IN PARTNERSHIP WITH:
CNRS

Université Claude Bernard (Lyon 1)

Ecole normale supérieure de Lyon

Activity Report 2013

Project-Team ARIC

Arithmetic and Computing

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

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Algorithmics, Computer Algebra and Cryptology

Table of contents

Project-Team ARIC

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

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

1. Members

Research Scientists

Nicolas Brisebarre [CNRS, Researcher]
Claude-Pierre Jeannerod [Inria, Researcher]
Vincent Lefèvre [Inria, Researcher]
Jean-Michel Muller [Team leader since Sep 2013; CNRS, Senior Researcher, HdR]
Nathalie Revol [Inria, Researcher]
Bruno Salvy [Inria, Senior Researcher]
Gilles Villard [CNRS, Senior Researcher, HdR]

Faculty Members

Florent de Dinechin [Team leader until Aug 2013; Professor at INSA Lyon from Sep 2013, HdR]
Stef Graillat [Univ. Paris 6, Associate Professor; CNRS partial secondment since Sep 2013, HdR]
Guillaume Hanrot [ENS de Lyon, Professor, HdR]
Fabien Laguillaumie [Univ. Lyon I, Professor, HdR]
Nicolas Louvet [Univ. Lyon I, Associate Professor]
Clément Pernet [Univ. Grenoble I, Associate Professor; CNRS partial secondment since Sep 2013]
Damien Stehlé [ENS de Lyon, Professor, HdR]

Engineers

Andrea Cameli [Inria, ANR TaMaDi project, from Sep 2013 until Dec 2013]
Matei Istoan [Inria, ANR TaMaDi project, until Jun 2013]
Serge Torres [ENS de Lyon]

PhD Students

Nicolas Brunie [CIFRE grant (Kalray)]
Silviu Filip [ENS de Lyon]
Adeline Langlois [ENS de Lyon]
Vincent Neiger [ENS de Lyon; Western University, London, Canada (international co-direction)]
Marie Paindavoine [Orange Labs]
Philippe Théveny [ENS de Lyon]

Post-Doctoral Fellows

Nicolas Estibals [ENS de Lyon, Temporary Lecturer (ATER), until Aug 2013]
Rishiraj Bhattacharyya [Inria]
Marc Mezzarobba [Inria, ANR TaMaDi project, until Apr 2013]

Visiting Scientists

Xiao-Wen Chang [ENS de Lyon, from Apr 2013 to Jun 2013]
Saruchi Goel [ENS de Lyon, from Apr 2013 to Aug 2013]
Warwick Tucker [ENS de Lyon, from Feb 2013 to Mar 2013]

Administrative Assistant

Damien Séon [ENS de Lyon]

Others

Cécile Baritel [Inria, summer internship, from Jun 2013 to Sep 2013]
Antoine Martinet [Inria, summer internship, from Jun 2013 to Jul 2013]

2. Overall Objectives

2.1. Overview

The overall objective of AriC is, through computer arithmetic, to improve computing at large, in terms of performance, efficiency, and reliability. Specifically, we focus on the following domains:

1. **Floating-point arithmetic:** The IEEE 754-2008 standard specifies the behavior of floating-point arithmetic. We are interested in preparing future evolutions of the standard, in implementing it efficiently on embedded processors, in exploring its "low level" properties for better numerical analysis (for instance by finding certified and tight error bounds of numerical algorithms), and in building correctly rounded mathematical function programs. We are also interested in designing efficient algorithms and software for multiple-precision arithmetic and complex arithmetic.
2. **Certified computing and computer algebra:** We are interested in computing certified approximations using computer algebra and formal proof systems, in analyzing the fundamental algorithms of semi-numerical computation, in finding best or nearly best approximations under special constraints, and in designing efficient algorithms for exact linear algebra. Also, we are working on the development and standardization of interval arithmetic.
3. **Hardware and FPGA arithmetic:** The main challenge here is the design of efficient arithmetic hardware. Instead of designing ad-hoc operators for a given technology and a given target (in terms of speed, accuracy, or power consumption requirements), we aim at building algorithms and programs that automatically design them. This allows one to find better solutions by being able to explore a larger part of the (in general, huge) design space, and to build specific operators for frequent "compound" functions such as, for example, $x/\sqrt{x^2+y^2}$.
4. **Cryptography and lattices:** Lattice-based cryptography (LBC) is a fast developing field, raising fascinating questions both on cryptography and lattices. Lattice algorithmics is an established research area that is being revived by the amazing application that is LBC and by the new tools and concepts that it introduced. We aim at contributing to a major technological switch, from conventional to lattice-based cryptography. This will help suppress the main limitation to the expansion of the cloud economy that are the privacy concerns. Further, thanks to the ubiquity of lattices, our work may significantly impact several other fields, including coding, computer algebra, and computer arithmetic.

2.2. Highlights of the Year

- Jean-Michel Muller received the CNRS-INS2I silver medal.
- Damien Stehlé was awarded a "starting" ERC grant for his project "Euclidean lattices: algorithms and cryptography" (LattAC).
- Vincent Lefèvre, Nicolas Louvet, and Jean-Michel Muller received the "Prix La Recherche pour les Sciences de l'Information".

3. Research Program

3.1. Introduction

We detail below the various themes of our research program. They all relate to one or several of our four main domains of interest: floating-point arithmetic; certified computing and computer algebra; hardware and FPGA arithmetic; and cryptography and lattices.

3.2. Function evaluation

3.2.1. Towards automatic design of function programs or circuits

Concerning function evaluation, what we have successfully automated so far is program generation for statically defined elementary functions of one real variable. These techniques will certainly need refining as we try to apply them to more functions, in particular to special functions, or to compound functions. To apply these techniques to arbitrary code at compile time further leads to several challenges. The first one is to identify a relevant function in a program, along with the useful information that will allow to implement it efficiently: range of the input values, needed output accuracy and/or rounding mode, etc. It requires interaction with compilation people working on classical compilers.

A second challenge then is to analyze such a function automatically, which typically implies the following tasks:

- Compute maximal-width subranges on which the output is trivial, that is, requires no computation at all (zero, infinity, etc.). This can be extremely tedious and error-prone if done by hand. An important side effect of this step is to generate test vectors for corner cases.
- Identify properties answering to questions like "Is further range reduction possible?", "Are there floating-point midpoints?", "Can overflow occur?", etc. Today, this is essentially handwritten by experts, so the challenge is to automate it.
- Investigate automating range reduction. Generic reduction schemes can be used (for example, interval splitting into sub-intervals) but involve many parameters, and we have to model the associated cost/performance/accuracy tradeoffs. More function-specific range reduction steps can be an outcome of the previous analysis steps.

This general automation process will be progressively set up and refined by working on concrete implementations of a significant set of operators, hopefully driven by applications and through industrial collaborations: all the C99 elementary functions, other functions such as the inverse cumulative distribution functions used in random number generators, variations around the Euclidean norm such as $x/\sqrt{x^2+y^2}$, complex arithmetic, interval operators, FFTs, etc.

Most of the software we design brings in some floating-point functionality. We currently target two main types of processors:

- Embedded processors, with or without a floating-point unit (FPU), for which we need to implement the basic operators, coarser elementary functions, and compile-time arbitrary functions. On such targets, memory may be limited, and power consumption is important.
- General-purpose processors that do possess an FPU for the basic operations. The challenge is then to use this FPU to its best to implement coarser operators and compile-time functions. The main metrics here are performance and, to a lesser extent, code size.

3.2.2. Mathematical tools for function evaluation

3.2.2.1. Challenges in function approximation

The algorithms currently implemented in the Sollya toolbox (see Section 5.1) provide, for functions of one variable, an end-to-end solution for finding near-optimal polynomial approximations whose coefficients are machine numbers. This includes a validated tight bound of the approximation error. We now want to generalize them in three main directions:

- We first want to design and implement algorithms for computing *multivariate* polynomials that approximate very well a given multivariate function.
- We also want to address the approximation of a function by a *rational* function. This is important for the software implementation of functions with poles. It would also make practical a hardware-oriented technique called the E-method.

- Currently, our algorithms and their implementations use a basis of the form (x^i) where i belongs to a finite subset of $\mathbb{N}$ with a cardinality bounded by, say, 100. We now aim at dealing with *other bases*. In particular, the classical basis of Chebyshev polynomials should lead to a better numerical analysis without losing any efficiency during evaluation. In general, we should be able to deal with any basis made of orthogonal polynomials. Using as basis the trigonometric polynomials should lead to efficient finite-precision *digital filters*.

In these three directions, we eventually want to constraint coefficients to be machine numbers.

3.2.2.2. Approximation for digital filters

A digital filter implements a given transfer function, either as a polynomial (finite impulse response, or FIR filter) or as a rational function (infinite impulse response, or IIR filter). Classical techniques and toolboxes exist for computing such filters, but they amount to computing infinite precision solutions and rounding them. This rounding turns out to be numerically unstable in some situations. We intend to study to what extent an improved rounding procedure might improve the filter, in terms of efficiency (software implementation) or size (hardware implementation).

Collaborations on this subject have begun with researchers from the signal processing community.

3.2.2.3. Challenges in the search for hard-to-round cases

For a given function f and a given floating-point format, the "hardest to round" (HR) points are the floating-point numbers x such that $f(x)$ is nearest to a value where the rounding function changes. Knowing these HR points makes it possible to design efficient programs that, given a floating-point number y, always return the floating-point number nearest $f(y)$. Such programs are called "correctly rounded" implementations of f.

We have obtained and published HR points in the binary64 ("double precision") and decimal64 formats of the IEEE 754-2008 standard for the most important functions of the standard mathematical library. However, in this line of research, we now have to tackle difficult challenges. First, our current methods for finding HR-points cannot be used in big precisions such as the binary128 ("quad precision") and decimal128 (128-bit decimal) formats of the IEEE 754-2008 standard. Also, the processes that generate our HR cases are based on complex and very long calculations (years of cumulated CPU time) that inevitably cast some doubt on the correctness on their results. Hence, we reconsider the methods used to get HR points, and mainly focus 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 of as more than enough for a final result, it may not be sufficient for the intermediate results of a long or critical calculation);
- **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.

These three aspects have been at the core of our TaMaDi ANR project (see Section 8.1.2): the project brought significant progress, but there is still much to be done.

3.3. Hardware and FPGA arithmetic

The main characteristic of reconfigurable circuits, or FPGAs, is precisely their reconfigurability: the circuit implemented in an FPGA can be changed according to the needs of the target applications. The challenge here is to exploit this reconfigurability to design operators specifically for the applications: not only should their low-level architecture match the peculiar metrics of FPGAs, but also, the high-level architecture, and even the operator specifications should be as application-specific as possible, and probably completely different to what we are used to design into VLSI circuits. Indeed, operators that would make no economical sense in a processor make perfect sense in an FPGA if an application requires them.

Exotic operators worth considering include specialized operators (such as a multiplier by a constant, a squarer, etc.), arbitrary numerical functions, fused operators such as the Euclidean norm $\sqrt{x^2+y^2}$, etc. The list is infinite, and exploring it is the purpose of the FloPoCo operator generator project started in 2008 (see Section 5.2). We plan to extend it to support many more operators, starting with a fully featured mathematical library.

To support this research on new operators, FloPoCo is also a prototype of arithmetic core generator in constant evolution. It already features an original approach to the generation of efficient and correct-by-construction arithmetic pipelines, and testbench generation. With increasingly complex operators, we now need to enrich it with a clean support for fixed-point semantic.

FloPoCo is also designed as a back-end for high-level synthesis (HLS) tools. The highly pipelined operators of FloPoCo may require specific optimization work from an HLS compiler.

3.4. Lattice-based cryptography

Lattice-based cryptography (LBC) is an utterly promising, attractive (and competitive) research ground in cryptography, thanks to a combination of unmatched properties:

- **Improved performance.** LBC primitives have low asymptotic costs, but remain cumbersome in practice (e.g., for parameters achieving security against computations of up to 2100 bit operations). To address this limitation, a whole branch of LBC has evolved where security relies on the restriction of lattice problems to a family of more structured lattices called *ideal lattices*. Primitives based on such lattices can have quasi-optimal costs (i.e., quasi-constant amortized complexities), outperforming all contemporary primitives. This asymptotic performance sometimes translates into practice, as exemplified by NTRUEncrypt.
- **Improved security.** First, lattice problems seem to remain hard even for quantum computers. Moreover, the security of most of LBC holds under the assumption that standard lattice problems are hard in the worst case. Oppositely, contemporary cryptography assumes that specific problems are hard with high probability, for some precise input distributions. Many of these problems were artificially introduced for serving as a security foundation of new primitives.
- **Improved flexibility.** The master primitives (encryption, signature) can all be realized based on worst-case (ideal) lattice assumptions. More evolved primitives such as ID-based encryption (where the public key of a recipient can be publicly derived from its identity) and group signatures, that were the playing-ground of pairing-based cryptography (a subfield of elliptic curve cryptography), can also be realized in the LBC framework, although less efficiently and with restricted security properties. More intriguingly, lattices have enabled long-wished-for primitives. The most notable example is homomorphic encryption, enabling computations on encrypted data. It is the appropriate tool to securely outsource computations, and will help overcome the privacy concerns that are slowing down the rise of the cloud.

We wish to address three issues, described below.

3.4.1. Design of versatile cryptosystems

We will design standard and important cryptographic primitives in the LBC framework, in particular primitives that can be realized with the integer factorization problem, with the discrete logarithm problem over generic groups, and with the discrete logarithm problem over elliptic curves with pairings. This is a first step towards the longer-term goal of showing the superiority of the LBC framework in terms of possible functionalities.

We will first consider group signatures, that enable a member of a group to anonymously sign a document in the name of the group, while allowing a group authority to remove the anonymity and to trace the signer from the signature.

Another primitive we will consider is traitor tracing, a type of broadcast encryption where unauthorized decryption boxes can be used to trace the keys that were used to build them. Traitor tracing is sometimes viewed as the encryption counterpart of group signature. The objective here will be to improve the sole LBC traitor tracing scheme to efficiently achieve full traceability (where all users can collude to build a pirate decryption box), as can be achieved with pairings.

Additionally, we will consider functional encryption, which enables the decryption of ciphertexts by a set of users who share some specific attributes. This sophisticated protocol is hard to design if one needs the scheme to be secure in the strongest sense, or the description of the attributes to be very expressive, while maintaining efficiency.

3.4.2. Security foundations of LBC

We wish to strengthen the security foundations of LBC. This will be achieved by unifying the diverse hardness assumptions and showing that the LBC hardness assumptions are weaker than the Integer Factorization and Discrete Logarithm problems.

Most LBC primitives rely on the worst-case hardness of standard and well-studied problems on lattices. The primitives are typically constructed via Ajtai's Short Integer Solution problem (SIS) and Regev's Learning With Errors problem (LWE), to which standard lattice problems reduce. SIS and LWE are more fitted to devise cryptosystems, as they are average-case in nature. However, other primitives, and in particular the most efficient ones, rely on less accepted hardness assumptions than SIS and LWE (and thus worst-case hardness assumptions on standard lattice problems).

The LBC primitives based on the variant problems Ring-SIS/Ring-LWE, and thus lattice problems restricted to ideal lattices, are drastically more efficient than those based on SIS/LWE. It is therefore a central objective to prove that these problems are hard. Another assumption commonly used is the hardness of the Approx-GCD problem. This problem consists in finding p from many samples $a_i \cdot p + b_i$ with small random a_i and b_i. It is not known to be harder than any standard lattice problem but is used as a security foundation anyway. It is an attractive open problem to prove its difficulty, for example by reducing standard problems over lattices to it. Achieving the above goals will unify the hardness assumptions underlying the security of LBC. We will also investigate alternatives to the well-accepted LWE/SIS approach.

3.4.3. The rise of efficient lattice-based cryptography

Increasing the efficiency of LBC requires algorithmic and implementation research efforts. The efficient cryptographic primitives rely on ideal lattices. These correspond, via the coordinates-coefficients mapping, to ideals of polynomial rings $Z[x]/(P)$, where P is a large degree irreducible polynomial, such as $x^n + 1$ with n a power of 2. They may also be defined as the ideals of the rings of integers of large-degree number fields (in the case of cyclotomic number fields, these definitions are equivalent). The cryptographic primitives relying on ideal lattices typically involve two types of tasks: multiplications and additions of polynomials in the ring $(Z/pZ)[x]/(P)$, where p is a medium-size integer (e.g., of 10 to 60 bits), and sampling from discrete Gaussian distributions (the integer counterpart of the normal law). We will optimize their algorithms and implementations. The objective is to obtain efficient software and hardware implementations of basic LBC primitives such as digital signatures and encryption.

Polynomial arithmetic is well known and has been well studied in computer algebra, but has not been optimized over rings Z/pZ where p has medium bit-size: so far, either very large (hundreds of bits) or very small (2 and 3) moduli have been considered. We will optimize the existing algorithms for this new range of parameters, for both software and hardware. Sampling from the (continuous) Gaussian distribution is also a well-studied topic. However, in LBC, we are mostly interested in the discrete Gaussian distribution, where the probability of obtaining the integer x is proportional to $exp(-\pi \cdot x^2/s^2)$, for any x. All known algorithms for this task are very slow.

3.5. Floating-point arithmetic

3.5.1. Properties of floating-point arithmetic

Thanks to the IEEE 754-2008 standard for floating-point arithmetic, we now have an accurate definition of floating-point formats and operations. The behavior of a sequence of operations becomes at least partially predictable. We therefore can build algorithms and proofs that use these specifications. Some of these algorithms are new, some others have been known for years, but only for radix-2 systems. Also, their proofs are not exempt from flaws: some algorithms do not work, for instance, when subnormal numbers appear. We wish to give rigorous proofs, including the exact domain of validity of the corresponding algorithms, and to extend when possible these algorithms and proofs to new formats specified by the recent floating-point standard (decimal formats, large precision formats).

3.5.2. Error-free transformations and compensated algorithms

To achieve a prescribed accuracy for the result of a given computation, it is often necessary to increase the precision of the intermediate operations beyond the highest precision available in hardware. On superscalar processors, an efficient solution is to compute, at runtime, the error due to critical floating-point operations, in order to later compensate for them. Such compensated algorithms have been studied for the summation of $n > 2$ floating-point numbers and for polynomial evaluation. They are based on *error-free transformations* (EFT): small, efficient algorithms, based on the specifications of the IEEE 754-2008 standard, that compute the sum or product of two floating-point number exactly. The result of an EFT is represented exactly as two floating-point numbers, one holding the rounded result, and the other holding the error term. We will keep investigating EFTs, and study compensated algorithms improving the accuracy of other computing kernels (such as matrix-vector and matrix-matrix products) in the context of vector floating-point units and multicore architectures.

3.6. Certified computing

3.6.1. Bounding roundoff errors and ranges

Many error analysis techniques are well known for obtaining *a priori* bounds on the global roundoff error generated by a floating-point program. Such error bounds can already be computed automatically with Gappa in the case of straight-line programs, assuming the precision of every arithmetic operation is fixed and known in advance. One of our next challenges will be to handle *a priori* error bounds in algorithms where the computing precision varies with each operation (this is the case in operators designed for FPGAs, or for some MPFR code for instance), and for programs involving loops of variable length.

On the other hand, techniques are also available to compute rigorous *a posteriori* error bounds. Interval arithmetic is probably the best established technique for this, but certified *a posteriori* error bounds can also be computed, at run-time, using error-free transformations and, more generally, the specifications of the IEEE-754 floating-point arithmetic. We plan to investigate and compare these two approaches.

Interval methods can also be used more generally for computing a rigorous enclosure for the range of a function on a given domain. This corresponds to the case where input data vary in a set. One can then deduce properties such as the occurrence of overflows, or the sign of the result. We will work on the automatic detection of such properties.

3.6.2. Higher order techniques: Taylor models, Chebyshev models

The team started developing Chebyshev models, an improvement of Taylor models which replaces Taylor approximation with Chebyshev interpolant approximation or Chebyshev truncated series approximation. The main advantage of these models is that they offer better convergence properties, leading to smaller remainders and converging on more flexible domains. We will investigate applications of Taylor and Chebyshev models to classical issues in rigorous computing, such as global optimization, certified quadrature or rigorous solving of ordinary differential equations. We also need to address the challenges of Taylor and Chebyshev models when implemented using floating-point arithmetic: combining high accuracy with the performance of hardware-supported arithmetic, possibly using error-free transformations.

3.6.3. Formal proof

The methods mentioned so far certify the quality of the result ... up to a bug in their implementation. To get a higher degree of confidence, the certification should be checked by a theorem prover such as Coq. It requires that the chosen arithmetic or method is implemented and proven within the theorem prover. This is not yet the case of symbolic-numeric computation of error bounds in the case or variable precision, nor of higher order variants of interval arithmetic. We are working on these formalizations with partners from the formal proof community.

3.6.4. Standardization

An ongoing work is the standardization of interval arithmetic, by the IEEE 1788 working group. We will also continue to participate to the C++ standardization, regarding the inclusion of interval arithmetic in the STL.

3.7. Linear algebra and polynomial evaluation

Linear algebra and polynomial evaluation are key tools for the design, synthesis, and validation of fast and accurate arithmetics. Conversely, arithmetic features can have a strong impact on the cost and numerical quality of computations with matrices, vectors, and polynomials. Thus, deepening our understanding of such interactions is one of our major long-term goals.

3.7.1. Code generation for polynomial expression evaluation

We plan to improve our work on code generation for polynomials, and to extend it to general arithmetic expressions as well as to operations typical of level 1 BLAS, like sums and dot products. Due to the intrinsic multivariate nature of such problems, the number of evaluation schemes is huge and a challenge here is to compute, and certify, in a reasonable amount of time, evaluation programs that satisfy both efficiency and accuracy constraints. To achieve this goal, we will in particular study the design of specific program transformation techniques driven by our certification tools for tight and guaranteed error bounds.

3.7.2. Exact linear algebra

We will pursue our work on the design and analysis of fast algorithms for exact linear algebra in three directions. First, for general matrices over a field k (algebraic complexity model), we want to improve upon existing algorithms by achieving efficiency both in terms of arithmetic cost (expressed via the exponent of matrix multiplication and the rank of the matrix) and in terms of memory usage (in-place algorithms). Second, this approach will be extended to families of structured matrices (Toeplitz-like, etc.) using the displacement rank as an additional parameter in the cost analyses; another challenge here is to move toward a complete understanding of complexity reductions from one structure to another. A third direction deals with polynomial matrices, that is, matrices over $k[x]$. Currently, algorithms for polynomial matrices allow to solve a few structured linear algebra problems more satisfactorily than with the classical structured linear algebra approach. Such algorithms do not have structured matrix analogues, and we thus plan to work on unifying these two seemingly different settings.

3.7.3. Condition numbers

A standard approach for reaching a prescribed output accuracy for the solution to a given problem is to try to compute approximate solutions to this problem using increasing precisions: for each precision, a certified error bound is computed, and the process stops when the prescribed accuracy is reached.

Combined with backward error analysis techniques, computing condition numbers is a well-known technique to obtain first-order error bounds on the computed solution to a given problem. Conversely, condition numbers can be used to estimate the precision required to obtain a prescribed output accuracy, thus accelerating the convergence of certified algorithms. Future research will focus on the computation or estimation of the conditioning of matrix factorizations (LU, QR, ...), in particular on algorithmic complexity issues and efficient software implementations. We will also investigate the use of automatic differentiation as a tool for computing condition numbers.

3.7.4. Iterative refinement methods for linear algebra

Another direction deals with improving the efficiency and quality of self-validating methods for computing error bounds at run time. The starting point is the result of a floating-point computation, like linear system solving. We aim at computing a bound on the error between that approximate result and the exact result, using interval arithmetic to get an enclosure. We believe that the methods of choice are iterative refinement methods: such methods are contractant, and thus particularly well-suited for interval computations. However, it is wise to use optimized floating-point routines for linear algebra, to reach the performances achieved in high-performance computing. Again, this work covers all aspects, from the manual proof of convergence to efficient implementation.

3.7.5. High performance linear algebra and links with Euclidean lattice reduction

Our theoretical studies on linear algebra will be applied to the design of high performance building blocks, scientific computing/computer algebra patterns, and linear algebra algorithms. Our aim in software design is especially to transfer our future research results on: the interplay between bit complexity and algebraic complexity; the interplay between exact computing and approximate (or certified) computing; asymptotically fast algorithms. Current lattice basis reduction algorithms heavily rely on fast linear algebra. High performance basis reduction will be one of our main directions.

4. Application Domains

4.1. Hardware Arithmetic

The application domains of hardware arithmetic operators are

- digital signal processing,
- image processing,
- embedded applications,
- reconfigurable computing,
- cryptography.

4.2. Floating-point and Validated Numerics

Our expertise on validated numerics is useful to analyze and improve, and guarantee the quality of numerical results in a wide range of applications including:

- scientific simulation,
- global optimization,
- 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.

Another interesting field of application is

- communications theory.

5. Software and Platforms

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 2013.

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 [32]. FloPoCo is a generator of operators written in C++ and outputting synthesizable VHDL automatically pipelined to an arbitrary frequency.

2013 saw more work on the *bit-heap* framework [28], [18]. In addition, several new operators were added, in particular for fixed-point sine, cosine [21] and arctangent.

Version 2.5.0 was released in 2013.

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. Tohoku, U. Tokyo, Virginia Tech U. and several companies.

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

- Version: 2.5.0 (June 2013)
- APP: IDDN.FR.001.400014.000.S.C.2010.000.20600 (version 2.0.0)
- License: pending, should be 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.

GNU MPFR 3.1.2 was released on 13 March 2013.

The main work done in the AriC project-team:

- Bug fixes and improved portability.
- Complete revision of the behavior on special values (signed zeros and infinities) and consistency with standards (IEEE 754-2008, ISO C, POSIX) checked. Thanks to this work, several problems in MPFR and the POSIX specification have been detected and the MPFR manual has been completed: https://sympa.inria.fr/sympa/arc/mpfr/2013-12/msg00001.html

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 and some minor bugs have been fixed.

5.5. FPLLL: A Lattice Reduction Library

Participant: 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 4.0.4 was released in 2013, fixing a number of user-interface bugs.

URL: http://perso.ens-lyon.fr/damien.stehle/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://perso.ens-lyon.fr/damien.stehle/fplll/fplll-doc.html

5.6. Sipe

Participant: Vincent Lefèvre.

Sipe is a mini-library in the form of a C header file, to perform radix-2 floating-point computations in very low precisions with correct rounding, either to nearest or toward zero. 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 addition, subtraction, multiplication (possibly with the error term), fused multiply-add/subtract (FMA/FMS), and miscellaneous comparisons and conversions. Sipe provides two implementations of these operations, with the same API and the same behavior: one based on integer arithmetic, and a new one based on floating-point arithmetic; see [25], [39].

New in 2013:

- the floating-point implementation;
- rounding toward zero (only with the integer implementation).
- 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: comment at the beginning of the code and Research report Inria RR-7832.
- URL: https://www.vinc17.net/research/sipe/

6. New Results

6.1. Cryptography and lattices

6.1.1. Group signatures

Group signatures are cryptographic primitives where users can anonymously sign messages in the name of a population they belong to. Gordon et al. (Asiacrypt 2010) suggested the first realization of group signatures based on lattice assumptions in the random oracle model. A significant drawback of their scheme is its linear signature size in the cardinality N of the group. A recent extension proposed by Camenisch et al. (SCN 2012) suffers from the same overhead.

F. Laguillaumie, A. Langlois, B. Libert (Technicolor), and D. Stehlé described in [24] the first lattice-based group signature schemes where the signature and public key sizes are essentially logarithmic in N (for any fixed security level). Their basic construction only satisfies a relaxed definition of anonymity (just like the Gordon et al. system) but readily extends into a fully anonymous group signature (i.e., that resists adversaries equipped with a signature opening oracle). They proved the security of their schemes in the random oracle model under the SIS and LWE assumptions.

6.1.2. Classical hardness of learning with errors

Z. Brakerski (Stanford U.), A. Langlois, C. Peikert (Georgia Institute of Technology), O. Regev (Courant Institute, New York U.), and D. Stehlé showed in [16] that the Learning with Errors (LWE) problem is classically at least as hard as standard worst-case lattice problems, even with polynomial modulus. Previously this was only known under quantum reductions. Their techniques capture the tradeoff between the dimension and the modulus of LWE instances, leading to a much better understanding of the landscape of the problem. The proof is inspired by techniques from several recent cryptographic constructions, most notably fully homomorphic encryption schemes.

6.1.3. Improved Zero-knowledge Proofs of Knowledge for the ISIS Problem, and Applications

In all existing efficient proofs of knowledge of a solution to the infinity norm Inhomogeneous Small Integer Solution ISIS_∞ problem, the knowledge extractor outputs a solution vector that is only guaranteed to be $\widetilde{O}(n)$ times longer than the witness possessed by the prover. As a consequence, in many cryptographic schemes that use these proof systems as building blocks, there exists a gap between the hardness of solving the underlying ISIS_∞ problem and the hardness underlying the security reductions. Together with S. Ling, K. Nguyen, and H. Wang (Nanyang Technological University, Singapore), D. Stehlé generalized in [26] Stern's protocol to obtain two statistical zero-knowledge proofs of knowledge for the ISIS_∞ problem that remove this gap. Their result yields the potential of relying on weaker security assumptions for various lattice-based cryptographic constructions. As applications of their proof system, they introduced a concurrently secure identity-based identification scheme based on the worst-case hardness of the $\text{SIVP}_{\widetilde{O}(n^{1.5})}$ problem (in the L2 norm) in general lattices in the random oracle model, and an efficient statistical zero-knowledge proof of plaintext knowledge with small constant gap factor for Regev's encryption scheme.

6.1.4. Decoding by Embedding: Correct Decoding Radius and DMT Optimality

In lattice-coded multiple-input multiple-output (MIMO) systems, optimal decoding amounts to solving the closest vector problem (CVP). Embedding is a powerful technique for the approximate CVP, yet its remarkable performance is not well understood. In [8], C. Ling (Imperial College, London), L. Luzzi (ENSEA, U. Cergy Pontoise), and D. Stehlé analyzed the embedding technique from a bounded distance decoding (BDD) viewpoint. They proved that the Lenstra, Lenstra and Lovász (LLL) algorithm can achieve $1/(2\gamma)$-BDD for $\gamma \approx O(2^{n/4})$, yielding a polynomial-complexity decoding algorithm performing exponentially better than Babai's which achieves $\gamma = O(2^{n/2})$. This substantially improves the existing result $\gamma = O(2^n)$ for embedding decoding. They also proved that BDD of the regularized lattice is optimal in terms of the diversity-multiplexing gain tradeoff (DMT).

6.1.5. A New View on HJLS and PSLQ: Sums and Projections of Lattices

The HJLS and PSLQ algorithms are the de facto standards for discovering non-trivial integer relations between a given tuple of real numbers. In [19], J. Chen, D. Stehlé, and G. Villard provided a new interpretation of these algorithms, in a more general and powerful algebraic setup: they view them as special cases of algorithms that compute the intersection between a lattice and a vector subspace. Further, they extracted from them the first algorithm for manipulating finitely generated additive subgroups of a Euclidean space, including projections of lattices and finite sums of lattices. They adapted the analyses of HJLS and PSLQ to derive correctness and convergence guarantees. They also investigated another approach based on embedding the input in a higher dimensional lattice and calling the LLL lattice reduction algorithm.

6.2. Certified computing and computer algebra

6.2.1. Polynomial system solving

Polynomial system solving is a core topic of computer algebra. While the worst-case complexity of this problem is known to be hopelessly large, the practical complexity for large families of systems is much more reasonable. Progress has been made in assessing precise complexity estimates in this area.

First, M. Bardet (U. Rouen), J.-C. Faugère (PolSys team), and B. Salvy studied the complexity of Gröbner bases computations, in particular in the generic situation where the variables are in simultaneous Noether position with respect to the system. They gave a bound on the number of polynomials of each degree in a Gröbner basis computed by Faugère's F5 algorithm in this generic case for the grevlex ordering (which is also a bound on the number of polynomials for a reduced Gröbner basis), and used it to bound the exponent of the complexity of the F5 algorithm [35].

Next, a fundamental problem in computer science is to find all the common zeroes of m quadratic polynomials in n unknowns over F_2. The cryptanalysis of several modern ciphers reduces to this problem. Up to now, the best complexity bound was reached by an exhaustive search in $4\log_2 n\, 2^n$ operations. In [1], M. Bardet (U. Rouen), J.-C. Faugère (PolSys team), B. Salvy, and P.-J. Spaenlehauer (CARAMEL team) 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, they showed that the deterministic variant of their algorithm has complexity bounded by $O(2^{0.841n})$ when $m = n$, while a probabilistic variant of the Las Vegas type has expected complexity $O(2^{0.792n})$. Experiments on random systems showed that the algebraic assumptions are satisfied with probability very close to 1. They have also given a rough estimate for the actual threshold between their method and exhaustive search, which is as low as 200, and thus very relevant for cryptographic applications.

6.2.2. Linear differential equations

Creative telescoping algorithms compute linear differential equations satisfied by multiple integrals with parameters. Together with A. Bostan and P. Lairez (SpecFun team), B. Salvy described a precise and elementary algorithmic version of the Griffiths–Dwork method for the creative telescoping of rational functions. This leads to bounds on the order and degree of the coefficients of the differential equation, and to the first complexity result which is simply exponential in the number of variables. One of the important features of the algorithm is that it does not need to compute certificates. The approach is vindicated by a prototype implementation [15].

In [2], B. Salvy proved with A. Bostan (SpecFun team) and K. Raschel (U. Tours) that the sequence $(e_n^{\mathfrak{S}})_{n\geq 0}$ of excursions in the quarter plane corresponding to a nonsingular step set $\mathfrak{S} \subseteq \{0, \pm 1\}^2$ with infinite group does not satisfy any nontrivial linear recurrence with polynomial coefficients. Accordingly, in those cases, the trivariate generating function of the numbers of walks with given length and prescribed ending point is not D-finite. Moreover, they displayed the asymptotics of $e_n^{\mathfrak{S}}$. This completes the classification of these walks.

With F. Johansson and M. Kauers (RISC, Linz, Austria), M. Mezzarobba presented in [23] a new algorithm for computing hyperexponential solutions of ordinary linear differential equations with polynomial coefficients. The algorithm relies on interpreting formal series solutions at the singular points as analytic functions and evaluating them numerically at some common ordinary point. The numerical data is used to determine a small number of combinations of the formal series that may give rise to hyperexponential solutions.

6.2.3. Exact linear algebra

Transforming a matrix over a field to echelon form, or decomposing the matrix as a product of simpler matrices that reveal the rank profile, is a fundamental building block of computational exact linear algebra. For such tasks the best previously available algorithms were either rank sensitive (i.e., of complexity expressed in terms of the exponent of matrix multiplication and the rank of the input matrix) or in place (i.e., using essentially no more memory that what is needed for matrix multiplication). In [6] C.-P. Jeannerod, C. Pernet, and A. Storjohann (U. Waterloo, Canada) have proposed algorithms that are both rank sensitive and in place. These algorithms required to introduce a matrix factorization of the form $A = CUP$ with C a column echelon form giving the row rank profile of the input matrix A, U a unit upper triangular matrix, and P a permutation matrix.

6.2.4. Certified multiple-precision evaluation of the Airy Ai function

The series expansion at the origin of the Airy function $Ai(x)$ is alternating and hence problematic to evaluate for $x > 0$ due to cancellation. S. Chevillard (APICS team) and M. Mezzarobba showed in [20] how an arbitrary and certified accuracy can be obtained in that case. Based on a method recently proposed by Gawronski, Müller, and Reinhard, they exhibited two functions F and G, both with nonnegative Taylor expansions at the origin, such that $Ai(x) = G(x)/F(x)$. The sums are now well-conditioned, but the Taylor coefficients of G turn out to obey an ill-conditioned three-term recurrence. They then used the classical Miller algorithm to overcome this issue. Finally, they bounded all errors and proposed an implementation which, by allowing an arbitrary and certified accuracy, can be used for example to provide correct rounding in arbitrary precision.

6.2.5. Standardization of interval arithmetic

The IEEE 1788 working group is devoted to the standardization of interval arithmetic. V. Lefèvre and N. Revol are very active in this group. This year is the last year granted by IEEE for the preparation of a draft text of the standard. 2014 will be devoted to a ballot on the whole text, first by the standardization working group and then by a group of experts appointed by IEEE. In 2013, the definition of interval literals, of constructors, and of input and output has been adopted. The work now concentrates on portions of the final text [42].

6.2.6. Parallel product of interval matrices

The problem considered here is the multiplication of two matrices with interval coefficients. Parallel implementations by N. Revol and Ph. Théveny [10] compute results that satisfy the inclusion property, which is the fundamental property of interval arithmetic, and offer good performances: the product of two interval matrices is not slower than 15 times the product of two floating-point matrices.

6.2.7. Numerical reproducibility

What is called *numerical reproducibility* is the problem of getting the same result when the scientific computation is run several times, either on the same machine or on different machines. In [43], the focus is on interval computations using floating-point arithmetic: N. Revol identifies implementation issues that may invalidate the inclusion property, and presents several ways to preserve this inclusion property. This work has also been presented at several conferences [30], [29], [31].

6.3. Floating-point arithmetic

6.3.1. Improved error bounds for complex floating-point arithmetic with a fused-multiply add

Assuming that a fused multiply-add (FMA) instruction is available, C.-P. Jeannerod, N. Louvet, and J.-M. Muller [22] obtained sharp error bounds for various alternatives to Kahan's FMA-based algorithm for 2 x 2 determinants (which they had analyzed in [5]). They showed how to combine such variants with Kahan's original scheme in order to derive componentwise-accurate algorithms for complex floating-point division. Finally, they established sharp or reasonably sharp error bounds for each of these division algorithms.

C.-P. Jeannerod, P. Kornerup (U. of Southern Denmark), N. Louvet, and J.-M. Muller [36] studied the impact of the FMA on the normwise relative accuracy of complex floating-point multiplication. They showed that the classical normwise relative error bound $\sqrt{5}\,u$ (with u the unit roundoff) can be decreased further to $2u$, and that this new constant is best possible for several FMA-based multiplication algorithms.

J.-M. Muller analyzed in [41] another 2 x 2 determinant algorithm, due to Cornea, Harrison, and Tang, and showed that for radix 2 it admits a sharp relative error bound of the form $2u + O(u^2)$.

6.3.2. Improved error bounds for numerical linear algebra

C.-P. Jeannerod and S. M. Rump (Hamburg University of Technology) [7] showed that when evaluating sums of n real numbers in standard floating-point arithmetic, the usual fraction $\gamma_n = nu/(1-nu)$, which has the form $nu + O(u^2)$ and requires $nu < 1$, can be replaced by nu without any restriction on n. Applications include simpler and more general error bounds for inner products, matrix-vector multiplication, and classical matrix multiplication.

In [45] they extended these results to LU and Cholesky factorizations as well as to triangular linear system solving by showing that the constants γ_n that appear classically in the backward error bounds for such problems can all be replaced by $O(u^2)$-free and unconditional constants nu. To get these new bounds the main ingredient is a general framework for bounding expressions of the form $|\rho - s|$, where s is the exact sum of a floating-point number and $n-1$ real numbers, and where ρ is a real number approximating the computed sum $\widehat{s}$.

6.3.3. On Ziv's rounding test

F. de Dinechin, J.-M. Muller and S. Torres studied with C. Lauter (Univ. Paris 6) the rounding test introduced by Ziv in its libultim software [4]. This test determines if an approximation to the value $f(x)$ of an elementary function at a given point x suffices to return the floating-point number nearest to $f(x)$. They showed that the same test may be used for efficient implementation of floating-point operations with input and output operands of different formats. That test depends on a "magic constant" e and they also showed how to choose that constant to make the test reliable and efficient. Various cases are considered, depending on the availability of an FMA instruction, and on the range of $f(x)$.

6.3.4. Various issues related to double roundings

Double rounding is a phenomenon that may occur when different floating-point precisions are available on the same system. Although double rounding is, in general, innocuous, it may change the behavior of some useful floating-point algorithms. G. Melquiond (Toccata team), E. Martin-Dorel (then in the Marelle team), and J.-M. Muller analyzed in [9] the potential influence of double rounding on the Fast2Sum and 2Sum algorithms, on some summation algorithms, and Veltkamp's splitting. When performing divisions using Newton-Raphson (or similar) iterations on a processor with a floating-point fused multiply-add instruction, one must sometimes scale the iterations, to avoid over/underflow and/or loss of accuracy. This may lead to double-roundings, resulting in output values that may not be correctly rounded when the quotient falls in the subnormal range. J.-M. Muller showed in [13] how to avoid this problem.

6.3.5. Comparison between binary and decimal floating-point numbers

The IEEE 754-2008 standard for floating-point arithmetic arithmetic specifies binary as well as decimal formats. N. Brisebarre, C. Lauter (Univ. Paris 6), M. Mezzarobba, and J.-M. Muller introduced in [17] an algorithm that allows one to quickly compare a binary64 floating-point number and a decimal64 floating-point number, assuming the "binary encoding" of the decimal formats specified by the IEEE-754 standard 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.

6.3.6. Conversions between binary and decimal floating-point numbers

Conversion between binary and decimal floating-point representations is ubiquitous. Floating-point radix conversion means converting both the exponent and the mantissa. O. Kupriianova and C. Lauter (Univ. Paris 6) and J.-M. Muller developed in [38] an atomic operation for floating-point radix conversion with simple straight-line algorithm, suitable for hardware design. Exponent conversion is performed with a small multiplication and a lookup table. It yields the correct result without error. Mantissa conversion uses a few multiplications and a small lookup table that is shared amongst all types of conversions. The accuracy changes by adjusting the computing precision.

6.3.7. Table-maker's dilemma

Computing hardest-to-round cases of elementary functions is a key issue when one wants to develop an efficient and reliable implementation of such a function. The algorithms developed until now required a large amount of computation and produced a simple yes/no answer. In [40], G. Hanrot developed together with E. Martin-Dorel (Toccata team), M. Mayero (IUT Villetaneuse, LIPN), and L. Théry (Marelle team) a certificate-based approach of the SLZ algorithm where the execution produces certificates which can then be validated using Coq. This allows one to validate a posteriori the fact that for a given function, a given input precision p and bound p', there is no pair (x, y) of floating-point representable numbers in precision p such that $2^{-e_p(f(x))}|f(x) - y| \leq 2^{-p'}$. This approach has been tested on the exponential function over $[1/2, 1]$, with an input precision of 53 bits and $p' = 300$.

6.4. Hardware and FPGA arithmetic

6.4.1. Reconfiguring arithmetic

With B. Pasca (Altera), F. de Dinechin contributed a book chapter about of the opportunities and challenges of computer arithmetic for reconfigurable/FPGA computing [32]. The main point of this chapter is to look beyond the heritage of processor arithmetic. Using many examples from the FloPoCo project and others, it shows the benefits of merging and fusing standard operators, it introduces an open-ended space of non-standard operators, and illustrates the power of machine-generation of such arithmetic cores.

6.4.2. The bit heap framework for fixed-point arithmetic

N. Brunie, F. de Dinechin, and M. Istoan, with students G. Sergent, K. Illyes, and B. Popa, extended FloPoCo with a versatile framework for manipulating sums of weighted bits [28], [18]. Such bit heaps may be used to express and optimize at the bit level a wide range of operators (from adders and multipliers to polynomials, filters, and other coarse arithmetic cores). A single piece of code can then be used to generate an architecture for any of these operators.

6.4.3. Elementary functions

F. de Dinechin, with P. Echeverria and M. Lopez-Vallejo (U. Madrid) and B. Pasca (Altera), published a hardware architecture for the floating-point pow and powr functions of the IEEE-754-2008 standard [3]. These functions compute x^y, and differ only in the specification of special cases. The implementation, distributed in FloPoCo, is parameterized in exponent and significand size. It combines suitably modified exponential and logarithm units.

F. de Dinechin and M. Istoan, with student G. Sergent, compared several hardware algorithms for the implementation of sine, cosine, and combined sine/cosine [21]: unrolled CORDIC in two variants with several minor improvements, polynomial approximation, and an ad-hoc architecture based on trigonometric identities. A surprising result is that the ad-hoc architecture betters CORDIC even when its multipliers and tables are synthesized as logic.

6.4.4. Contributions to processor architecture

S. Collange (ALF team) and N. Brunie with G. Diamos (Nvidia) suggested improvements for the architecture of general-purpose graphical processing units [11]. As threads take different paths across the control-flow graph, SIMD lockstep execution is partially lost, and must be regained whenever possible in order to maximize the occupancy of SIMD units. Two techniques are described to handle SIMT control divergence and identify reconvergence points. The most advanced one operates in constant space and handles indirect jumps and recursion. In terms of performance, this solution is at least as efficient as state-of-the-art techniques in use in current GPUs.

N. Brunie and F. de Dinechin studied with B. de Dinechin (Kalray) the integration of a tightly coupled reconfigurable accelerator in a massively parallel multiprocessor [27]. For this purpose, they described an architecture exploration framework that produces an architecture along with the relevant compilation software. This framework was demonstrated on AES, SHA2, and a FIR filter.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

7.1.1. Contract with STMicroelectronics

A contract between STMicroelectronics and Inria supported our work on floating-point arithmetic code generation and specialization for embedded processors (duration: 36 months; amount: 36,000 euros; signature: fall 2010). This work, which was done jointly with the Compilation Expertise Center of STMicroelectronics Grenoble, was also supported by the PhD CIFRE grant of Jingyan Jourdan-Lu.

7.1.2. Collaboration with Bosch

Bosch (Stuttgart) ordered us a study on the choice of an adequate representation of numbers (fixed-point or floating-point) for some embedded systems. The study was conducted by Florent de Dinechin and Jean-Michel Muller.

7.1.3. Collaboration with Intel

INTEL made a $20000 donation in recognition of our work on the correct rounding of functions.

7.2. Bilateral Grants with Industry

7.2.1. Kalray CIFRE PhD Grant

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

7.2.2. Orange Labs PhD Grant

Marie Paindavoine is supported by an Orange Labs PhD Grant (from October 2013 to November 2016). She will work on privacy-preserving encryption mechanisms.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR HPAC Project

Participants: Claude-Pierre Jeannerod, Nicolas Louvet, Clément Pernet, 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-based 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, their integration into state-of-the-art software, and innovative high performance solutions for cryptology challenges.

8.1.2. ANR TaMaDi Project

Participants: Nicolas Brisebarre, Florent de Dinechin, Guillaume Hanrot, Vincent Lefèvre, Jean-Michel Muller, Damien Stehlé, Serge Torres.

The TaMaDi project (Table Maker's Dilemma, 2010-2013) was funded by the ANR and headed by Jean-Michel Muller. It started in October 2010 and ended in October 2013. 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 was 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 on the project can be found at http://tamadiwiki.ens-lyon.fr/tamadiwiki/index.php/Main_Page.

8.1.3. PEPS Quarenum

Participants: Nicolas Louvet, Nathalie Revol.

"Quarenum" is an abbreviation for *Qualité et Reproductibilité Numériques dans le Calcul Scientifique Haute Performance*. This project focuses on the numerical quality of scientific software, more precisely of high-performance numerical codes. Numerical validation is one aspect of the project, the second one regards numerical reproducibility.

8.2. European Initiatives

8.2.1. FP7 Projects

Damien Stehlé was awarded in 2013 a "starting" ERC grant for his project "Euclidean lattices: algorithms and cryptography" (LattAC).

8.3. International Initiatives

8.3.1. Inria Associate Teams

QOLAPS (Quantifier elimination, Optimization, Linear Algebra and Polynomial Systems) is an 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: Clément Pernet, Nathalie Revol, Gilles Villard.

8.3.2. Inria International Partners

8.3.2.1. Declared Inria International Partners

We contributed to the creation in 2008 of the IEEE 1788 working group on the standardization of interval arithmetic http://grouper.ieee.org/groups/1788/. and N. Revol chairs this group since its creation. More than 140 persons from over 20 countries take part in the discussions, around 2500 public messages were exchanged in 2013. The deadline granted by IEEE is December 2014. In 2013 we managed to elaborate a close-to-final draft of the standard text. This last year will be devoted to the final ballot from the working group and to a sponsor ballot, by experts designated by IEEE.

The annual in-person meeting, chaired by N. Revol, took place at the end of the IFSA-NAFIPS 2013 conference in Edmonton, Canada, the 25th of June.

V. Lefèvre participated in various discussions, either in the mailing-list or in small subgroups (he sent around 390 email messages in 2013).

8.3.2.2. Informal International Partners

Our international academic collaborators are from Courant Institute of Mathematical Sciences (USA), Hamburg University of Technology (Germany), Imperial College (UK), Macquarie University (Australia), Mc Gill University (Canada), Monash University (Australia), Nanyang Technological University (Singapore), North Carolina State University (USA), Technical University of Cluj-Napoca (Romania), University of California, Los Angeles (USA), University of Delaware (USA), University of Southern Denmark (Denmark), University of Western Ontario (Canada), University of Waterloo (Canada), Uppsala University (Sweden).

We also collaborate with Intel (Portland, USA).

8.3.3. Participation In other International Programs

CANTAL (Cryptography, Algorithmic Number Theory and Lattices) is a CNRS Associate Team between the cryptography group of Macquarie University (Australia), the cryptography group of Monash University (Australia) and the AriC team. Participants: Nicolas Brisebarre, Guillaume Hanrot, Fabien Laguillaumie, Adeline Langlois, Damien Stehlé.

Damien Stehlé is a Partner Investigator in the Australian Research Council Discovery Grant on Cryptography and Algorithmic Number Theory, headed by Christophe Doche (Macquarie U.), Igor Shparlinski (U. of New South Wales), and Ron Steinfeld (U. of Monash), and in a Singaporean Ministry of Education grant of Code-based and Lattice-based cryptography, headed by San Ling (Nanyang Technological U.) and Huaxiong Wang (Nanyang Technological U.).

8.4. International Research Visitors

8.4.1. Visits of International Scientists

Xiao-Wen Chang (McGill U., Canada) visited the team from mid-April to mid-June 2013, under the invited professor scheme from ENS de Lyon.

Warwick Tucker (Uppsala U., Sweden) visited the team from mid-February to the end of March 2013, both under the invited professor scheme from ENS de Lyon and thanks to a funding provided by the LIP laboratory.

Peter Kornerup (U. of Southern Denmark) visited the team the last two weeks of September 2013.

8.4.1.1. Internships

Saruchi (IIT Delhi) did a 3-month Master degree internship under the supervision of Damien Stehlé, from April to June 2013.

8.4.2. Visits to International Teams

Nicolas Brunie was invited for 6 months by Intel (Portland, USA) to work on the implementation of elementary functions.

9. Dissemination

9.1. Scientific Animation

- Florent de Dinechin is an associate editor of the journal *IEEE Transactions on Computers*. He was a member of the Program Committees of the conferences CompAs (Grenoble, January 2013), Applied Reconfigurable Computing (Los Angeles, March 2013), Highly Efficient Accelerators and Reconfigurable Technologies (Edimburgh, June 2013), Field-Programmable Logic (Porto, September 2013), Field-Programmable Technology (Kyoto, December 2013), ReConfig 2013 (Cancun, December 2013). He also organized a tutorial half-day on arithmetic core generation using the FloPoCo framework at HiPEAC 2013 (Berlin, January 2013).
- Guillaume Hanrot has been deputy director of the LIP (laboratoire d'informatique du parallélisme) since 01/01/13. He has also been in charge of the computer science master at ENS de Lyon for the academic year 2012-2013. He has been a member of hiring committees for an assistant professor position at Caen IUT, for an assistant professor position at Saint-Étienne IUT, for a professor position at ENSIIE Strasbourg, and of the national committee for "Prime d'excellence scientifique". He is a member of the scientific council of ENSIIE (Évry). He chairs a working group in charge of making recommendations concerning general teaching and training policy at ENS de Lyon.
- Claude-Pierre Jeannerod is a member of the scientific committee of "Journées Nationales de Calcul Formel".
- Fabien Laguillaumie has been member of an hiring committee for an assistant professor position at Université Lyon 1. He was member of ProvSec 2013 program committee. He is responsible for the second year "Ingénierie des Risques" of the Master SAFIR.
- Vincent Lefèvre and Nicolas Louvet were in the scientific committee of the CNRS thematic school *Précision et reproductibilité en calcul numérique* (Fréjus, France, March, 2013).
- Jean-Michel Muller co-chairs the Groupement de Recherche (GDR) *Informatique Mathématique* of CNRS. He is an associate editor of the journal *IEEE Transactions on Computers*. He participated to the evaluation committee of the LIRMM laboratory (Montpellier) in November 2013. He is a member of the scientific councils of CERFACS (Toulouse) and ENS de Lyon. He was a member of the Program Committees of the conferences IEEE ARITH 21 (Austin, Texas, April 2013) and IEEE ASAP'2013 (Washington DC, June 2013).
- Nathalie Revol was in the hiring committee for junior researchers (CR) of Inria Grenoble - Rhône-Alpes. She is a member of the CES (Commission des Emplois Scientifiques), the hiring committee for postdocs at Inria Grenoble - Rhône-Alpes. She was the chair of the organization committee and took charge of the "gender aspects" of the Forum 2013 des Jeunes Mathématicien-ne-s, 13-15 November, Lyon. She is a member of the "comité de diffusion" of the MILyon labex. She belongs to the steering committee for the MMI (Maison des Mathématiques et de l'Informatique). She is a member of the selecting committee of CapMaths.
- Bruno Salvy is a member of the editorial boards of the "Journal of Symbolic Computation", of the "Journal of Algebra" (section Computational Algebra) and of the collections "Texts and Monographs in Symbolic Computation" (Springer) and "Mathématiques et Applications" (SMAI-Springer). He is organizing the working group Computer Algebra of the CNRS GDR IM. This year, he has been in the program committees of ISSAC 2013 (Boston, Mass., June 2013) and Analco 2013 (New Orleans, Louisiana, January 2013).
- Damien Stehlé has been deputy director and Erasmus coordinator of the ENS de Lyon Computer Science department from 01/01/13 until 30/06/13. He has been the director of the Computer Science department since 01/07/13. He is a member of the steering committee of the Cryptography and Coding CNRS working group (GdT C2 du GDR IM). He is a member of the steering committee of the PQCrypto conference series. In 2013, he served in the program committees of the PQCrypto (Limoges, France, June 2013) and Asiacrypt (Bengalore, India, December 2013) conferences.

- Gilles Villard is chair of LIP laboratory and a member of the editorial board of the "Journal of Symbolic Computation."

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Licence : Nicolas Louvet, *Computer Architecture*, 74h, L2, Univ. Lyon 1.

Licence : Nicolas Louvet, *Algorithms and Data Structures*, 30h, L3, Univ. Lyon 1.

Licence : Nicolas Louvet, *Operating Systems*, 50h, L3, Univ. Lyon 1.

Master: Nicolas Brisebarre and Bruno Salvy, *Approximations: from symbolic to numerical computation, and applications*, 24h, ENS de Lyon.

Master: Guillaume Hanrot and Jean-Michel Muller, *Computer Algebra*, 24h, ENS de Lyon.

Master: Claude-Pierre Jeannerod, Nicolas Louvet, Nathalie Revol, *Numerical Algorithms*, 48h, Univ. Lyon 1.

Master: Fabien Laguillaumie, *Cryptography, Error Correcting Codes*, 150h, Univ. Lyon 1.

Master: Vincent Lefèvre, *Computer Arithmetic*, 14h, Univ. Lyon 1.

Master: Jean-Michel Muller, *Floating-Point Arithmetic and Formal Proof* (8h + coordination of the 24h course), ENS de Lyon.

Master: Bruno Salvy, *Computer Algebra*, 12h, MPRI.

Master: Damien Stehlé, *Cryptography*, 24h, ENS de Lyon.

Doctorat: Nicolas Brisebarre, *Lattices in computer arithmetic*, 3h, École de Printemps d'Informatique Théorique, Autrans, March 21.

Doctorat: Florent de Dinechin, *Arithmétique flottante*, 1h30, École CNRS *Précision et reproductibilité en calcul numérique*, Fréjus, March 25-29.

Doctorat: Guillaume Hanrot, *Introduction to lattice algorithms*, 3h, École de Printemps d'Informatique Théorique, Autrans, March 18.

Doctorat: Vincent Lefèvre, *Arithmétique flottante en précision arbitraire*, 3h, École CNRS *Précision et reproductibilité en calcul numérique*, Fréjus, March 25-29.

Doctorat: Jean-Michel Muller, *Arithmétique flottante*, 2h, École CNRS *Précision et reproductibilité en calcul numérique*, Fréjus, March 25-29.

Doctorat: Jean-Michel Muller, *Arithmétique flottante*, 2h, École HPC, Lyon, September 3.

Doctorat: Nathalie Revol and Philippe Théveny, *Arithmétique flottante et intervalles*, 3h, École CNRS *Précision et reproductibilité en calcul numérique*, Fréjus, March 25-29.

Doctorat: Nathalie Revol and Philippe Théveny, *Précision et arithmétique flottante : outils, bibliothèques*, 3h30, JDEV : Journées du Développement Logiciel, École Polytechnique, 4-6 September.

Doctorat: Nathalie Revol and Philippe Théveny, *Introduction à l'arithmétique par intervalles*, 3h, professional training entitled "Contrôler et améliorer la qualité numérique d'un code de calcul industriel", Collège de l'X, Paris, 21-22 November.

Doctorat: Damien Stehlé, *Introduction to lattices*, 3h, École de Printemps d'Informatique Théorique, Autrans, March 18.

9.2.2. Supervision

PhD: Julien Devigne, *Protocoles de re-chiffrement pour le stockage de données*, September 2011 - December 2013 (Orange Labs - Univ. Caen); co-supervised by Fabien Laguillaumie (together with Sébastien Canard and Brigitte Vallée).

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

PhD in progress: Louis Dumont, *Algorithmique efficace pour les diagonales*, since September 2013, co-supervised by Bruno Salvy (together with Alin Bostan).

PhD in progress: Silviu Filip, *Filtroptim : tools for an optimal synthesis of numerical filters*, since September 2013, co-supervised by Nicolas Brisebarre and Guillaume Hanrot.

PhD in progress: Pierre Lairez, *Algorithmique efficace pour la création télescopique et ses applications*, since September 2011, co-supervised by Bruno Salvy (together with Alin Bostan).

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

PhD in progress: Vincent Neiger, *Multivariate interpolation in computer algebra: efficient algorithms ans applications*, since September 2013, co-supervised by Claude-Pierre Jeannerod and Gilles Villard (together with Éric Schost).

PhD in progress: Marie Paindavoine, *Méthodes de calculs sur des données chiffrées*, since October 2013 (Orange Labs - UCBL), co-supervised by Fabien Laguillaumie (together with Sébastien Canard).

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

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

9.2.3. Juries

- Nicolas Brisebarre was a member of the PhD committee of Răzvan Bărbulescu (Nancy, December 2013).
- Guillaume Hanrot was an external referee for the PhD of Md Mohammed Haque (Macquarie U., Australia, sept. 2013) and for the PhD of Mariya Georgieva (Caen, 2013-12-09). He was a member of the committee for the PhD of Léo Ducas (ENS Paris, 2013-11-12) and Yuanmi Chen (ENS Paris, 2013-11-13).
- Fabien Laguillaumie was an external referee for the PhD of Léo Ducas (ENS Paris, 2013-11-12) and Viet Cuong Trinh (Univ. Paris 8, 2013-12-19). He was a member of the PhD committees of Mario Stefler (2013-09-26, ENS Paris), Nicolas Estibals (2013-10-30), Mariya Georgieva (Caen, 2013-12-09), and Aurore Guillevic (ENS Paris, 2013-12-20)
- Jean-Michel Muller was a referee for the habilitation of Stef Graillat (U. Paris 6, 2013-12-2).
- Nathalie Revol was the external referee for the PhD of Bingzhou Zheng (U. McMaster, Hamilton, Canada, 2013-12-10).
- Bruno Salvy was a member of the PhD committee of Fabien Monfreda (Toulouse, July 2013).
- Damien Stehlé was an external referee for the PhD of Romar Basillaje Dela Cruz (Nanyang Technological U., Singapore, March 2013) and of Zhenfei Zhang (U. of Wollongong, Australia, October 2013).

9.3. Invited Conferences

- Florent de Dinechin gave invited talks or lectures at the CERN/Intel OpenLab workshop at CERN, at the LIF seminar in Luminy, at the Intel Summer School in Nizhniy Novgorod, and at the Journées Développement Logiciel (JDEV) in Palaiseau.
- Jean-Michel Muller gave an invited talk *Proof of Properties in Floating-Point Arithmetic* at the conference *Continuity, Computability, Constructivity - From Logic to Algorithms* (CCC 2013), Swansea University/Gregynog, UK, June 26–30, 2013.

- Nathalie Revol gave talks about numerical reproducibility, with a focus on interval computations, at PPAM'2013 (Warsaw, Poland, 2013/09/8-11), at RAIM 2013 (Paris, France, 2013/11/18-20), at McMaster University (Hamilton, Ontario, Canada, 2013-12-10), and at University of Toronto (Ontario, Canada, 2013-12-13).
- Bruno Salvy was invited to give a talk *Implicit Species at the basis of Analytic Combinatorics* at the 24th International Meeting on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithm (AofA 2013, Menorca, Spain, May 27–31) and on *Newton iteration in computer algebra and combinatorics* at the *Journées d'Informatique Fondamentale de Paris Diderot* (April 22–26).
- Damien Stehlé gave an invited lecture talk at *Journées Nationales du Calcul Formel* (Luminy, France, May 13–17, 2013), invited plenary talks at the conferences *SIAM Conference on Applied Algebraic Geometry (AG13)* (Fort Collins, Colorado, USA, August 1–4, 2013) and *ICISC* (Seoul, Korea, November 25–29, 2013) and at the *Microsoft Research India Workshop on Lattice-Based Cryptography* (Bengalore, India, November 30, 2013).

9.4. Popularization

- Nathalie Revol gave talks for pupils at collèges and lycées, as an incentive to choose scientific careers: lycée Jeanne d'Arc (Cessy, Ain), lycée Rosa Parks (Neuville, Rhône). During the "Week of mathematics", she gave a 2-hour talk at lycée de la Plaine de l'Ain (Ambérieu-en-Bugey, Ain). She was present, took part in speed-meetings, and gave talks for the "Mondial des Métiers" (Eurexpo Lyon, Chassieu, Rhône) and for "Science au Carré(e)" (Forum des Halles, Paris). For the Science Fair, she gave 8 talks at ENS de Lyon. She was invited to "Interacadémiques" in Lyon, for an audience of inspecteurs d'académie. She supervised the internship of Quentin Chopinet (1e S, one week) and hosted Elsa Courtais (spé TSI, one day).
- Damien Stehlé was interviewed for an article in *La Recherche*, published in September 2013.

10. Bibliography

Publications of the year

Articles in International Peer-Reviewed Journals

[1] M. BARDET, J.-C. FAUGÈRE, B. SALVY, P.-J. SPAENLEHAUER. *On the Complexity of Solving Quadratic Boolean Systems*, in "Journal of Complexity", February 2013, vol. 29, n° 1, pp. 53-75 [*DOI :* 10.1016/J.JCO.2012.07.001], http://hal.inria.fr/hal-00655745

[2] A. BOSTAN, K. RASCHEL, B. SALVY. *Non-D-finite excursions in the quarter plane*, in "Journal of Combinatorial Theory, Series A", October 2013, vol. 121, pp. 45-63 [*DOI :* 10.1016/J.JCTA.2013.09.005], http://hal.inria.fr/hal-00697386

[3] F. DE DINECHIN, P. ECHEVERRIA, M. LOPEZ-VALLEJO, B. PASCA. *Floating-Point Exponentiation Units for Reconfigurable Computing*, in "ACM Transactions on Reconfigurable Technology and Systems", May 2013, vol. 6, n° 1, 4:1 p. , To appear [*DOI :* 10.1145/2457443.2457447], http://hal.inria.fr/ensl-00718637

[4] F. DE DINECHIN, C. LAUTER, J.-M. MULLER, S. TORRES. *On Ziv's rounding test*, in "ACM Transactions on Mathematical Software", 2013, vol. 39, n° 4, 26 p. , http://hal.inria.fr/ensl-00693317

[5] C.-P. JEANNEROD, N. LOUVET, J.-M. MULLER. *Further analysis of Kahan's algorithm for the accurate computation of 2 x 2 determinants*, in "Mathematics of Computation", 2013, vol. 82, n° 284, pp. 2245-2264 [*DOI :* 10.1090/S0025-5718-2013-02679-8], http://hal.inria.fr/ensl-00649347

[6] C.-P. JEANNEROD, C. PERNET, A. STORJOHANN. *Rank-profile revealing Gaussian elimination and the CUP matrix decomposition*, in "Journal of Symbolic Computation", 2013, vol. 56, pp. 46-68 [*DOI :* 10.1016/J.JSC.2013.04.004], http://hal.inria.fr/hal-00841300

[7] C.-P. JEANNEROD, S. M. RUMP. *Improved error bounds for inner products in floating-point arithmetic*, in "SIAM Journal on Matrix Analysis and Applications", April 2013, vol. 34, n^{o} 2, pp. 338-344 [*DOI :* 10.1137/120894488], http://hal.inria.fr/hal-00840926

[8] C. LING, L. LUZZI, D. STEHLÉ. *Decoding by Embedding: Correct Decoding Radius and DMT Optimality*, in "IEEE Transactions on Information Theory", 2013, vol. 59, n^{o} 5, pp. 2960-2973 [*DOI :* 10.1109/TIT.2012.2236144], http://hal.inria.fr/hal-00922195

[9] É. MARTIN-DOREL, G. MELQUIOND, J.-M. MULLER. *Some issues related to double roundings*, in "BIT Numerical Mathematics", December 2013, vol. 53, n^{o} 4, pp. 897-924 [*DOI :* 10.1007/S10543-013-0436-2], http://hal.inria.fr/ensl-00644408

[10] N. REVOL, P. THÉVENY. *Parallel Implementation of Interval Matrix Multiplication*, in "Reliable Computing", 2013, vol. 19, n^{o} 1, pp. 91-106, http://hal.inria.fr/hal-00801890

Articles in National Peer-Reviewed Journals

[11] N. BRUNIE, S. COLLANGE. *Reconvergence de contrôle implicite pour les architectures SIMT*, in "Technique et Science Informatiques (TSI)", February 2013, vol. 32, n^{o} 2, pp. 153-178 [*DOI :* 10.3166/TSI.32.153-178], http://hal.inria.fr/hal-00787749

Articles in Non Peer-Reviewed Journals

[12] F. LANGROGNET, F. JÉZÉQUEL, N. REVOL. *JDEV 2013 : Développer pour Calculer : Des outils pour calculer avec précision & Comment calculer avec des intervalles*, in "HPC magazine", November 2013, http://hal.inria.fr/hal-00921492

Invited Conferences

[13] J.-M. MULLER. *Avoiding double roundings in scaled Newton-Raphson division*, in "Asilomar Conference on Signals, Systems, and Computers", Pacific Grove, CA, United States, 2013, 4 p. , http://hal.inria.fr/ensl-00875366

International Conferences with Proceedings

[14] R. BHATTACHARYYA, A. ROY. *Secure Message Authentication against Related Key Attack*, in "Fast Software Encryption", Singapore, 2013, pp. 245-262, http://hal.inria.fr/hal-00920172

[15] A. BOSTAN, P. LAIREZ, B. SALVY. *Creative telescoping for rational functions using the Griffiths-Dwork method*, in "ISSAC'13 - 38th International Symposium on Symbolic and Algebraic Computation", Boston, United States, Northeastern University, Boston, Massachusetts, USA, 2013, pp. 93-100 [*DOI :* 10.1145/2465506.2465935], http://hal.inria.fr/hal-00777675

[16] Z. BRAKERSKI, A. LANGLOIS, C. PEIKERT, O. REGEV, D. STEHLÉ. *Classical Hardness of Learning with Errors*, in "Proceedings of STOC", United States, 2013, pp. 575-584, http://hal.inria.fr/hal-00922194

[17] 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, A. NANNARELLI, P.-M. SEIDEL, P. T. P. TANG (editors), IEEE Computer Society, April 2013, 8 p. [*DOI :* 10.1109/ARITH.2013.23], http://hal.inria.fr/ensl-00737881

[18] N. BRUNIE, F. DE DINECHIN, M. ISTOAN, G. SERGENT, K. ILLYES, B. POPA. *Arithmetic core generation using bit heaps*, in "23rd International Conference on Field Programmable Logic and Applications", Porto, Portugal, September 2013, http://hal.inria.fr/ensl-00738412

[19] J. CHEN, D. STEHLÉ, G. VILLARD. *A New View on HJLS and PSLQ: Sums and Projections of Lattices*, in "Proceedings of ISSAC", United States, 2013, pp. 149-156, http://hal.inria.fr/hal-00922192

[20] 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, A. NANNARELLI, P.-M. SEIDEL, P. T. P. TANG (editors), 2013, pp. 175-182 [*DOI :* 10.1109/ARITH.2013.33], http://hal.inria.fr/hal-00767085

[21] F. DE DINECHIN, M. ISTOAN, G. SERGENT. *Fixed-Point Trigonometric Functions on FPGAs*, in "Fourth International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies", Edimburgh, United Kingdom, June 2013, http://hal.inria.fr/ensl-00802777

[22] 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 p. , http://hal.inria.fr/ensl-00734339

[23] F. JOHANSSON, M. KAUERS, M. MEZZAROBBA. *Finding Hyperexponential Solutions of Linear ODEs by Numerical Evaluation*, in "ISSAC - 28th International Symposium on Symbolic and Algebraic Computation", Boston, Massachusetts, United States, 2013, http://hal.inria.fr/hal-00818789

[24] F. LAGUILLAUMIE, A. LANGLOIS, B. LIBERT, D. STEHLÉ. *Lattice-Based Group Signatures with Logarithmic Signature Size*, in "ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security", Bangaluru, India, K. SAKO, P. SARKAR (editors), LNCS, 2013, vol. 8270 [*DOI :* 10.1007/978-3-642-42045-0_3], http://hal.inria.fr/hal-00920420

[25] V. LEFÈVRE. *SIPE: Small Integer Plus Exponent*, in "21th IEEE Symposium on Computer Arithmetic - Arith'21", Austin, Texas, United States, 2013, http://hal.inria.fr/hal-00763954

[26] 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, pp. 107–124, http://hal.inria.fr/hal-00767548

National Conferences with Proceedings

[27] 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

[28] 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

[29] F. JÉZÉQUEL, P. LANGLOIS, N. REVOL. *First steps towards more numerical reproducibility*, in "SMAI'2013 : 6ème biennale des Mathématiques Appliquées et Industrielles", Seignosse, France, ESAIM: Proceedings, October 2013, pp. 001-010, http://hal.inria.fr/lirmm-00872562

Conferences without Proceedings

[30] N. REVOL, P. THÉVENY. *Numerical reproducibility in HPC: issues in interval arithmetic*, in "SWIM'2013: Small Workshop on Interval Methods", Brest, France, L. JAULIN, N. RAMDANI (editors), June 2013, http://hal.inria.fr/hal-00922114

[31] N. REVOL, P. THÉVENY. *Numerical reproducibility in HPC: the interval point of view*, in "PPAM'2013: 10th International Conference on Parallel Processing and Applied Mathematics", Warsaw, Poland, September 2013, http://hal.inria.fr/hal-00922117

Scientific Books (or Scientific Book chapters)

[32] 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

[33] R. BHATTACHARYYA. , *Non-Adaptive Programmability of Random Oracles*, December 2013, Submitted, http://hal.inria.fr/hal-00920173

[34] R. BHATTACHARYYA, S. CHAKRABORTY. , *Constant Query Locally Decodable Codes against Computationally Bounded Adversary*, December 2013, Submitted, http://hal.inria.fr/hal-00920174

Other Publications

[35] M. BARDET, J.-C. FAUGÈRE, B. SALVY. , *On the Complexity of the F5 Gröbner basis Algorithm*, December 2013, 20 p. , http://hal.inria.fr/hal-00915522

[36] C.-P. JEANNEROD, P. KORNERUP, N. LOUVET, J.-M. MULLER. , *Error bounds on complex floating-point multiplication with an FMA*, December 2013, http://hal.inria.fr/hal-00867040

[37] C.-P. JEANNEROD, S. M. RUMP. , *On relative errors of floating-point operations: optimal bounds and applications*, January 2014, http://hal.inria.fr/hal-00934443

[38] O. KUPRIIANOVA, C. LAUTER, J.-M. MULLER. , *Radix Conversion for IEEE754-2008 Mixed Radix Floating-Point Arithmetic*, 2013, http://hal.inria.fr/ensl-00916532

[39] V. LEFÈVRE. , *Sipe: a Mini-Library for Very Low Precision Computations with Correct Rounding*, September 2013, http://hal.inria.fr/hal-00864580

[40] É. MARTIN-DOREL, G. HANROT, M. MAYERO, L. THÉRY. , *Formally verified certificate checkers for hardest-to-round computation*, December 2013, http://hal.inria.fr/hal-00919498

[41] J.-M. MULLER. , *On the error of Computing ab + cd using Cornea, Harrison and Tang's method*, 2013, Submitted to ACM TOMS, http://hal.inria.fr/ensl-00862910

[42] N. REVOL. , *Latest Developments on the IEEE 1788 Effort for the Standardization of Interval Arithmetic*, 2013, standardization effort supported by the Inria D2T – submitted to ASCE-ICVRAM 2014 conference, http://hal.inria.fr/hal-00920662

[43] N. REVOL, P. THÉVENY. , *Numerical Reproducibility and Parallel Computations: Issues for Interval Algorithms*, 2013, submitted to IEEE Transactions on Computers, http://hal.inria.fr/hal-00916931

[44] N. REVOL, P. THÉVENY. , *Numerical Reproducibility and Parallel Computations: Issues for Interval Algorithms*, July 2013, http://hal.inria.fr/hal-00845839

[45] S. M. RUMP, C.-P. JEANNEROD. , *Improved backward error bounds for LU and Cholesky factorizations*, July 2013, http://hal.inria.fr/hal-00841361