by the type of coefficients and the implementation of polynomials, and they instantiate this interface to the case of Taylor models with interval coefficients, while providing all the machinery for computing them.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

7.1.1. STMicroelectronics CIFRE PhD Grant

Jingyan Jourdan-Lu was supported by a CIFRE PhD grant (from March 2009 to September 2012) from STMicroelectronics (Compilation Expertise Center, Grenoble) on the theme of floating-point arithmetic code generation and specialization for embedded processors. Advisors: Claude-Pierre Jeannerod and Jean-Michel Muller (AriC), Christophe Monat (STMicroelectronics). A contract between STMicroelectronics and Inria (duration: 36 months; amount: 36,000 euros; signature: fall 2010) aimed at supporting the developments done in the context of this PhD, defended 2012/11/15.

7.1.2. Kalray CIFRE PhD Grant

Nicolas Brunie is supported by a CIFRE PhD grant (from 15/04/2011 to 14/04/2014) from Kalray. Its purpose is the study of a tightly-coupled reconfigurable accelerator to be embedded in the Kalray multicore processor. Advisors: Florent de Dinechin (Arénaire) and B. de Dinechin (Kalray). The support contract between Kalray and Inria amounts to 76,000 euros on three years.

7.1.3. Intel Donation

Intel is making a donation of 20,000$ to AriC to support research around the automatic construction of libm functions.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR HPAC Project

Participants: Claude-Pierre Jeannerod, Nicolas Louvet, Nathalie Revol, Damien Stehlé, Philippe Théveny, Gilles Villard.

"High-performance Algebraic Computing" (HPAC) is a four year ANR project that started in January 2012. The Web page of the project is http://hpac.gforge.inria.fr/. HPAC is headed by Jean-Guillaume Dumas (CASYS team, LJK laboratory, Grenoble); it involves AriC as well as the Inria project-team MOAIS (LIG, Grenoble), the Inria project-team PolSys (LIP6 lab., Paris), the ARITH group (LIRMM laboratory, Montpellier), and the HPC Project company.

The overall ambition of HPAC is to provide international reference high-performance libraries for exact linear algebra and algebraic systems on multi-processor architecture and to influence parallel programming approaches for algebraic computing. The central goal is to extend the efficiency of the LinBox and FGb libraries to new trend parallel architectures such as clusters of multi-processor systems and graphics processing units in order to tackle a broader class of problems in lattice cryptography and algebraic cryptanalysis. HPAC conducts researches along three axes:
- A domain specific parallel language (DSL) adapted to high-performance algebraic computations;
- Parallel linear algebra kernels and higher-level mathematical algorithms and library modules;
- Library composition and innovative high performance solutions for cryptology challenges.

8.1.2. ANR TaMaDi Project

Participants: Nicolas Brisebarre, Florent de Dinechin, Guillaume Hanrot, Vincent Lefèvre, Érik Martin-Dorel, Micaela Mayero, Jean-Michel Muller, Ioana Pasca, Damien Stehlé, Serge Torres.

The TaMaDi project (Table Maker's Dilemma, 2010-2013) is funded by the ANR and headed by Jean-Michel Muller. It was submitted in January 2010, accepted in June, and started in October 2010. The other French teams involved in the project are the MARELLE team-project of Inria Sophia Antipolis-Méditerranée, and the PEQUAN team of LIP6 lab., Paris.

The aim of the project is to find "hardest to round" (HR) cases for the most common functions and floating-point formats. In floating-point (FP) arithmetic having fully-specified "atomic" operations is a key-requirement for portable, predictable and provable numerical software. Since 1985, the four arithmetic operations and the square root are IEEE specified (it is required that they should be correctly rounded: the system must always return the floating-point number nearest the exact result of the operation). This is not fully the case for the basic mathematical functions (sine, cosine, exponential, etc.). Indeed, the same function, on the same argument value, with the same format, may return significantly different results depending on the environment. As a consequence, numerical programs using these functions suffer from various problems. The lack of specification is due to a problem called the Table Maker's Dilemma (TMD). To compute $f(x)$ in a given format, where x is a FP number, we must first compute an approximation to $f(x)$ with a given precision, which we round to the nearest FP number in the considered format. The problem is the following: finding what the accuracy of the approximation must be to ensure that the obtained result is always equal to the "exact" $f(x)$ rounded to the nearest FP number. In the last years, our team-project and the CACAO team-project of Inria Nancy-Grand Est designed algorithms for finding hardest-to-round cases. These algorithms do not allow to tackle with large formats. The TaMaDi project mainly focuses on three aspects:

- big precisions: we must get new algorithms for dealing with precisions larger than double precision. Such precisions will become more and more important (even if double precision may be thought as more than enough for a final result, it may not be sufficient for the intermediate results of long or critical calculations);
- formal proof: we must provide formal proofs of the critical parts of our methods. Another possibility is to have our programs generating certificates that show the validity of their results. We should then focus on proving the certificates;
- aggressive computing: the methods we have designed for generating HR points in double precision require weeks of computation on hundreds of PCs. Even if we design faster algorithms, we must massively parallelize our methods, and study various ways of doing that.

The various documents can be found at http://tamadiwiki.ens-lyon.fr/tamadiwiki/index.php/Main_Page.

8.2. International Initiatives

8.2.1. Inria Associate Teams

QOLAPS (Quantifier elimination, Optimization, Linear Algebra and Polynomial Systems) Associate Team between the Symbolic Computation Group at North Carolina State University (USA), the PolSys team at LIP6, Paris 6, and the AriC team. Participants: Nathalie Revol and Gilles Villard.

8.2.2. Participation in International Programs

Joint CNRS-Royal Society grant with Cong Ling (Imperial College, London). Participants: Guillaume Hanrot and Damien Stehlé.

CNRS Associate Team (PICS) with the Cryptography groups of Macquarie University (Christophe Doche and Igor Shparlinski) and Monash University (Ron Steinfeld). Participants: Nicolas Brisebarre, Guillaume Hanrot, Fabien Laguillaumie, Adeline Langlois and Damien Stehlé.

Merlion grant, co-funded by the French Embassy in Singapore and NTU (Nanyang Technological University), with the cryptography group of NTU (San Ling, Khoa Nguyen and Huaxiong Wang). Participants: Adeline Langlois and Damien Stehlé.

8.3. International Research Visitors

8.3.1. Visits of International Scientists

Prof. Peter Kornerup (Odense University, Denmark): September 5–19.

Dr. Benoît Libert (Université de Louvain-la Neuve, Belgium), Inria invited researcher: May 28–July 13.

Prof. San Ling (Nanyang Technological University, Singapore), ENS Lyon invited professor: August 20–October 11.

Prof. Dave Saunders (University of Delaware, U.S.A.), ENS Lyon invited professor: April 15–July 25.

9. Dissemination

9.1. Scientific Animation

- Florent de Dinechin was in the program committee of HEART 2012 (Highly Efficient Accelerators for Reconfigurable Computing), FPL 2012 (Field-Programmable Logic), FPT 2012 (Field-Programmable Technologies) and ARC 2012 (Applied Reconfigurable Computing), and on the steering committee of the french Symposium on Architectures.
- Guillaume Hanrot was in the hiring committees for professors at the universities of Caen, Toulon, and UCB Lyon 1. He is a member of the scientific council of ENSIIE (Évry).
- Claude-Pierre Jeannerod, Nicolas Louvet, Nathalie Revol, Dave Saunders (University of Delaware, U.S.A.), Philippe Théveny and Gilles Villard organized the LyonBox meeting that gathered members of the LinBox project (ENS de Lyon, July 19-21).
- Claude-Pierre Jeannerod was in the software exhibits committee of ISSAC 2012. He is also a member of the scientific committee of "Journées Nationales de Calcul Formel".
- Fabien Laguillaumie was in the program committee of ACISP 2012 (17th Australasian Conference on Information Security and Privacy).
- Jean-Michel Muller chaired the Aeres visiting committees of laboratories LIAFA and PPS. He was in the program committee of ARITH'21 (21st IEEE Symposium on Computer Arithmetic) and ASAP'2012 (Application-Specific Systems, Architectures and Processors). He is a member of the scientific councils of École Normale Supérieure de Lyon and Cerfacs.
- Nathalie Revol is a member of the steering and scientific committees of SCAN 2012, Novosibirsk. She organized a session on interval arithmetic at RAIM 2012, Dijon. She is a member of the "comité de diffusion" of the MILyon labex, one of her tasks was the selection of participants to ISSMYS 2012, Lyon. She is a member of the selecting committee of CapMaths. She is a member of the CES (Commission des Emplois Scientifiques), the hiring committee for postdocs at Inria Grenoble - Rhône-Alpes.

- Bruno Salvy is a member of the editorial boards of the Journal of Symbolic Computation and of the Journal of Algebra (section Computational Algebra), as well as the Springer series "Texts and Monographs in Symbolic Computation" and the series "Mathématiques & Applications" of the French SMAI. He is organizing the working group Computer Algebra of the CNRS GDR IM (Mathematical Computer Science). This year, he is a member of the program committee of AofA 2012 (Analysis of Algorithms), Montreal, Analco13 (Analytic Algorithmics and Combinatorics), New Orleans and ISSAC 2013 (Symbolic and Algebraic Computation), Boston. He has also been a member of several committees: PES at Inria; hiring junior researchers (CR) at Inria; hiring professors in Caen and Grenoble; visiting committee of the laboratory Liafa (Paris 7) for the Aeres.
- Damien Stehlé was in the program committees of INDOCRYPT'12, CRYPTO'12 and ISSAC'12. He was in the hiring committees for lecturers at the universities of Grenoble and Montpellier. He is a member of the steering committee of the Cryptography and Coding working group of GDR IM.
- Gilles Villard is chair of LIP laboratory.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Licence : Introduction to computer science, 24h, L1, ENS Lyon / classe passerelle pour l'enseignement supérieur, taught by Guillaume Hanrot (autumn 2012)

Licence : Introduction to Functional Programming, 30h, L1, UCB Lyon 1, taught by Nicolas Louvet (autumn 2012);

Licence : Computer Architecture, 74h, L2, UCB Lyon 1, taught by Nicolas Louvet (spring 2012);

Licence : Algorithms and Data Structures, 30h, L3, UCB Lyon 1, taught by Nicolas Louvet (autumn 2012);

Licence : Operating Systems, 60h, L3, UCB Lyon 1, taught by Nicolas Louvet (spring and autumn 2012);

Licence : "Groupe de lecture" Error-Correcting Codes, 24h, L3, ENS Lyon, taught by Nicolas Brisebarre, Eleonora Guerrini, Guillaume Hanrot and Damien Stehlé (spring 2012);

Master: Numerical Algorithms, M2, UCB Lyon 1, 20h taught by Claude-Pierre Jeannerod and 16h taught by Nicolas Louvet in a 36h course (spring 2012);

Master: Formal Proofs of Floating-Point Algorithms, ENS Lyon, 10h taught by Jean-Michel Muller in a 24h course (autumn 2012);

Master: Computer Arithmetic, 30h, M2, UCB Lyon 1, taught by Vincent Lefèvre (autumn 2012);

Master: Approximations: from symbolic to numerical computation, and applications, ENS Lyon, 12h taught by Nicolas Brisebarre and 12h taught by Bruno Salvy in a 24h course (autumn 2012);

Master: Computer Algebra, 24h, M1, ENS Lyon, taught by Guillaume Hanrot and Claude-Pierre Jeannerod (spring 2012);

Introduction to informatics for high school teachers, part of which (16h) was taught by Guillaume Hanrot (spring 2012).

9.2.2. Supervision

PhD & HdR :

PhD: Jingyan Jourdan-Lu, *Custom floating-point arithmetic for integer processors: algorithms, implementation, and selection* [11], ENS de Lyon - Université de Lyon, defended on November 15, 2012. Supervisors: Claude-Pierre Jeannerod, Christophe Monat (STMicroelectronics Compilation Expertise Center, Grenoble), and Jean-Michel Muller.

PhD: Érik Martin-Dorel, *Contributions à la vérification formelle d'algorithmes arithmétiques* [12], ENS de Lyon - Université de Lyon, defended on September 26, 2012. Supervisors: Micaela Mayero and Jean-Michel Muller.

PhD: Adrien Panhaleux, *Contributions à l'arithmétique flottante : codage et arrondi correct de fonctions algébriques* [13], ENS de Lyon - Université de Lyon, defended on June 27, 2012. Supervisors: Nicolas Louvet and Jean-Michel Muller.

PhD in progress : Nicolas Brunie, *Architecture et réalisation d'un accélérateur reconfigurable à couplage fort pour processeurs parallèles*, begun in September 2010 (CIFRE from April 2011), Florent de Dinechin, Renaud Ayrignac (Kalray).

PhD in progress : Julien Devigne, *Chiffrement pour la protection de la vie privée*, begun in September 2011 (Orange Labs - University of Caen), co-supervised by F. Laguillaumie (together with Sébastien Canard and Brigitte Vallée)

PhD in progress: Adeline Langlois, *Foundations of lattice-based cryptography*, begun in September 2010, supervised by D. Stehlé

PhD in progress: Philippe Théveny, *Numerical quality and high performance in scientific computing on emerging architectures*, begun in September 2011, supervised by N. Revol

PhD in progress : Serge Torres, *Some tools for the design of efficient and reliable function evaluation libraries*, started in September 2010, Supervisors: Nicolas Brisebarre and Jean-Michel Muller.

9.2.3. Juries

- Nicolas Brisebarre was a member of the board of examiners for the PhD defense of A. Benoit (École Polytechnique, 2012-07-18).
- Florent de Dinechin was a referee for the PhD of Naeem Abbas (Université Rennes-1, 2012-05-22).
- Guillaume Hanrot chaired the board of examiners for the PhD defense of P. Lezowski (Univ. Bordeaux 1, 2012-12-01).
- Claude-Pierre Jeannerod was in the PhD committee of J. Jourdan-Lu (ENS de Lyon, 2012-11-15).
- Fabien Laguillaumie was in the PhD committees of Kaoutar Elkhiyaoui (EURECOM, 2012-09-12) and Olivier Blazy (ENS de Paris, 2012-09-27) as a referee and of Roch Lescuyer (Orange Labs/ENS de Paris, 2012-11-21).
- Jean-Michel Muller chaired the boards of examiners for the PhD defenses of T.M.T. Nguyen (Univ. Paris Sud, 2012-06-11) and R. Lebreton (École Polytechnique, 2012-12-11), and was a member of the boards of examiners for the PhD defenses of E. Martin-Dorel (ENS Lyon, 2012-09-26), A. Panhaleux (ENS Lyon, 2012-06-27), and J. Jourdan-Lu (ENS Lyon, 2012-11-15), and for the Habilitation defenses of M. Mayero (Univ. Paris Nord, 2012-11-22) and E. Thomé (Univ. Nancy, 2012-12-13).
- Bruno Salvy has been a member of the PhD committees of P.-J. Spaenlauher (Paris 6, 2012-10-09), O. Roussel (Paris 6, 2012-09-25) and of the committee for the habilitation of E. Thomé (Nancy, 2012-12-13), for which he was a referee.
- Damien Stehlé was in the PhD juries of P. Lezowski (Université Bordeaux 1, 2012-12-01) and G. Quintin (Ecole Polytechnique, 2012-11-22), for which he was a referee.

9.3. Invited Conferences

- Florent de Dinechin gave invited lectures at CERFACS (Toulouse), at the CERN/Intel OpenLab workshop at CERN and and at CASPUR (Rome). He gave talks at Altera (High Wycombe, UK) and Maxeler Technologies (London).
- Jean-Michel Muller gave invited talks in satellite workshop of the SIAM'2012 Conference on Applied Linear Algebra (Valencia, Spain, June 2012), in the Workshop on Numerical Software (Santander, Spain, July 2012), and the LMS Colloquium "Verification and Numerical Algorithms" (London, November 2012).

- Nathalie Revol gave invited talks at national meetings: “Précision numérique” at CNES (Toulouse, January 2012), “Précision et incertitudes” organized by the thematic groups GAMNI and MAIRCI of the SMAI (Paris, February 2012), and at international meetings: “IFIP Working Group 2.5 on Numerical Software” and “Numerical software: design, analysis and verification” (both at Santander, Spain, July 2012).
- Damien Stehlé gave an invited talk at the workshop on Mathematical and Statistical Aspects of Cryptography (Kolkata, India, January 2012), lectures at International Workshop on Recent Advances in Lattice Reduction Algorithms and their Applications (Hyderabad, India, April 2012), an invited talk at Journées Charles Hermite (Nancy, France, June 2012), a plenary invited talk at the international conference SCC 2012 (Castro Urdiales, Spain, July 2012), lectures at Ecrypt Summer School on Lattices (Porto, Portugal, October 2012), an invited talk at the workshop on Post-Quantum Cryptography and Quantum Algorithms (Leiden, Netherlands, November 2012) and an invited talk at Colloquium Jacques Morgenstern (Sophia-Antipolis, December 2012).

9.4. Popularization

- Nathalie Revol gives talks for pupils at collèges and lycées, as an incentive to choose scientific careers: lycée Chabrillan (Montélimar, Drôme), collège Jacques Cœur (Lentilly, Rhône). She gave two talks around Women’s Day, one at collège Pierre Moreto (Thuir, Pyrénées-Orientales) and one for a general audience, invited by the city of Canohes (Pyrénées-Orientales). The French Ministry of Education launched the first week of mathematics: N. Revol took part in the preparation of a television report for France 2 and whe gave two 2-hours talks at lycées Pierre Brossolette (Villeurbanne) and Juliette Récamier (Lyon). For the Science Fair, she gave two talks at ENS Lyon.

10. Bibliography

Major publications by the team in recent years

[1] A. BOSTAN, C.-P. JEANNEROD, É. SCHOST. *Solving structured linear systems with large displacement rank*, in "Theoretical Computer Science", November 2008, vol. 407, n^{o} 1:3, p. 155–181.

[2] N. BRISEBARRE, M. JOLDEŞ. *Chebyshev interpolation polynomial-based tools for rigorous computing*, in "ISSAC 2010: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation", Munich, Germany, S. M. WATT (editor), ACM, July 2010, p. 147-154.

[3] G. HANROT, V. LEFÈVRE, D. STEHLÉ, P. ZIMMERMANN. *Worst Cases of a Periodic Function for Large Arguments*, in "Proceedings of the 18th IEEE Symposium on Computer Arithmetic (ARITH-18)", IEEE Computer Society, 2007, p. 133–140, http://doi.ieeecomputersociety.org/10.1109/ARITH.2007.37.

[4] G. HANROT, D. STEHLÉ. *Improved Analysis of Kannan’s Shortest Lattice Vector Algorithm (Extended Abstract)*, in "Proceedings of Crypto 2007", LNCS, Springer, 2007, vol. 4622, p. 170–186.

[5] C.-P. JEANNEROD, G. VILLARD. *Essentially optimal computation of the inverse of generic polynomial matrices*, in "Journal of Complexity", 2005, vol. 21, n^{o} 1, p. 72–86.

[6] P. KORNERUP, C. LAUTER, V. LEFÈVRE, N. LOUVET, J.-M. MULLER. *Computing Correctly Rounded Integer Powers in Floating-Point Arithmetic*, in "ACM Transactions on Mathematical Software", 2010, vol. 37, n^{o} 1, p. 4:1-4:23.

[7] J.-M. MULLER, N. BRISEBARRE, F. DE DINECHIN, C.-P. JEANNEROD, V. LEFÈVRE, G. MELQUIOND, N. REVOL, D. STEHLÉ, S. TORRES. *Handbook of Floating-Point Arithmetic*, Birkhäuser Boston, December 2010, 572, ISBN: 978-0-8176-4704-9, http://hal.inria.fr/ensl-00379167/en.

[8] A. NOVOCIN, D. STEHLÉ, G. VILLARD. *An LLL-reduction algorithm with quasi-linear time complexity: extended abstract*, in "Proceedings of the 43rd ACM Symposium on Theory of Computing, (STOC 2011)", ACM, 2011, p. 403–412.

[9] N. REVOL, K. MAKINO, M. BERZ. *Taylor models and floating-point arithmetic: proof that arithmetic operations are validated in COSY*, in "Journal of Logic and Algebraic Programming", 2005, vol. 64, p. 135–154.

[10] F. DE DINECHIN, B. PASCA. *Designing Custom Arithmetic Data Paths with FloPoCo*, in "IEEE Design & Test of Computers", July 2011, vol. 28, n^{o} 4, p. 18-27, http://hal.inria.fr/ensl-00646282/en.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] J. JOURDAN-LU. *Custom floating-point arithmetic for integer processors: algorithms, implementation, and selection*, Ecole normale supérieure de lyon - ENS LYON, November 2012.

[12] É. MARTIN-DOREL. *Contributions à la vérification formelle d'algorithmes arithmétiques*, Ecole normale supérieure de lyon - ENS LYON, September 2012, http://hal.inria.fr/tel-00745553.

[13] A. PANHALEUX. *Contributions à l'arithmétique flottante : codages et arrondi correct de fonctions algébriques*, Ecole normale supérieure de lyon - ENS LYON, June 2012, http://hal.inria.fr/tel-00744373.

Articles in International Peer-Reviewed Journals

[14] C. ALIAS, B. PASCA, A. PLESCO. *FPGA-Specific Synthesis of Loop Nests with Pipelined Computational Cores*, in "Microprocessors and Microsystems - Embedded Hardware Design", 2012, vol. 36, n^{o} 8, p. 606-619 [*DOI :* 10.1016/J.MICPRO.2012.06.009], http://hal.inria.fr/hal-00761515.

[15] N. ATTRAPADUNG, J. HERRANZ, F. LAGUILLAUMIE, B. LIBERT, E. DE PANAFIEU, C. RÀFOLS. *Attribute-Based Encryption Schemes with Constant-Size Ciphertexts*, in "Theoretical Computer Science", 2012, vol. 422, p. 15-38, http://hal.inria.fr/hal-00763158.

[16] M. BARDET, J.-C. FAUGÈRE, B. SALVY, P.-J. SPAENLEHAUER. *On the Complexity of Solving Quadratic Boolean Systems*, in "Journal of Complexity", August 2012, vol. 29, n^{o} 1, p. 53-75 [*DOI :* 10.1016/J.JCO.2012.07.001], http://hal.inria.fr/hal-00655745.

[17] X.-W. CHANG, D. STEHLÉ, G. VILLARD. *Perturbation Analysis of the QR Factor R in the Context of LLL Lattice Basis Reduction*, in "Mathematics of Computation", 2012, vol. 81, p. 1487–1511, http://hal.inria.fr/ensl-00529425.

[18] F. DE DINECHIN. *Multiplication by rational constants*, in "IEEE Transactions on Circuits and Systems. Part II, Express Briefs", 2012, vol. 59, n^{o} 2, p. 98-102 [*DOI :* 10.1109/TCSII.2011.2177706], http://hal.inria.fr/ensl-00610328.

[19] P. KORNERUP, V. LEFÈVRE, N. LOUVET, J.-M. MULLER. *On the computation of correctly-rounded sums*, in "IEEE Transactions on Computers", March 2012, vol. 61, n^{o} 3, p. 289-298 [*DOI :* 10.1109/TC.2011.27], http://hal.inria.fr/ensl-00331519.

[20] C. LING, L. LUZZI, D. STEHLÉ. *"Decoding by Embedding: Correct Decoding Radius and DMT Optimality"*, in "IEEE Transactions on Information Theory", 2012, Accepted for publication. Full version of the ISIS'11 proceedings article with the same title., http://hal.inria.fr/hal-00767543.

[21] C. PIVOTEAU, B. SALVY, M. SORIA. *Algorithms for combinatorial structures: Well-founded systems and Newton iterations*, in "Journal of Combinatorial Theory, Series A", November 2012, vol. 119, n^{o} 8, p. 1711-1773 [*DOI :* 10.1016/J.JCTA.2012.05.007], http://hal.inria.fr/inria-00622853.

Invited Conferences

[22] N. REVOL, H. D. NGUYEN, P. THÉVENY. *Tradeoffs between Accuracy and Efficiency for Interval Matrix Multiplication*, in "Numerical Software 2012: Design, Analysis and Verification", Santander, Spain, July 2012, http://hal.inria.fr/hal-00750021.

International Conferences with Proceedings

[23] A. BOSTAN, M. F. I. CHOWDHURY, R. LEBRETON, B. SALVY, É. SCHOST. *Power Series Solutions of Singular (q)-Differential Equations*, in "ISSAC '12: 37th International Symposium on Symbolic and Algebraic Computation", Grenoble, France, M. VAN HOEIJ, J. VAN DER HOEVEN (editors), July 2012, p. 107-114, http://hal.inria.fr/hal-00697733.

[24] A. BOSTAN, F. CHYZAK, Z. LI, B. SALVY. *Fast Computation of Common Left Multiples of Linear Ordinary Differential Operators*, in "ISSAC 2012 - 37th International Symposium on Symbolic and Algebraic Computation", Grenoble, France, M. VAN HOEIJ, J. VAN DER HOEVEN (editors), July 2012, p. 99-106, http://hal.inria.fr/hal-00698610.

[25] N. BRISEBARRE, M. ERCEGOVAC, J.-M. MULLER. *(M,p,k)-friendly points: a table-based method for trigonometric function evaluation*, in "2012 IEEE 23rd International Conference on Application-Specific Systems, Architectures and Processors", Delft, Netherlands, IEEE Computer Society, July 2012, p. 46-52 [*DOI :* 10.1109/ASAP.2012.17], http://hal.inria.fr/ensl-00759912.

[26] N. BRISEBARRE, M. JOLDEŞ, É. MARTIN-DOREL, M. MAYERO, J.-M. MULLER, I. PASCA, L. RIDEAU, L. THÉRY. *Rigorous Polynomial Approximation using Taylor Models in Coq*, in "Fourth NASA Formal Methods Symposium", Norfolk, Virginia, United States, A. GOODLOE, S. PERSON (editors), LNCS, Springer, April 2012, 15, http://hal.inria.fr/ensl-00653460.

[27] N. BRISEBARRE, M. MEZZAROBBA, J.-M. MULLER, C. LAUTER. *Comparison between binary64 and decimal64 floating-point numbers*, in "21st IEEE Symposium on Computer Arithmetic", Austin, TX, United States, IEEE Computer Society, April 2013, 8, to appear, http://hal.inria.fr/ensl-00737881.

[28] N. BRUNIE, S. COLLANGE, G. DIAMOS. *Simultaneous Branch and Warp Interweaving for Sustained GPU Performance*, in "39th Annual International Symposium on Computer Architecture (ISCA)", Portland, OR, United States, June 2012, p. 49 - 60 [*DOI :* 10.1109/ISCA.2012.6237005], http://hal.inria.fr/ensl-00649650.

[29] N. BRUNIE, F. DE DINECHIN, B. DE DINECHIN. *Mixed-precision Fused Multiply and Add*, in "45th Asilomar Conference on Signals, Systems & Computers", United States, March 2012, p. 165-169, http://hal.inria.fr/ensl-00642157.

[30] S. CANARD, G. FUCHSBAUER, A. GOUGET, F. LAGUILLAUMIE. *Plaintext-Checkable Encryption*, in "CT-RSA 2012", San Francisco, United States, O. DUNKELMAN (editor), LNCS, 2012, vol. 7178, p. 332-348, http://hal.inria.fr/hal-00768305.

[31] S. CHEVILLARD, M. MEZZAROBBA. *Multiple precision evaluation of the Airy Ai function with reduced cancellation*, in "21st IEEE Symposium on Computer Arithmetic", Austin, TX, United States, 2013, to appear, http://hal.inria.fr/hal-00767085.

[32] F. DE DINECHIN, L.-S. DIDIER. *Table-based division by small integer constants*, in "Applied Reconfigurable Computing", Hong Kong, Hong Kong, March 2012, http://hal.inria.fr/ensl-00642145.

[33] E. DUTISSEUIL, J.-M. TANGUY, A. VOICILA, R. LAUBE, F. BORE, H. TAKEUGMING, F. DE DINECHIN, F. CEROU, A. GABRIEL CHARLET. *34 Gb/s PDM-QPSK coherent receiver using SiGe ADCs and a single FPGA for digital signal processing*, in "Optical Fiber Communication Conference", United States, March 2012, OM3H.7, http://hal.inria.fr/ensl-00766801.

[34] J. HERRANZ, F. LAGUILLAUMIE, B. LIBERT, C. RÀFOLS. *Short Attribute-Based Signatures for Threshold Predicates*, in "RSA Conference 2012", San Francisco, United States, LNCS, Springer, 2012, vol. 7178, p. 51-67, http://hal.inria.fr/hal-00611651.

[35] C.-P. JEANNEROD, J. JOURDAN-LU. *Simultaneous floating-point sine and cosine for VLIW integer processors*, in "23rd IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP 2012)", Delft, Netherlands, February 2012, p. 69-76, http://hal.inria.fr/hal-00672327.

[36] C.-P. JEANNEROD, J. JOURDAN-LU, C. MONAT. *Non-generic floating-point software support for embedded media processing*, in "7th IEEE International Symposium on Industrial Embedded Systems (SIES'12)", Karlsruhe, Germany, 2012, http://hal.inria.fr/hal-00695333.

[37] C.-P. JEANNEROD, N. LOUVET, J.-M. MULLER. *On the componentwise accuracy of complex floating-point division with an FMA*, in "21st IEEE Symposium on Computer Arithmetic", Austin, TX, United States, IEEE Computer Society, April 2013, 8, to appear, http://hal.inria.fr/ensl-00734339.

[38] F. LAGUILLAUMIE, G. CASTAGNOS. *Homomorphic Encryption for Multiplications and Pairing Evaluation*, in "Security and Cryptography for Networks - 8th International Conference, SCN 2012", Amalfi, Italy, I. VISCONTI, R. DE PRISCO (editors), 2012, http://hal.inria.fr/hal-00763110.

[39] V. LEFÈVRE. *SIPE: Small Integer Plus Exponent*, in "21st IEEE Symposium on Computer Arithmetic", Austin, TX, United States, IEEE Computer Society, 2013, to appear, http://hal.inria.fr/hal-00763954.

[40] S. LING, K. NGUYEN, D. STEHLÉ, H. WANG. *Improved Zero-knowledge Proofs of Knowledge for the ISIS Problem, and Applications*, in "Proceedings of PKC 2013", Japan, 2013, to appear, http://hal.inria.fr/hal-00767548.

[41] M. MEZZAROBBA. *A Note on the Space Complexity of Fast D-Finite Function Evaluation*, in "CASC - Computer Algebra in Scientific Computing", Maribor, Slovenia, V. GERDT, W. KOEPF, E. MAYR, E.

VOROZHTSOV (editors), LNCS, Springer, 2012, vol. 7442, p. 212-223 [*DOI :* 10.1007/978-3-642-32973-9_18], http://hal.inria.fr/hal-00687818.

[42] H. D. NGUYEN, N. REVOL, P. THÉVENY. *Tradeoffs between Accuracy and Efficiency for Optimized and Parallel Interval Matrix Multiplication*, in "PARA 2012 - Workshop on the State-of-the-Art in Scientific and Parallel Computing", Helsinki, Finland, LNCS, Springer, June 2012, http://hal.inria.fr/hal-00704288.

National Conferences with Proceeding

[43] N. BRUNIE, F. DE DINECHIN, B. DE DINECHIN. *Conception d'une matrice reconfigurable pour coprocesseur fortement couplé*, in "Symposium en Architectures nouvelles de machines", France, January 2013, http://hal.inria.fr/ensl-00763067.

[44] N. BRUNIE, F. DE DINECHIN, M. ISTOAN, G. SERGENT. *L'arithmétique sur le tas*, in "Symposium en Architectures nouvelles de machines", France, January 2013, http://hal.inria.fr/ensl-00762990.

Conferences without Proceedings

[45] N. REVOL. *IEEE-1788 standardization of interval arithmetic: work in progress (a personal view)*, in "IFIP Working Group 2.5 on Numerical Software", Santander, Spain, July 2012, Invited talk, http://hal.inria.fr/hal-00759206.

[46] N. REVOL. *L'effort de normalisation IEEE-1788 de l'arithmétique par intervalles*, in "RAIM 2012 : Rencontres "Arithmétique de l'Informatique Mathématique"", Dijon, France, June 2012, http://hal.inria.fr/hal-00750019.

[47] P. THÉVENY, N. REVOL. *Interval matrix multiplication on parallel architectures*, in "SCAN 2012: 15th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computations", Novosibirsk, Russian Federation, September 2012, http://hal.inria.fr/hal-00750022.

[48] P. THÉVENY. *Divers algorithmes de produits de matrices intervalles*, in "RAIM 2012 : Rencontres "Arithmétique de l'Informatique Mathématique"", Dijon, France, June 2012, http://hal.inria.fr/hal-00750017.

Scientific Books (or Scientific Book chapters)

[49] F. DE DINECHIN, B. PASCA. *Reconfigurable arithmetic for HPC*, in "High-Performance Computing using FPGAs", W. VANDERBAUWHEDE, K. BENKRID (editors), Springer, March 2013, http://hal.inria.fr/ensl-00758377.

Research Reports

[50] F. DE DINECHIN, P. ECHEVERRIA, M. LOPEZ-VALLEJO, B. PASCA. *Floating-Point Exponentiation Units for Reconfigurable Computing*, ENS Lyon, July 2012, To appear in ACM Transactions on Reconfigurable Technology and Systems, http://hal.inria.fr/ensl-00718637.

[51] F. DE DINECHIN, M. ISTOAN, G. SERGENT, K. ILLYES, B. POPA, N. BRUNIE. *Arithmetic around the bit heap*, ENS Lyon, October 2012, http://hal.inria.fr/ensl-00738412.

[52] F. DE DINECHIN, C. LAUTER, J.-M. MULLER, S. TORRES. *On Ziv's rounding test*, ENS Lyon, 2012, http://hal.inria.fr/ensl-00693317.

Other Publications

[53] J.-C. BELFIORE, C. LING, L. LUZZI, D. STEHLÉ. *Semantically Secure Lattice Codes for the Gaussian Wiretap Channel*, 2012, Maths Arxiv posting, http://hal.inria.fr/hal-00767551.

[54] A. LANGLOIS, D. STEHLÉ. *Worst-Case to Average-Case Reductions for Module Lattices*, 2012, Eprint archive posting, http://hal.inria.fr/hal-00767547.

[55] S. LING, D. STEHLÉ. *A Lattice-Based Traitor Tracing Scheme*, 2012, Eprint archive posting, http://hal.inria.fr/hal-00767545.

References in notes

[56] J.-M. MULLER. *Elementary Functions, Algorithms and Implementation*, Birkhäuser Boston, 2nd Edition, 2006.