Efficiently and reliably delegating computations.

Scientific computing, exact computations, outsourcing.

Contact: Jean-Guillaume.Dumas@imag.fr, tel: 04 76 51 48 66.

Laboratoire: Jean Kuntzmann (LJK), tour IRMA, BP 53 X
51, av. des Mathématiques, 38041 Grenoble. ljk.imag.fr

Earnings: Standard (via NSF-ANR project HPAC, hpac.gforge.inria.fr).

PhD follow-up: In collaboration with North Carolina State Univ., Univ. of Delaware.

In an emerging computing paradigm, computational capabilities, from processing power to storage capacities, are offered to users over communication networks as a service.

This new paradigm holds enormous promise for increasing the utility of computationally weak devices. A natural approach is for weak devices to delegate expensive tasks, such as storing a large file or running a complex computation, to more powerful entities (say servers) connected to the same network. While the delegation approach seems promising, it raises an immediate concern: when and how can a weak device verify that a computational task was completed correctly? This practically motivated question touches on foundational questions in cryptography, coding theory, complexity theory, proofs and algorithms.

More generally, the question of verifying a result at a lower cost (time, memory) than that of recomputing it, is of paramount importance. Another example of application is for the extension of the trust about results computed via probabilistic or approximate algorithms. There the idea is to gain confidence into the correctness, but only at a cost negligible when compared to that of the computation.

On the one hand, efficient protocols (interactive proofs between a prover, responsible for the computation, and a verifier, to be convinced) can be designed for delegating computational tasks. In the last 5 years, generic protocols have been designed by teams around Shafi Goldwasser at the MIT or Harvard, for circuits with polylogarithmic depth [7, 11], namely for problems that can be efficiently solved on a parallel computer (in the NC or AC complexity class). The resulting protocols are interactive and their cost for the verifier is usually only roughly proportional to the input size. They however can produce a non negligible overhead for the prover and are restricted to certain classes of circuits. A similar approach as been developed by Gentry et al., it is is Pinocchio. It solves a broader range of problems, to the cost of using relatively inefficient homomorphic routines [10].

On the other hand, dedicated certificates (data structures and algorithms that are verifiable a posteriori, without interaction) have also been developed in the last few years, e.g., for exact linear algebra [6, 9, 5], even for problems that are not in NC. There the certificate constitute a proof of correctness of a result, not of a computation, and can thus also stand for later on verification purpose. For some specific routines, such as the rank of matrices, the obtained certificates are essentially optimal but achieve this to the price of being subject to standard cryptographic hardness assumptions [3].

In general, the main difficulty is to be able to design verification algorithms for a problem that are completely orthogonal to the computational algorithms solving it, while remaining checkable in time and space not much larger than the input.

Therefore, the focus of this thesis is verifying the correctness of outsourced computations, for problems in computer algebra, whether by delegation or via external, probabilistic, software.

Two examples of open problems in computer algebra that could be addressed by this thesis are:
• Given a sparse matrix, provide an algorithm to verify its determinant (or any other linear algebra result), in time linear in the number of non-zero elements of the matrix. We currently know how to this only for the rank and for system solving.

• Remove the cryptographic computational hardness assumption for essentially optimal a posteriori certificates, both in theory and for practical verifiers. In linear algebra we currently know how to this only for the rank and the determinant of dense matrices, and only in terms of theoretical complexity.

The tools used to provide efficient certificates will stem from programs that check their work [4], to proof of knowledge protocols [2], via error-correcting codes [8] and complexity theory [1]. The interaction of these different methodologies is crucial: novel ideas will arise from parallel programming and error-correcting codes on the one hand and computer algebra, complexity and cryptography on the other hand.

More generally, the thesis could also explore the links between interactive proofs and parallel programs, both in terms of theoretical and practical complexity; as well as the links between certification of correctness and error-correcting codes; or the extension of the protocols to memory delegation, where privacy of the data and homomorphic cryptography comes into play.

References


