

RESEARCH CENTRE
Saclay - Île-de-France

2021
ACTIVITY REPORT

Project-Team
SPECFUN

**Symbolic Special Functions : Fast and
Certified**

DOMAIN

Algorithmics, Programming, Software
and Architecture

THEME

Algorithmics, Computer Algebra and
Cryptology

Contents

Project-Team SPECFUN	1
1 Team members, visitors, external collaborators	2
2 Overall objectives	2
2.1 Scientific challenges, expected impact	2
2.2 Use computer algebra but convince users beyond reasonable doubt	3
2.3 Make computer algebra and formal proofs help one another	4
2.4 Experimental mathematics with special functions	4
2.5 Research axes	4
2.6 Computer algebra certified by the Coq system	5
2.6.1 Libraries of formalized mathematics	5
2.6.2 Manipulation of large algebraic data in a proof assistant	5
2.6.3 Formal-proof-producing normalization algorithms	5
2.7 Better symbolic computations with special functions	6
2.7.1 Special-function integration and summation	6
2.7.2 Applications to experimental mathematics	6
2.8 Interactive and certified mathematical web sites	6
3 Research program	7
3.1 Studying special functions by computer algebra	7
3.2 Equations as a data structure	7
3.3 Algorithms combining functions	7
3.4 Solving functional equations	8
3.5 Multi-precision numerical evaluation	8
3.6 Guessing heuristics	8
3.7 Complexity-driven design of algorithms	8
3.8 Encyclopedias	8
3.9 Computer algebra and symbolic logic	9
3.10 Certifying systems for computer algebra	9
3.11 Semantics for computer algebra	9
3.12 Formal proofs for symbolic components of computer-algebra systems	9
3.13 Formal proofs for numerical components of computer-algebra systems	9
3.14 Machine-checked proofs of formalized mathematics	9
3.15 Logical foundations and proof assistants	10
3.16 Computations in formal proofs	10
3.17 Large-scale computations for proofs inside the Coq system	10
3.18 Relevant contributions from the Mathematical Component libraries	10
3.19 User interaction with the proof assistant	11
4 Application domains	11
4.1 Computer Algebra in Mathematics	11
5 Highlights of the year	11
5.1 Refounding the team: towards computer algebra, experimental mathematics, and interactions	11
5.2 ERC project: 10,000 DIGITS	12
5.3 David P. Robbins Prize	12
5.4 Promotion	12

6	New results	12
6.1	Algebraic algorithms on fundamental objects	12
6.1.1	The art of algorithmic guessing in gfun	12
6.1.2	A Sage package for the symbolic-numeric factorization of linear differential operators	12
6.1.3	A simple and fast algorithm for computing the N -th term of a linearly recurrent sequence	12
6.1.4	Fast computation of the N -th term of a q -holonomic sequence and applications	12
6.1.5	Improved algorithms for left factorial residues	13
6.1.6	Explicit degree bounds for right factors of linear differential operators	13
6.2	Polynomial systems and geometry	13
6.2.1	Gröbner bases and critical values: the asymptotic combinatorics of determinantal systems	13
6.2.2	Computing the dimension of real algebraic sets	14
6.2.3	An algorithmic approach to Rupert's problem	14
6.3	Applications to special functions and number theory	14
6.3.1	A hypergeometric proof that Iso is bijective	14
6.3.2	On an integral identity	14
6.3.3	A short proof of a non-vanishing result by Conca, Krattenthaler and Watanabe	14
6.4	Applications to combinatorics	15
6.4.1	Counting walks with large steps in an orthant	15
6.4.2	The generating function of Kreweras walks with interacting boundaries is not algebraic	15
6.4.3	Random walks in orthants and lattice path combinatorics	15
6.4.4	On some combinatorial sequences associated to invariant theory	15
6.4.5	Computer algebra in the service of enumerative combinatorics	16
6.5	Applications to probability	16
6.5.1	Martin boundary of killed random walks on isoradial graphs	16
6.5.2	Genus and classification of random walks in the quarter plane	16
6.5.3	Reflected Brownian motion in a nonconvex cone	16
6.5.4	Persistence probabilities and Mallows–Riordan polynomials	17
6.6	Diagonals	17
6.6.1	On the q -analogue of Pólya's Theorem	17
6.6.2	Diagonal representation of algebraic power series: a glimpse behind the scenes	17
6.6.3	On a class of hypergeometric diagonals	17
6.7	Proceedings of a conference on our topics	18
6.7.1	Transcendence in Algebra, Combinatorics, Geometry and Number Theory	18
7	Partnerships and cooperations	18
7.1	International initiatives	18
7.1.1	Participation in other International Programs	18
7.2	European initiatives	18
7.2.1	Horizon Europe	18
7.3	National initiatives	18
7.3.1	ANR	18
7.4	Regional initiatives	19
8	Dissemination	19
8.1	Promoting scientific activities	19
8.1.1	Scientific events: organisation	19
8.1.2	Scientific events: selection	19
8.1.3	Journal	20
8.1.4	Invited talks	20
8.1.5	Scientific expertise	21
8.1.6	Research administration	21
8.2	Teaching - Supervision - Juries	21
8.2.1	Teaching	21

8.2.2	Supervision	21
8.2.3	Juries	22
9	Scientific production	22
9.1	Major publications	22
9.2	Publications of the year	23
9.3	Cited publications	24

Project-Team SPECFUN

Creation of the Project-Team: 2014 July 01

Keywords

Computer sciences and digital sciences

A8.1. – Discrete mathematics, combinatorics

A8.3. – Geometry, Topology

A8.4. – Computer Algebra

A8.5. – Number theory

Other research topics and application domains

B9.5.2. – Mathematics

B9.5.3. – Physics

1 Team members, visitors, external collaborators

Research Scientists

- Frédéric Chyzak [Team leader, Inria, Researcher, HDR]
- Alin Bostan [Inria, Researcher, HDR]
- Guy Fayolle [Inria, Emeritus]
- Pierre Lairez [Inria, Researcher]

PhD Students

- Alexandre Goyer [Inria]
- Hadrien Notarantonio [Univ Paris-Saclay, from Oct 2021]
- Raphael Pages [Univ de Bordeaux]
- Eric Pichon-Pharabod [Univ Paris-Saclay, from Sep 2021]
- Sergey Yurkevich [Université de Vienne - Autriche, from Sep 2021]

Interns and Apprentices

- Hadrien Notarantonio [Inria, from Apr 2021 until Aug 2021]
- Eric Pichon-Pharabod [Inria, from Apr 2021 until Aug 2021]

Administrative Assistant

- Bahar Carabetta [Inria]

External Collaborators

- Philippe Dumas [Ministère de l'Éducation Nationale]
- Marc Mezzarobba [CNRS, until Sep 2021]
- Pierre Vanhove [CEA, from Sep 2021, HDR]

2 Overall objectives

2.1 Scientific challenges, expected impact

The general orientation of our team is described by the short name given to it: *Special Functions*, that is, particular mathematical functions that have established names due to their importance in mathematical analysis, physics, and other application domains. Indeed, we ambition to study special functions with the computer, by combined means of computer algebra and formal methods.

Computer-algebra systems have been advertised for decades as software for “doing mathematics by computer” [88]. For instance, computer-algebra libraries can uniformly generate a corpus of mathematical properties about special functions, so as to display them on an interactive website. This possibility was recently shown by the computer-algebra component of the team [41]. Such an automated generation significantly increases the reliability of the mathematical corpus, in comparison to the content of existing static authoritative handbooks. The importance of the validity of these contents can be measured by the very wide audience that such handbooks have had, to the point that a book like [38] remains one of the most cited mathematical publications ever and has motivated the 10-year-long project of writing its successor [78]. However, can the mathematics produced “by computer” be considered as *true* mathematics?

More specifically, whereas it is nowadays well established that the computer helps in discovering and observing new mathematical phenomena, can the mathematical statements produced with the aid of the computer and the mathematical results computed by it be accepted as valid mathematics, that is, as having the status of mathematical *proofs*? Beyond the reported weaknesses or controversial design choices of mainstream computer-algebra systems, the issue is more of an epistemological nature. It will not find its solution even in the advent of the ultimate computer-algebra system: the social process of peer-reviewing just falls short of evaluating the results produced by computers, as reported by Th. Hales [65] after the publication of his proof of the Kepler Conjecture about sphere packing.

A natural answer to this deadlock is to move to an alternative kind of mathematical software and to use a proof assistant to check the correctness of the desired properties or formulas. The success of large-scale formalization projects, like the Four-Color Theorem of graph theory [60], the above-mentioned Kepler Conjecture [65], and the **Odd Order Theorem** of group theory, have increased the understanding of the appropriate software-engineering methods for this peculiar kind of programming. For computer algebra, this legitimates a move to proof assistants now.

The **Dynamic Dictionary of Mathematical Functions** (DDMF) [41] is an online computer-generated handbook of mathematical functions that ambitions to serve as a reference for a broad range of applications. This software was developed by the computer-algebra component of the team as a **project** of the MSR–INRIA Joint Centre. It bases on a library for the computer-algebra system Maple, **Algolib**, whose development started 20 years ago in project-team **Algorithms**. As suggested by the constant questioning of certainty by new potential users, DDMF deserves a formal guarantee of correctness of its content, on a level that proof assistants can provide. Fortunately, the maturity of special-functions algorithms in Algolib makes DDMF a stepping stone for such a formalization: it provides a well-understood and unified algorithmic treatment, without which a formal certification would simply be unreachable.

The formal-proofs component of the team emanates from another project of the MSR–INRIA Joint Centre, namely the **Mathematical Components** project (MathComp). Since 2006, the MathComp group has endeavoured to develop computer-checked libraries of formalized mathematics, using the Coq proof assistant [84]. The methodological aim of the project was to understand the design methods leading to successful large-scale formalizations. The work culminated in 2012 with the completion of a formal proof of the Odd Order Theorem, resulting in the largest corpus of algebraic theories ever machine-checked with a proof assistant and a whole methodology to effectively combine these components in order to tackle complex formalizations. In particular, these libraries provide a good number of the many algebraic objects needed to reason about special functions and their properties, like rational numbers, iterated sums, polynomials, and a rich hierarchy of algebraic structures.

The present team takes benefit from these recent advances to explore the formal certification of the results collected in DDMF. The aim of this project is to concentrate the formalization effort on this delimited area, building on DDMF and the Algolib library, as well as on the Coq system [84] and on the libraries developed by the MathComp project.

2.2 Use computer algebra but convince users beyond reasonable doubt

The following few opinions on computer algebra are, we believe, typical of computer-algebra users' doubts and difficulties when using computer-algebra systems:

- Fredrik Johansson, expert in the multi-precision numerical evaluation of special functions and in fast computer-algebra algorithms, writes on his blog [71]: “Mathematica is great for cross-checking numerical values, but it’s not unusual to run into bugs, so *triple checking is a good habit*.” One answer in the discussion is: “We can claim that Mathematica has [...] *an impossible to understand semantics*: If Mathematica’s output is wrong then change the input. If you don’t like the answer, change the question. That seems to be the philosophy behind.”
- A professor’s advice to students [80] on using Maple: “You may wish to use Maple to check your homework answers. If you do then keep in mind that Maple sometimes gives the *wrong answer, usually because you asked incorrectly, or because of niceties of analytic continuation*. You may even be bitten by an occasional Maple bug, though that has become fairly unlikely. Even with as powerful a tool as Maple you will still *have to devise your own checks* and you will still have to think.”

- Jacques Carette, former head of the maths group at Maplesoft, about a bug [74] when asking Maple to take the limit $\text{limit}(f(n) * \exp(-n), n = \text{infinity})$ for an undetermined function f : “The problem is that there is an *implicit assumption in the implementation* that unknown functions do not ‘grow too fast’.”

As explained by the expert views above, complaints by computer-algebra users are often due to their misunderstanding of what a computer-algebra systems is, namely a purely syntactic tool for calculations, that the user must complement with a semantics. Still, robustness and consistency of computer-algebra systems are not ensured as of today, and, whatever Zeilberger may provocatively say in his Opinion 94 [91], a firmer logical foundation is necessary. Indeed, the fact is that many “bugs” in a computer-algebra system cannot be fixed by just the usual debugging method of tracking down the faulty lines in the code. It is sort of “by design”: assumptions that too often remain implicit are really needed by the design of symbolic algorithms and cannot easily be expressed in the programming languages used in computer algebra. A similar certification initiative has already been undertaken in the domain of numerical computing, in a successful manner [67, 44]. It is natural to undertake a similar approach for computer algebra.

2.3 Make computer algebra and formal proofs help one another

Some of the mathematical objects that interest our team are still totally untouched by formalization. When implementing them and their theory inside a proof assistant, we have to deal with the pervasive discrepancy between the published literature and the actual implementation of computer-algebra algorithms. Interestingly, this forces us to clarify our computer-algebraic view on them, and possibly make us discover holes lurking in published (human) proofs. We are therefore convinced that the close interaction of researchers from both fields, which is what we strive to maintain in this team, is a strong asset.

For a concrete example, the core of Zeilberger’s creative telescoping manipulates rational functions up to simplifications. In summation applications, checking that these simplifications do not hide problematic divisions by 0 is most often left to the reader. In the same vein, in the case of integrals, the published algorithms do not check the convergence of all integrals, especially in intermediate calculations. Such checks are again left to the readers. In general, we expect to revisit the existing algorithms to ensure that they are meaningful for genuine mathematical sequences or functions, and not only for algebraic idealizations.

Another big challenge in this project originates in the scientific difference between computer algebra and formal proofs. Computer algebra seeks speed of calculation on *concrete instances* of algebraic data structures (polynomials, matrices, etc). For their part, formal proofs manipulate symbolic expressions in terms of *abstract variables* understood to represent generic elements of algebraic data structures. In view of this, a continuous challenge is to develop the right, hybrid thinking attitude that is able to effectively manage concrete and abstract values simultaneously, alternatively computing and proving with them.

2.4 Experimental mathematics with special functions

Applications in combinatorics and mathematical physics frequently involve equations of so high orders and so large sizes, that computing or even storing all their coefficients is impossible on existing computers. Making this tractable is an extraordinary challenge. The approach we believe in is to design algorithms of good—ideally quasi-optimal—complexity in order to extract precisely the required data from the equations, while avoiding the computationally intractable task of completely expanding them into an explicit representation.

Typical applications with expected high impact are the automatic discovery and algorithmic proof of results in combinatorics and mathematical physics for which human proofs are currently unattainable.

2.5 Research axes

The implementation of certified symbolic computations on special functions in the Coq proof assistant requires both investigating new formalization techniques and renewing the traditional computer-algebra viewpoint on these standard objects. Large mathematical objects typical of computer algebra occur during formalization, which also requires us to improve the efficiency and ergonomics of Coq. In order

to feed this interdisciplinary activity with new motivating problems, we additionally pursue a research activity oriented towards experimental mathematics in application domains that involve special functions. We expect these applications to pose new algorithmic challenges to computer algebra, which in turn will deserve a formal-certification effort. Finally, DDMF is the motivation and the showcase of our progress on the certification of these computations. While striving to provide a formal guarantee of the correctness of the information it displays, we remain keen on enriching its mathematical content by developing new computer-algebra algorithms.

2.6 Computer algebra certified by the Coq system

Our formalization effort consists in organizing a cooperation between a computer-algebra system and a proof assistant. The computer-algebra system is used to produce efficiently algebraic data, which are later processed by the proof assistant. The success of this cooperation relies on the design of appropriate libraries of formalized mathematics, including certified implementations of certain computer-algebra algorithms. On the other side, we expect that scrutinizing the implementation and the output of computer-algebra algorithms will shed a new light on their semantics and on their correctness proofs, and help clarifying their documentation.

2.6.1 Libraries of formalized mathematics

The appropriate framework for the study of efficient algorithms for special functions is *algebraic*. Representing algebraic theories as Coq formal libraries takes benefit from the methodology emerging from the success of ambitious projects like the formal proof of a major classification result in finite-group theory (the Odd Order Theorem) [58].

Yet, a number of the objects we need to formalize in the present context has never been investigated using any interactive proof assistant, despite being considered as commonplaces in computer algebra. For instance there is up to our knowledge no available formalization of the theory of non-commutative rings, of the algorithmic theory of special-functions closures, or of the asymptotic study of special functions. We expect our future formal libraries to prove broadly reusable in later formalizations of seemingly unrelated theories.

2.6.2 Manipulation of large algebraic data in a proof assistant

Another peculiarity of the mathematical objects we are going to manipulate with the Coq system is their size. In order to provide a formal guarantee on the data displayed by DDMF, two related axes of research have to be pursued. First, efficient algorithms dealing with these large objects have to be programmed and run in Coq. Recent evolutions of the Coq system to improve the efficiency of its internal computations [39, 42] make this objective reachable. Still, how to combine the aforementioned formalization methodology with these cutting-edge evolutions of Coq remains one of the prospective aspects of our project. A second need is to help users *interactively* manipulate large expressions occurring in their conjectures, an objective for which little has been done so far. To address this need, we work on improving the ergonomics of the system in two ways: first, ameliorating the reactivity of Coq in its interaction with the user; second, designing and implementing extensions of its interface to ease our formalization activity. We expect the outcome of these lines of research to be useful to a wider audience, interested in manipulating large formulas on topics possibly unrelated to special functions.

2.6.3 Formal-proof-producing normalization algorithms

Our algorithm certifications inside Coq intend to simulate well-identified components of our Maple packages, possibly by reproducing them in Coq. It would however not have been judicious to re-implement them inside Coq in a systematic way. Indeed for a number of its components, the output of the algorithm is more easily checked than found, like for instance the solving of a linear system. Rather, we delegate the discovery of the solutions to an external, untrusted oracle like Maple. Trusted computations inside Coq then formally validate the correctness of the a priori untrusted output. More often than not, this validation consists in implementing and executing normalization procedures *inside* Coq. A challenge of this automation is to make sure they go to scale while remaining efficient, which requires a Coq

version of non-trivial computer-algebra algorithms. A first, archetypal example we expect to work on is a non-commutative generalization of the normalization procedure for elements of rings [64].

2.7 Better symbolic computations with special functions

Generally speaking, we design algorithms for manipulating special functions symbolically, whether univariate or with parameters, and for extracting algorithmically any kind of algebraic and analytic information from them, notably asymptotic properties. Beyond this, the heart of our research is concerned with parametrised definite summations and integrations. These very expressive operations have far-ranging applications, for instance, to the computation of integral transforms (Laplace, Fourier) or to the solution of combinatorial problems expressed via integrals (coefficient extractions, diagonals). The algorithms that we design for them need to really operate on the level of linear functional systems, differential and of recurrence. In all cases, we strive to design our algorithms with the constant goal of good theoretical complexity, and we observe that our algorithms are also fast in practice.

2.7.1 Special-function integration and summation

Our long-term goal is to design fast algorithms for a general method for special-function integration (*creative telescoping*), and make them applicable to general special-function inputs. Still, our strategy is to proceed with simpler, more specific classes first (rational functions, then algebraic functions, hyperexponential functions, D-finite functions, non-D-finite functions; two variables, then many variables); as well, we isolate analytic questions by first considering types of integration with a more purely algebraic flavor (constant terms, algebraic residues, diagonals of combinatorics). In particular, we expect to extend our recent approach [47] to more general classes (algebraic with nested radicals, for example): the idea is to speed up calculations by making use of an analogue of Hermite reduction that avoids considering certificates. Homologous problems for summation will be addressed as well.

2.7.2 Applications to experimental mathematics

As a consequence of our complexity-driven approach to algorithms design, the algorithms mentioned in the previous paragraph are of good complexity. Therefore, they naturally help us deal with applications that involve equations of high orders and large sizes.

With regard to combinatorics, we expect to advance the algorithmic classification of combinatorial classes like walks and urns. Here, the goal is to determine if enumerative generating functions are rational, algebraic, or D-finite, for example. Physical problems whose modelling involves special-function integrals comprise the study of models of statistical mechanics, like the Ising model for ferro-magnetism, or questions related to Hamiltonian systems.

Number theory is another promising domain of applications. Here, we attempt an experimental approach to the automated certification of integrality of the coefficients of mirror maps for Calabi–Yau manifolds. This could also involve the discovery of new Calabi–Yau operators and the certification of the existing ones. We also plan to algorithmically discover and certify new recurrences yielding good approximants needed in irrationality proofs.

It is to be noted that in all of these application domains, we would so far use general algorithms, as was done in earlier works of ours [46, 50, 48]. To push the scale of applications further, we plan to consider in each case the specifics of the application domain to tailor our algorithms.

2.8 Interactive and certified mathematical web sites

In continuation of our past project of an [encyclopedia](#), we ambition to both enrich and certify the formulas about the special functions that we provide online. For each function, our website shows its essential properties and the mathematical objects attached to it, which are often infinite in nature (numerical evaluations, asymptotic expansions). An interactive presentation has the advantage of allowing for adaption to the user’s needs. More advanced content will broaden the encyclopedia:

- the algorithmic discussion of equations with parameters, leading to certified automatic case analysis based on arithmetic properties of the parameters;

- lists of summation and integral formulas involving special functions, including validity conditions on the parameters;
- guaranteed large-precision numerical evaluations.

3 Research program

3.1 Studying special functions by computer algebra

Computer algebra manipulates symbolic representations of exact mathematical objects in a computer, in order to perform computations and operations like simplifying expressions and solving equations for “closed-form expressions”. The manipulations are often fundamentally of algebraic nature, even when the ultimate goal is analytic. The issue of efficiency is a particular one in computer algebra, owing to the extreme swell of the intermediate values during calculations.

Our view on the domain is that research on the algorithmic manipulation of special functions is anchored between two paradigms:

- adopting linear differential equations as the right data structure for special functions,
- designing efficient algorithms in a complexity-driven way.

It aims at four kinds of algorithmic goals:

- algorithms combining functions,
- functional equations solving,
- multi-precision numerical evaluations,
- guessing heuristics.

This interacts with three domains of research:

- computer algebra, meant as the search for quasi-optimal algorithms for exact algebraic objects,
- symbolic analysis/algebraic analysis;
- experimental mathematics (combinatorics, mathematical physics, ...).

This view is made explicit in the present section.

3.2 Equations as a data structure

Numerous special functions satisfy linear differential and/or recurrence equations. Under a mild technical condition, the existence of such equations induces a finiteness property that makes the main properties of the functions decidable. We thus speak of *D-finite functions*. For example, 60 % of the chapters in the handbook [38] describe D-finite functions. In addition, the class is closed under a rich set of algebraic operations. This makes linear functional equations just the right data structure to encode and manipulate special functions. The power of this representation was observed in the early 1990s [90], leading to the design of many algorithms in computer algebra. Both on the theoretical and algorithmic sides, the study of D-finite functions shares much with neighbouring mathematical domains: differential algebra, D-module theory, differential Galois theory, as well as their counterparts for recurrence equations.

3.3 Algorithms combining functions

Differential/recurrence equations that define special functions can be recombined [90] to define: additions and products of special functions; compositions of special functions; integrals and sums involving special functions. Zeilberger’s fast algorithm for obtaining recurrences satisfied by parametrised binomial sums was developed in the early 1990s already [92]. It is the basis of all modern definite summation and integration algorithms. The theory was made fully rigorous and algorithmic in later works, mostly by a group in RISC (Linz, Austria) and by members of the team [79, 87, 53, 51, 52, 72]. The past ÉPI Algorithms contributed several implementations (gfun [82], Mgfund [53]).

3.4 Solving functional equations

Encoding special functions as defining linear functional equations postpones some of the difficulty of the problems to a delayed solving of equations. But at the same time, solving (for special classes of functions) is a sub-task of many algorithms on special functions, especially so when solving in terms of polynomial or rational functions. A lot of work has been done in this direction in the 1990s; more intensively since the 2000s, solving differential and recurrence equations in terms of special functions has also been investigated.

3.5 Multi-precision numerical evaluation

A major conceptual and algorithmic difference exists for numerical calculations between data structures that fit on a machine word and data structures of arbitrary length, that is, *multi-precision* arithmetic. When multi-precision floating-point numbers became available, early works on the evaluation of special functions were just promising that “most” digits in the output were correct, and performed by heuristically increasing precision during intermediate calculations, without intended rigour. The original theory has evolved in a twofold way since the 1990s: by making computable all constants hidden in asymptotic approximations, it became possible to guarantee a *prescribed* absolute precision; by employing state-of-the-art algorithms on polynomials, matrices, etc, it became possible to have evaluation algorithms in a time complexity that is linear in the output size, with a constant that is not more than a few units. On the implementation side, several original works exist, one of which (NumGfun [77]) is used in our DDME.

3.6 Guessing heuristics

“Differential approximation”, or “Guessing”, is an operation to get an ODE likely to be satisfied by a given approximate series expansion of an unknown function. This has been used at least since the 1970s and is a key stone in spectacular applications in experimental mathematics [50]. All this is based on subtle algorithms for Hermite–Padé approximants [40]. Moreover, guessing can at times be complemented by proven quantitative results that turn the heuristics into an algorithm [49]. This is a promising algorithmic approach that deserves more attention than it has received so far.

3.7 Complexity-driven design of algorithms

The main concern of computer algebra has long been to prove the feasibility of a given problem, that is, to show the existence of an algorithmic solution for it. However, with the advent of faster and faster computers, complexity results have ceased to be of theoretical interest only. Nowadays, a large track of works in computer algebra is interested in developing fast algorithms, with time complexity as close as possible to linear in their output size. After most of the more pervasive objects like integers, polynomials, and matrices have been endowed with fast algorithms for the main operations on them [59], the community, including ourselves, started to turn its attention to differential and recurrence objects in the 2000s. The subject is still not as developed as in the commutative case, and a major challenge remains to understand the combinatorics behind summation and integration. On the methodological side, several paradigms occur repeatedly in fast algorithms: “divide and conquer” to balance calculations, “evaluation and interpolation” to avoid intermediate swell of data, etc. [45].

3.8 Encyclopedias

Handbooks collecting mathematical properties aim at serving as reference, therefore trusted, documents. The decision of several authors or maintainers of such knowledge bases to move from paper books [38, 78, 83] to websites and wikis, for instance for *special functions* or for *integer sequences*, allows for a more collaborative effort in proof reading. Another step toward further confidence is to manage to generate the content of an encyclopedia by computer-algebra programs, as is the case with the *Wolfram Functions Site* or *DDMF*. Yet, due to the lingering doubts about computer-algebra systems, some encyclopedias propose both cross-checking by different systems and *handwritten companion paper proofs* of their content. As of today, there is no encyclopedia certified with formal proofs.

3.9 Computer algebra and symbolic logic

Several attempts have been made in order to extend existing computer-algebra systems with symbolic manipulations of logical formulas. Yet, these works are more about extending the expressivity of computer-algebra systems than about improving the standards of correctness and semantics of the systems. Conversely, several projects have addressed the communication of a proof system with a computer-algebra system, resulting in an increased automation available in the proof system, to the price of the uncertainty of the computations performed by this oracle.

3.10 Certifying systems for computer algebra

More ambitious projects have tried to design a new computer-algebra system providing an environment where the user could both program efficiently and elaborate formal and machine-checked proofs of correctness, by calling a general-purpose proof assistant like the Coq system. This approach requires a huge manpower and a daunting effort in order to re-implement a complete computer-algebra system, as well as the libraries of formal mathematics required by such formal proofs.

3.11 Semantics for computer algebra

The move to machine-checked proofs of the mathematical correctness of the output of computer-algebra implementations demands a prior clarification about the often implicit assumptions on which the presumably correctly implemented algorithms rely. Interestingly, this preliminary work, which could be considered as independent from a formal certification project, is seldom precise or even available in the literature.

3.12 Formal proofs for symbolic components of computer-algebra systems

A number of authors have investigated ways to organize the communication of a chosen computer-algebra system with a chosen proof assistant in order to certify specific components of the computer-algebra systems, experimenting various combinations of systems and various formats for mathematical exchanges. Another line of research consists in the implementation and certification of computer-algebra algorithms inside the logic [86, 64, 73] or as a proof-automation strategy. Normalization algorithms are of special interest when they allow to check results possibly obtained by an external computer-algebra oracle [56]. A discussion about the systematic separation of the search for a solution and the checking of the solution is already clearly outlined in [70].

3.13 Formal proofs for numerical components of computer-algebra systems

Significant progress has been made in the certification of numerical applications by formal proofs. Libraries formalizing and implementing floating-point arithmetic as well as large numbers and arbitrary-precision arithmetic are available. These libraries are used to certify floating-point programs, implementations of mathematical functions and for applications like hybrid systems.

3.14 Machine-checked proofs of formalized mathematics

To be checked by a machine, a proof needs to be expressed in a constrained, relatively simple formal language. Proof assistants provide facilities to write proofs in such languages. But, as merely writing, even in a formal language, does not constitute a formal proof just per se, proof assistants also provide a proof checker: a small and well-understood piece of software in charge of verifying the correctness of arbitrarily large proofs. The gap between the low-level formal language a machine can check and the sophistication of an average page of mathematics is conspicuous and unavoidable. Proof assistants try to bridge this gap by offering facilities, like notations or automation, to support convenient formalization methodologies. Indeed, many aspects, from the logical foundation to the user interface, play an important role in the feasibility of formalized mathematics inside a proof assistant.

3.15 Logical foundations and proof assistants

While many logical foundations for mathematics have been proposed, studied, and implemented, type theory is the one that has been more successfully employed to formalize mathematics, to the notable exception of the Mizar system [75], which is based on set theory. In particular, the calculus of construction (CoC) [54] and its extension with inductive types (CIC) [55], have been studied for more than 20 years and been implemented by several independent tools (like Lego, Matita, and Agda). Its reference implementation, Coq [84], has been used for several large-scale formalizations projects (formal certification of a compiler back-end; four-color theorem). Improving the type theory underlying the Coq system remains an active area of research. Other systems based on different type theories do exist and, whilst being more oriented toward software verification, have been also used to verify results of mainstream mathematics (prime-number theorem; Kepler conjecture).

3.16 Computations in formal proofs

The most distinguishing feature of CoC is that computation is promoted to the status of rigorous logical argument. Moreover, in its extension CIC, we can recognize the key ingredients of a functional programming language like inductive types, pattern matching, and recursive functions. Indeed, one can program effectively inside tools based on CIC like Coq. This possibility has paved the way to many effective formalization techniques that were essential to the most impressive formalizations made in CIC.

Another milestone in the promotion of the computations-as-proofs feature of Coq has been the integration of compilation techniques in the system to speed up evaluation. Coq can now run realistic programs in the logic, and hence easily incorporates calculations into proofs that demand heavy computational steps.

Because of their different choice for the underlying logic, other proof assistants have to simulate computations outside the formal system, and indeed fewer attempts to formalize mathematical proofs involving heavy calculations have been made in these tools. The only notable exception, which was finished in 2014, the Kepler conjecture, required a significant work to optimize the rewriting engine that simulates evaluation in Isabelle/HOL.

3.17 Large-scale computations for proofs inside the Coq system

Programs run and proved correct inside the logic are especially useful for the conception of automated decision procedures. To this end, inductive types are used as an internal language for the description of mathematical objects by their syntax, thus enabling programs to reason and compute by case analysis and recursion on symbolic expressions.

The output of complex and optimized programs external to the proof assistant can also be stamped with a formal proof of correctness when their result is easier to *check* than to *find*. In that case one can benefit from their efficiency without compromising the level of confidence on their output at the price of writing and certify a checker inside the logic. This approach, which has been successfully used in various contexts, is very relevant to the present research project.

3.18 Relevant contributions from the Mathematical Component libraries

Representing abstract algebra in a proof assistant has been studied for long. The libraries developed by the MathComp project for the proof of the Odd Order Theorem provide a rather comprehensive hierarchy of structures; however, they originally feature a large number of instances of structures that they need to organize. On the methodological side, this hierarchy is an incarnation of an original work [58] based on various mechanisms, primarily type inference, typically employed in the area of programming languages. A large amount of information that is implicit in handwritten proofs, and that must become explicit at formalization time, can be systematically recovered following this methodology.

Small-scale reflection [61] is another methodology promoted by the MathComp project. Its ultimate goal is to ease formal proofs by systematically dealing with as many bureaucratic steps as possible, by automated computation. For instance, as opposed to the style advocated by Coq's standard library, decidable predicates are systematically represented using computable boolean functions: comparison

on integers is expressed as program, and to state that $a \leq b$ one compares the output of this program run on a and b with *true*. In many cases, for example when a and b are values, one can prove or disprove the inequality by pure computation.

The MathComp library was consistently designed after uniform principles of software engineering. These principles range from simple ones, like naming conventions, to more advanced ones, like generic programming, resulting in a robust and reusable collection of formal mathematical components. This large body of formalized mathematics covers a broad panel of algebraic theories, including of course advanced topics of finite group theory, but also linear algebra, commutative algebra, Galois theory, and representation theory. We refer the interested reader to the online documentation of these libraries [85], which represent about 150,000 lines of code and include roughly 4,000 definitions and 13,000 theorems.

Topics not addressed by these libraries and that might be relevant to the present project include real analysis and differential equations. The most advanced work of formalization on these domains is available in the HOL-Light system [66, 68, 69], although some existing developments of interest [43, 76] are also available for Coq. Another aspect of the MathComp libraries that needs improvement, owing to the size of the data we manipulate, is the connection with efficient data structures and implementations, which only starts to be explored.

3.19 User interaction with the proof assistant

The user of a proof assistant describes the proof he wants to formalize in the system using a textual language. Depending on the peculiarities of the formal system and the applicative domain, different proof languages have been developed. Some proof assistants promote the use of a declarative language, when the Coq and Matita systems are more oriented toward a procedural style.

The development of the large, consistent body of MathComp libraries has prompted the need to design an alternative and coherent language extension for the Coq proof assistant [63, 62], enforcing the robustness of proof scripts to the numerous changes induced by code refactoring and enhancing the support for the methodology of small-scale reflection.

The development of large libraries is quite a novelty for the Coq system. In particular any long-term development process requires the iteration of many refactoring steps and very little support is provided by most proof assistants, with the notable exception of Mizar [81]. For the Coq system, this is an active area of research.

4 Application domains

4.1 Computer Algebra in Mathematics

Our expertise in computer algebra and complexity-driven design of algebraic algorithms has applications in various domains, including:

- combinatorics, especially the study of combinatorial walks,
- theoretical computer science, like by the study of automatic sequences,
- number theory, by the analysis of the nature of so-called periods.

5 Highlights of the year

5.1 Refounding the team: towards computer algebra, experimental mathematics, and interactions

The team has worked on its renewal and has presented a project for a new team, MATHEXP. This new team will develop and implement symbolic and semi-numerical computational methods to deal with special functions and numbers in experimental mathematics.

5.2 ERC project: 10,000 DIGITS

Lairez's ERC proposal has been retained for funding with a grant of roughly 1.4 million Euro. The project will focus on the foundations of transcendental methods in numerical nonlinear algebra.

5.3 David P. Robbins Prize

Alin Bostan, of the team, together with Irina Kurkova and Kilian Raschel, will receive the [2022 AMS David P. Robbins Prize](#) for their paper "A human proof of Gessel's lattice path conjecture," published in Transactions of the American Mathematical Society in 2017. The paper proves highly nontrivial enumeration results on a family of lattice paths known as Gessel walks.

5.4 Promotion

Alin Bostan was nominated "Directeur de Recherche" in 2021.

6 New results

6.1 Algebraic algorithms on fundamental objects

6.1.1 The art of algorithmic guessing in gfun

The technique of guessing can be very fruitful when dealing with sequences which arise in practice. This holds true especially when guessing is performed algorithmically and efficiently. The ideal tool for it exists as a package named gfun in the software Maple. In this submitted paper [\[36\]](#) Sergey Yurkevich explores and explains some of gfun's possibilities and illustrates them on two examples from recent mathematical research by him and his collaborators.

6.1.2 A Sage package for the symbolic-numeric factorization of linear differential operators

Alexandre Goyer presented a SageMath implementation of the symbolic-numeric algorithm introduced by van der Hoeven in 2007 for factoring linear differential operators whose coefficients are rational functions [\[21\]](#).

6.1.3 A simple and fast algorithm for computing the N -th term of a linearly recurrent sequence

In [\[24\]](#) Alin Bostan and Ryuhei Mori (Tokyo Institute of Technology, Japan) designed a simple and fast algorithm for computing the N -th term of a given linearly recurrent sequence. The new algorithm uses $O(M(d) \log N)$ arithmetic operations, where d is the order of the recurrence, and $M(d)$ denotes the number of arithmetic operations for computing the product of two polynomials of degree d . The state-of-the-art algorithm, due to Fiduccia (1985), had the same arithmetic complexity up to a constant factor. The new algorithm is simpler, faster and obtained by a totally different method. They also discuss several algorithmic applications, notably to polynomial modular exponentiation ($P^N \bmod Q$), on which many other useful algorithms rely, either in computer algebra (e.g., polynomial factoring over finite fields), or in algorithmic number theory (e.g., primality tests) or in effective algebraic geometry (e.g., counting points on curves over finite fields).

6.1.4 Fast computation of the N -th term of a q -holonomic sequence and applications

In 1977, Strassen invented a famous baby-step/giant-step algorithm that computes the factorial $N!$ in arithmetic complexity quasi-linear in \sqrt{N} . In 1988, the Chudnovsky brothers generalized Strassen's algorithm to the computation of the N -th term of any holonomic sequence in essentially the same arithmetic complexity. In [\[17\]](#), Alin Bostan together with his PhD student Sergey Yurkevich designed q -analogues of these algorithms. They first extend Strassen's algorithm to the computation of the q -factorial of N , then Chudnovskys' algorithm to the computation of the N -th term of any q -holonomic sequence. Both algorithms work in arithmetic complexity quasi-linear in \sqrt{N} ; surprisingly, they are

simpler than their analogues in the holonomic case. They provide a detailed cost analysis, in both arithmetic and bit complexity models. Moreover, they describe various algorithmic consequences, including the acceleration of polynomial and rational solving of linear q -differential equations, and the fast evaluation of large classes of polynomials, including a family recently considered by Nogneng and Schost.

6.1.5 Improved algorithms for left factorial residues

In [11], Alin Bostan together with Vladica Andrejić (University of Belgrade, Serbia) and Milos Tatarevic (CoinList, Alameda, CA) presented improved algorithms for computing the left factorial residues $!p = 0! + 1! + \dots + (p-1)! \pmod p$. They used these algorithms for the calculation of the residues $!p \pmod p$, for all primes p up to 2^{40} . Their results confirm that Kurepa's left factorial conjecture is still an open problem, as they show that there are no odd primes $p < 2^{40}$ such that p divides $!p$. Additionally, they confirmed that there are no socialist primes p with $5 < p < 2^{40}$.

6.1.6 Explicit degree bounds for right factors of linear differential operators

If a linear differential operator with rational function coefficients is reducible, its factors may have coefficients with numerators and denominators of very high degree. When the base field is \mathbb{C} , Alin Bostan together with Bruno Salvy (Inria and ENS Lyon) and Tanguy Rivoal (CNRS and U. Grenoble) gave in [15] a completely explicit bound for the degrees of the monic right factors in terms of the degree and the order of the original operator, as well as the largest modulus of the local exponents at all its singularities. As a consequence, if a differential operator L has rational function coefficients over a number field, they obtain degree bounds for its monic right factors in terms of the degree, the order and the height of L , and of the degree of the number field.

6.2 Polynomial systems and geometry

6.2.1 Gröbner bases and critical values: the asymptotic combinatorics of determinantal systems

In [30] Alin Bostan, together with co-authors Jérémy Berthomieu, Andrew Ferguson and Mohab Safey El Din (all from Sorbonne Université), studied *determinantal polynomial systems*. These are polynomial systems involving maximal minors of some given matrix. An important situation where these arise is the computation of the critical values of a polynomial map restricted to an algebraic set. This leads directly to a strategy for, among other problems, polynomial optimisation.

Computing Gröbner bases is a classical method for solving polynomial systems in general. For practical computations, this consists of two main stages. First, a Gröbner basis is computed with respect to a DRL (degree reverse lexicographic) ordering. Then, a change of ordering algorithm, such as Sparse-FGLM, designed by Faugère and Mou, is used to find a Gröbner basis of the same system but with respect to a lexicographic ordering. The complexity of this latter step, in terms of the number of arithmetic operations in the ground field, is $O(mD^2)$, where D is the degree of the ideal generated by the input and m is the number of non-trivial columns of a certain $D \times D$ matrix.

While asymptotic estimates are known for m in the case of *generic* polynomial systems, thus far, the complexity of Sparse-FGLM was unknown for the class of determinantal systems.

By assuming Fröberg's conjecture, a classical conjecture in commutative algebra, and thus ensuring that the Hilbert series of generic determinantal ideals have the necessary structure, the authors expand the work of Moreno-Socías by detailing the structure of the DRL staircase in the determinantal setting. Then, they study the asymptotics of the quantity m by relating it to the coefficients of these Hilbert series. Consequently, they arrive at a new bound on the complexity of the Sparse-FGLM algorithm for generic determinantal systems and, in particular, for generic critical point systems.

The ideal is considered inside the polynomial ring $\mathbb{K}[x_1, \dots, x_n]$, where \mathbb{K} is some infinite field, generated by p generic polynomials of degree d and the maximal minors of a $p \times (n-1)$ polynomial matrix with generic entries of degree $d-1$. Then, in this setting, for the case $d=2$ and for $n \gg p$ the paper [30] establishes an exact formula for m in terms of n and p . Moreover, for $d \geq 3$, it gives a tight asymptotic formula, as $n \rightarrow \infty$, for m in terms of n , p and d .

6.2.2 Computing the dimension of real algebraic sets

In [25], Pierre Lairez and Mohab Safey El Din (Sorbonne Université) designed a new algorithm for computing the dimension in a real-algebraic setting. Let V be the set of real common solutions to $F = (f_1, \dots, f_s)$ in $\mathbb{R}[x_1, \dots, x_n]$ and D be the maximum total degree of the f_i 's. The authors design an algorithm which on input F computes the dimension of V . Letting L be the evaluation complexity of F and $s = 1$, it runs using $O^\sim(LD^{n(d+3)+1})$ arithmetic operations in \mathbb{Q} and at most $D^{n(d+1)}$ isolations of real roots of polynomials of degree at most D^n .

Their algorithm depends on the *real* geometry of V ; its practical behavior is more governed by the number of topology changes in the fibers of some well-chosen maps. Hence, the above worst-case bounds are rarely reached in practice, the factor D^{nd} being in general much lower on practical examples. They report on an implementation showing its ability to solve problems which were out of reach of the state-of-the-art implementations.

6.2.3 An algorithmic approach to Rupert's problem

A polyhedron $\mathbf{P} \subseteq \mathbb{R}^3$ has Rupert's property if a hole can be cut into it, such that a copy of \mathbf{P} can pass through this hole. There are several works investigating this property for some specific polyhedra: for example, it is known that all 5 Platonic and 9 out of the 13 Archimedean solids admit Rupert's property. A commonly believed conjecture states that every convex polyhedron is Rupert. By translating the problem to the decidability question of emptiness of semi-algebraic sets, Jakob Steininger and Sergey Yurkevich prove in [35] that Rupert's problem is algorithmically decidable for polyhedra with algebraic coordinates. They also design a probabilistic algorithm which can efficiently prove that a given polyhedron is Rupert. Using this algorithm the authors not only confirm this property for the known Platonic and Archimedean solids, but also prove it for one of the remaining Archimedean polyhedra and many others. Moreover, almost all known Nieuwland numbers are significantly improved. Finally, Steininger and Yurkevich conjecture, based on statistical evidence, that the Rhombicosidodecahedron is in fact not Rupert.

6.3 Applications to special functions and number theory

6.3.1 A hypergeometric proof that Iso is bijective

A short and elementary proof of the main technical result of the recent article "On the uniqueness of Clifford torus with prescribed isoperimetric ratio" [89] by Thomas Yu and Jingmin Chen has been found by Alin Bostan and Sergey Yurkevich in [16]. The key of the new proof is an explicit expression of the central function (Iso, proved to be bijective) as a quotient of Gaussian hypergeometric functions.

6.3.2 On an integral identity

In [14], Alin Bostan together with Fernando Chamizo (Universidad Autónoma de Madrid and ICMAT, Spain) and Mikael Persson Sundqvist (Lund University, Sweden) gave three elementary proofs of a nice equality of definite integrals, recently proven by Ekhad, Zeilberger and Zudilin. The equality arises in the theory of bivariate hypergeometric functions, and has connections with irrationality proofs in number theory. They furthermore provide a generalization together with an equally elementary proof and discuss some consequences.

6.3.3 A short proof of a non-vanishing result by Conca, Krattenthaler and Watanabe

In their 2009 paper *Regular sequences of symmetric polynomials*, Aldo Conca, Christian Krattenthaler and Junzo Watanabe needed to prove, as an intermediate result, the fact that for any $h \geq 1$, the rational number

$$\sum_{b=0}^{\lfloor h/3 \rfloor} \frac{(-1)^{h-b}}{h-b} \binom{h-b}{2b} \left(\frac{2}{3}\right)^b$$

is non-zero, except for $h = 3$. The proof in their paper (Appendix, pp. 190–199) performs a long and quite intricate 3-adic analysis. In [12], Alin Bostan proposes a shorter and elementary proof.

6.4 Applications to combinatorics

6.4.1 Counting walks with large steps in an orthant

In the past fifteen years, the enumeration of lattice walks with steps taken in a prescribed set and confined to a given cone, especially the first quadrant of the plane, has been intensely studied. As a result, the generating functions of quadrant walks are now well-understood, provided the allowed steps are *small*. In particular, having small steps is crucial for the definition of a certain group of bi-rational transformations of the plane. It has been proved that this group is finite if and only if the corresponding generating function is D-finite. This group is also the key to the uniform solution of 19 of the 23 small step models possessing a finite group. In contrast, almost nothing was known for walks with arbitrary steps. In [13], Alin Bostan together with Mireille Bousquet-Mélou (CNRS, Bordeaux) and Stephen Melczer (U. Pennsylvania, Philadelphia, USA), extended the definition of the group, or rather of the associated orbit, to this general case, and generalized the above uniform solution of small step models. When this approach works, it invariably yields a D-finite generating function. They applied it to many quadrant problems, including some infinite families. After developing the general theory, the authors of [13] considered the 13 110 two-dimensional models with steps in $\{-2, -1, 0, 1\}^2$ having at least one -2 coordinate. They proved that only 240 of them have a finite orbit, and solve 231 of them with their method. The 9 remaining models are the counterparts of the 4 models of the small step case that resist the uniform solution method (and which are known to have an algebraic generating function). They conjecture D-finiteness for their generating functions (but only two of them are likely to be algebraic!), and proved non-D-finiteness for the 12 870 models with an infinite orbit, except for 16 of them.

6.4.2 The generating function of Kreweras walks with interacting boundaries is not algebraic

Beaton, Owczarek and Xu (2019) studied generating functions of Kreweras walks and of reverse Kreweras walks in the quarter plane, with interacting boundaries. They proved that for the reverse Kreweras step set, the generating function is always algebraic, and for the Kreweras step set, the generating function is always D-finite. However, apart from the particular case where the interactions are symmetric in x and y , they left open the question of whether the latter one is algebraic. Using computer algebra tools, Alin Bostan, together with Manuel Kauers and Thibaut Verron (University, Linz, Austria) confirmed [23] the previous intuition that the generating function of Kreweras walks is not algebraic, apart from the particular case already identified.

6.4.3 Random walks in orthants and lattice path combinatorics

In the second edition of the book [57], original methods were proposed to determine the invariant measure of random walks in the quarter plane with small jumps (size 1), the general solution being obtained via reduction to boundary value problems. Among other things, an important quantity, the so-called *group of the walk*, allows to deduce theoretical features about the nature of the solutions. In particular, when the *order* of the group is finite and the underlying algebraic curve is of genus 0 or 1, necessary and sufficient conditions have been given for the solution to be rational, algebraic or D-finite (i.e. solution of a linear differential equation). In this framework, a number of difficult open problems related to lattice path combinatorics are currently being explored boundary Alin Bostan, Frédéric Chyzak, and Guy Fayolle, both from theoretical and computer algebra points of view: concrete computation of the criteria, utilization of differential Galois theory, genus greater than 1 (i.e., when some jumps are of size ≥ 2), etc. This relates simple product-form stochastic networks (so-called *Jackson networks*) and explicit solutions of functional equations for counting lattice walks. Some partial extensions of [33] are under development.

6.4.4 On some combinatorial sequences associated to invariant theory

In [31], Alin Bostan together with Jordan Tirrell (Washington College, USA), Bruce W. Westbury (U. Texas at Dallas, USA) and Yi Zhang (Xi'an Jiaotong-Liverpool University, Suzhou, China) study the enumerative and analytic properties of some sequences constructed using tensor invariant theory. The first family, containing the so-called *octant sequences*, is constructed from the exceptional Lie group G_2 . The second

family, containing the so-called *quadrant sequences*, is constructed from the special linear group $SL(3)$. All sequences are defined as the dimension of the subspace of invariant tensors in the tensor powers of the corresponding representation. corresponding sequences are related by binomial transforms. The authors first give combinatorial interpretations for the first octant sequence, T_3 , based on interpretations of the sequences in the first family as lattice walks in the plane. They then show that the second octant sequence E_3 is the binomial transform of T_3 ; this result provides an unexpected connection between the invariant theory of G_2 and the combinatorics of set partitions. A third result is a proof (actually three independent proofs) of a recurrence satisfied by T_3 that was conjectured by Mihailovs in the early 2000s. Similar results are obtained for the sequences of the quadrant sequences. These sequences also have interpretations as enumerating two-dimensional lattice walks. They are all P-recursive, and recurrence relations are proved for them. In all cases the associated differential operators are of third order and have the remarkable property that they can be solved to give closed formulae for the ordinary generating functions in terms of classical Gaussian hypergeometric functions. Moreover, it is shown that the octant sequences and the quadrant sequences are related by the branching rules for the inclusion of $SL(3)$ in G_2 .

6.4.5 Computer algebra in the service of enumerative combinatorics

Alin Bostan gave a plenary invited talk at the conference [ISSAC'21](#). On this occasion, he wrote the overview article [\[22\]](#) which can be seen as a condensed version of his Habilitation thesis defended in 2017. The main topic is the use of computer algebra tools to explore and to solve a number of difficult questions in enumerative combinatorics, notably related to the classification of lattice walks. Alin Bostan gives an overview of recent results on structural properties (e.g., algebraicity versus transcendence) and on explicit formulas for generating functions of walks with small steps in the quarter plane. In doing so, he emphasizes the algorithmic nature of the methodology, especially two important paradigms: “guess-and-prove” and “creative telescoping”.

6.5 Applications to probability

6.5.1 Martin boundary of killed random walks on isoradial graphs

Alin Bostan contributed to an article by C. Boutillier (Sorbonne Université) and K. Raschel (CNRS, Université de Tours) [\[19\]](#), devoted to the study of random walks on isoradial graphs. Contrary to the lattice case, isoradial graphs are not translation invariant, do not admit any group structure and are spatially non-homogeneous. However, Boutillier and Raschel have been able to obtain analogues of a celebrated result by Ney and Spitzer (1966) on the so-called *Martin kernel* (ratio of Green functions started at different points). Alin Bostan provided in the Appendix two different proofs of the fact that some algebraic power series arising in this context have non-negative coefficients.

6.5.2 Genus and classification of random walks in the quarter plane

In collaboration with R. Iasnogorodski (SPCPA, Saint-Petersburg), Guy Fayolle analyzes the *kernel* $K(x, y, t)$ of the basic functional equation associated with the trivariate counting generating function (CGF) of walks in the quarter plane. In their paper [\[20\]](#), taking $t \in]0, 1[$, they provide the conditions on the step set $\{p_{i,j}\}$ to decide whether the walks are *singular* or *regular*, as defined in [\[57, Section 2.3\]](#). These conditions are independent of $t \in]0, 1[$ and given in terms of *step set configurations*. They also find the configurations for the kernel to be of genus 0, knowing that the genus is always ≤ 1 . All these conditions are very similar to the case $t = 1$ considered in [\[57\]](#). Their results extend an [earlier work](#), which considers only very special situations, namely when $t \in]0, 1[$ is a transcendental number over the field $Q(p_{i,j})$.

6.5.3 Reflected Brownian motion in a nonconvex cone

In an ongoing work in collaboration with S. Franceschi (LMO, Paris-Saclay University) and K. Raschel (CNRS, Tours University), Guy Fayolle states a system of functional equations satisfied by the Laplace transform of the stationary distribution of a reflected Brownian motion (SRBM) in a two-dimensional non-convex cone. While the case of convex cones is now reasonably well studied, the framework of non-convex cones turns out to be more challenging, as shown by similar research carried out in a discrete

setting. They show in particular that the problem can be reduced to a boundary value problem of Riemann–Hilbert–Carleman type on an hyperbola, for a two-dimensional vector of meromorphic functions. This seems to be a quite original result.

6.5.4 Persistence probabilities and Mallows-Riordan polynomials

Mallows-Riordan polynomials, sometimes also called *inversion polynomials*, form a family of polynomials with integer coefficients appearing in many counting problems in enumerative combinatorics. They are also connected with the cumulant generating function of the classical log-normal distribution in probability theory. In [29] Alin Bostan, together with his probabilist co-authors Gerold Alsmeyer (U. Münster), Kilian Raschel (CNRS, U. Angers) and Thomas Simon (U. Lille), provide a probabilistic interpretation of the Mallows-Riordan polynomials that is not only quite different from the classical connection with the log-normal distribution, but in fact also rather unexpected. More precisely, they establish exact formulae in terms of Mallows-Riordan polynomials for the persistence probabilities of a class of order-one autoregressive processes with symmetric uniform innovations. These exact formulae then lead to precise asymptotics of the corresponding persistence probabilities. The connection of the Mallows-Riordan polynomials with the volumes of certain polytopes is also discussed. Two further results provide general factorizations of AR(1) models with continuous symmetric innovations, one for negative and one for positive drift. The second factorization extends a classical universal formula of Sparre Andersen for symmetric random walks.

6.6 Diagonals

6.6.1 On the q -analogue of Pólya's Theorem

Bostan and Yurkevich answer in [32] a question posed by Michael Aissen in 1979 about the q -analogue of a classical theorem of George Pólya (1922) on the algebraicity of (generalized) diagonals of bivariate rational power series. In particular, they prove that the answer to Aissen's question, in which he considers q as a variable, is negative in general. Moreover, they show that the answer is positive if and only if q is a root of unity.

6.6.2 Diagonal representation of algebraic power series: a glimpse behind the scenes

There are many viewpoints on algebraic power series, ranging from the abstract ring-theoretic notion of Henselization to the very explicit perspective as diagonals of certain rational functions. Denef and Lipshitz proved in 1987 that any algebraic power series in n variables can be written as a diagonal of a rational power series in one variable more. Their proof uses a lot of involved theory and machinery which remains hidden to the reader in the original article. In the work [28], which is based on his master's thesis, Sergey Yurkevich explained these tools by motivating while defining them and reproving most of their interesting parts. Moreover, he provided a new significant improvement on the Artin-Mazur lemma, proving the existence of a 2-dimensional code of algebraic power series.

6.6.3 On a class of hypergeometric diagonals

In [18], Alin Bostan together with his PhD student Sergey Yurkevich proved that the diagonal of any finite product of algebraic functions of the form

$$(1 - x_1 - \dots - x_n)^R, \quad R \in \mathbb{Q},$$

is a generalized hypergeometric function, and they provided explicit description of its parameters. The particular case $(1 - x - y)^R / (1 - x - y - z)$ corresponds to the main identity of Abdelaziz, Koutschan and Maillard in [37, §3.2]. The result in [18] is useful in both directions: on the one hand it shows that Christol's conjecture holds true for a large class of hypergeometric functions, on the other hand it allows for a very explicit and general viewpoint on the diagonals of algebraic functions of the type above. Finally, in contrast to [37], the new proof is completely elementary and does not require any algorithmic help.

6.7 Proceedings of a conference on our topics

6.7.1 Transcendence in Algebra, Combinatorics, Geometry and Number Theory

Alin Bostan together with Kilian Raschel (CNRS, U. Angers) served as editors of the book “Transcendence in Algebra, Combinatorics, Geometry and Number Theory” [27], published by Springer in the collection “*Proceedings in Mathematics and Statistics*”. This proceedings volume gathers together original articles and survey works that originate from presentations given at the conference *Transient Transcendence in Transylvania*, held in Braşov, Romania, from May 13th to 17th, 2019. The conference, organized by Alin Bostan and Kilian Raschel, had gathered international experts from various fields of mathematics and computer science, with diverse interests and viewpoints on transcendence. The covered topics are related to algebraic and transcendental aspects of special functions and special numbers arising in algebra, combinatorics, geometry and number theory. Besides contributions on key topics from invited speakers, this volume also brings selected papers from attendees.

7 Partnerships and cooperations

7.1 International initiatives

7.1.1 Participation in other International Programs

PhD project of Yurkevich

Title: Integer sequences, algebraic series and differential operators.

Partner Institution(s): • University of Vienna, Austria.

Date/Duration: September 2020 – August 2023

Additional info/keywords: The PhD thesis project of Sergey Yurkevich is a *cotutelle* with the University of Vienna (Austria). The supervisors are Alin Bostan on the French side and Herwig Hauser on the Austrian side. The investigation topic covers on the one hand integer sequences naturally arising in various scientific disciplines such as number theory, combinatorics and physics, and on the other hand solutions to special kinds of differential equations.

7.2 European initiatives

7.2.1 Horizon Europe

- *ERC Starting Grant*. Pierre Lairez was awarded an ERC Starting Grant for his project “10000 DIGITS”. The project will start in 2022.

7.3 National initiatives

7.3.1 ANR

- *De rerum natura*. This project, set up by the team, was accepted this year and will be funded until 2023. It gathers over 20 experts from four fields: computer algebra; the Galois theories of linear functional equations; number theory; combinatorics and probability. Our goal is to obtain classification algorithms for number theory and combinatorics, particularly so for deciding irrationality and transcendence. (Permanent members with pm listed: Bostan, Chyzak, Lairez.)
- *∂ifference*. This project, led by Olivier Bournez (Lix), started in November 2020. Its objective is to consider a novel approach in between the two worlds: discrete-oriented computations on the one side and differential equations on the other side. We aim at providing new insights on classical complexity theory, computability and logic through this prism and at introducing new perspectives in algorithmic methods for differential equations solving and computer science applications. (Permanent members with pm listed: Bostan, Chyzak.)

- *Tremplin ERC*. Pierre Lairez has been awarded a “tremplin” project by ANR. This will help him prepare an ERC project submission “10000 Digits, Foundations of transcendental methods in numerical algebraic geometry”.

7.4 Regional initiatives

- Alin Bostan submitted a PCRI-ANR proposal **EAGLE** – “Efficient Algorithms for Guessing, Summation, and Inequalities”. This is a bilateral ANR/FWF project between 2 computer algebra teams in France and 2 computer algebra teams in Austria. The Austrian co-leader is Manuel Kauers from Univ. Linz. The goal is to work together on four axes: structured and multivariate guessing, inequalities and D-finiteness, creative telescoping and applications in combinatorics, number theory and theoretical physics. The requested funding is of 770,000 euros in total.
- Alin Bostan is co-leader of the Amadeus (Campus France) bilateral project “Integer Sequences arising in Number Theory, Combinatorics and Physics” between France and Austria. The Austrian co-leader is Herwig Hauser (U. Vienna, Austria).

8 Dissemination

8.1 Promoting scientific activities

8.1.1 Scientific events: organisation

General chair, scientific chair

- Frédéric Chyzak was General Chair of the International Symposium on Symbolic and Algebraic Computation in 2021 (*ISSAC 2021*).
- Alin Bostan is part of the Scientific advisory board of the conference series *Effective Methods in Algebraic Geometry* (MEGA).
- Since 2020, for a period of 5 years, Alin Bostan is member of the steering committee of the *Journées Nationales de Calcul Formel* (JNCF), the annual meeting of the French computer algebra community.
- Alin Bostan is part of the scientific committee of the **GDR EFI** (“Functional Equations and Interactions”) dependent on the mathematical institute (INSMI) of the CNRS. The goal of this GDR is to bring together various research communities in France working on functional equations in fields of computer science and mathematics.

Member of the organizing committees

- Alin Bostan co-organizes, with Lucia Di Vizio, the *Séminaire Différentiel* between U. Versailles and Inria Saclay, with a bi-annual frequency.
- Alin Bostan co-organizes, with Lucia Di Vizio and Kilian Raschel the working group *Transcendance et Combinatoire*, at Institut Henri Poincaré (Paris), with a weekly frequency.
- Alin Bostan, together with Mohab Safey El Din, Bruno Salvy and Gilles Villard, started organizing a thematic program “Recent Trends in Computer Algebra (RTCA)”, to be held in 2023 in Paris and Lyon. The proposal has been accepted, the main funders being IHP (120,000 euros) and Labex Milyon (60,000 euros).

8.1.2 Scientific events: selection

Member of the conference program committees

- Alin Bostan has served in the program committee of FPSAC’21 (Formal Power Series and Algebraic Combinatorics).

Reviewer

- Alin Bostan has been a reviewer for *ISSAC'21 (International Symposium on Symbolic and Algebraic Computation)*, *FPSAC'21 (Formal Power Series and Algebraic Combinatorics)*, *MEGA'21 (Méthodes Effectives en Géométrie Effective)* and *Maple Conference 2021*.
- Sergey Yurkevich has been a reviewer for *ISSAC'21 (International Symposium on Symbolic and Algebraic Computation)*.

8.1.3 Journal

Member of the editorial boards

- Alin Bostan is on the editorial board of the *Journal of Symbolic Computation*.
- Alin Bostan is on the editorial board of the *Annals of Combinatorics*.
- Alin Bostan is on the editorial board of the *Maple Transactions*.
- Alin Bostan is on the editorial board of the *Bulletin of the Transilvania University of Braşov, Series III: Mathematics and Computer Science*.
- Frédéric Chyzak is on the editorial board of the *Journal of Systems Science and Complexity*.
- Guy Fayolle is associate editor of the journal *Markov Processes and Related Fields (MPRF)*.

Editors of special issues

- Frédéric Chyzak is co-editor (with George Labahn, University of Waterloo, Ontario, Canada) of a special issue in the *Journal of Symbolic Computation* after the ISSAC 2021 conference.
- Pierre Lairez is co-editor (with Anton Leykin, Georgia Tech) of a special issue in the *Journal of Symbolic Computation* after the ISSAC 2020 conference.

Reviewer - reviewing activities

- In 2021, Frédéric Chyzak was reviewer for a special issue after the conference ISSAC 2020 and for the *Journal of Symbolic Computation*.
- In 2021, Alin Bostan has been a reviewer for *Annals of Combinatorics*, *Experimental Mathematics*, *American Mathematical Monthly*, *Combinatorial Theory*, *Glasgow Mathematical Journal*, *European Journal of Combinatorics*, *Journal of Algebraic Combinatorics*.
- Guy Fayolle has been a reviewer for *Advances in Applied Probability*, *Markov Processes and Related Fields*, *Probability Theory and Related Fields*, *Queueing Systems: Theory and Applications*, *European Journal of Combinatorics*, *Journal of Statistical Physics*, *Physica A*, *Springer Science*.
- In 2021, Sergey Yurkevich has been a reviewer for *The American Mathematical Monthly*, *Experimental Mathematics*, *European Journal of Combinatorics*.

8.1.4 Invited talks

- Alin Bostan has been plenary speaker at the international conference **ISSAC'21** (46th International Symposium on Symbolic and Algebraic Computation, Saint Petersburg, Russia).
- Alin Bostan has been invited speaker at the international conference at the international conference “**Lattice Paths, Combinatorics and Interactions**”, Marseille, France.
- Alin Bostan gave invited talks at the minisymposium “**Generating series and confined lattice walks**” at **CanadAM 2021** (Canada), at the **Journée de rentrée de l'EDMH** (École doctorale de mathématiques Hadamard) at **IHES** and at the **École de Jeunes Chercheurs en Informatique Mathématique 2021**.

- Sergey Yurkevich gave a talk at the [Combinatorics and Arithmetic for Physics: special days CAP21](#) in [IHES](#).

8.1.5 Scientific expertise

- Guy Fayolle is scientific advisor and associate researcher at the *Robotics Laboratory of Mines Paris-Tech*.

8.1.6 Research administration

- Guy Fayolle is a member for *Computer System Modeling of the International Federation for Information Processing* (IFIP WG 7.3).

8.2 Teaching - Supervision - Juries

8.2.1 Teaching

- **Bachelor:**
 - Alexandre Goyer, *Mathématiques Générales (LSMA100)*, 60h, L1, Université de Versailles Saint-Quentin-en-Yvelines, France.
- **Master:**
 - Alin Bostan, *Algorithmes efficaces en calcul formel*, 36h, M2, MPRI, France.
 - Frédéric Chyzak, *Algorithmes efficaces en calcul formel*, 36h, M2, MPRI, France. (Also responsible for the [course](#).)
 - Pierre Lairez, *Algorithmes efficaces en calcul formel*, 9h, M2, MPRI, France.
 - Pierre Lairez, *Competitive programming (INF473A)*, TD, 40h, M2, École polytechnique, France.
 - Pierre Lairez, *Les bases de la programmation et de l'algorithmique (INF411)*, TD, 40h, M1, École polytechnique, France.

8.2.2 Supervision

- **Master internships:**
 - Alin Bostan and Frédéric Chyzak co-supervised together with Mohab Safey El Din (Sorbonne U.) the Master thesis of Hadrien Notarantonio on the topic “Calcul formel et systèmes polynomiaux pour la combinatoire”.
 - Alin Bostan and Pierre Lairez co-supervised together with Bruno Salvy (Inria Lyon) the Master thesis of Eric Pichon-Pharabod on the topic “Géométrie complexe et asymptotique automatique des sommes binomiales”.
- **PhD theses:**
 - Alin Bostan co-supervises together with Xavier Caruso (CNRS, IMB Bordeaux) the PhD thesis of Raphaël Pagès on the topic “Algorithms for factoring linear differential operators in positive characteristic”.
 - Alin Bostan co-supervises together with Herwig Hauser (U. Vienna, Austria) the PhD thesis of Sergey Yurkevich on the topic “Integer Sequences arising in Number Theory, Combinatorics and Physics”.
 - Alin Bostan and Frédéric Chyzak co-supervise together with Mohab Safey El Din (Sorbonne U.) the PhD thesis of Hadrien Notarantonio on the topic “Geometry-driven algorithms for the efficient solving of combinatorial functional equations”.
 - Frédéric Chyzak co-supervises together with Marc Mezzarobba (CNRS, Lix) the PhD thesis of Alexandre Goyer on the topic “Symbolic-numeric algorithms in differential algebra”.

- Pierre Lairez co-supervises together with Pierre Vanhove (CEA, IPhT) the PhD thesis of Eric Pichon-Pharabod on the topic “Periods in algebraic geometry: computation and application to Feynman’s integrals”.

8.2.3 Juries

- Frédéric Chyzak was examiner in the PhD jury of Mathilde Chenu, *Primitives cryptographiques résistantes aux ordinateurs quantiques basées sur les isogénies*, Institut Polytechnique de Paris, December 17, 2021.
- Alin Bostan has served as a referee of the PhD thesis of Manfred Buchacher, *Algorithms for the Enumeration of Lattice Walks*, U. Linz (Austria), November 8, 2021.
- Alin Bostan has served as the president of the PhD jury of Ali El Hajj, *Algorithmes symboliques pour l’étude et la résolution de systèmes d’équations fonctionnelles linéaires*, Limoges Univ., December 17, 2021.
- Alin Bostan has served as a reviewer in the mid-PhD examination of Antonin Leroux, *Algèbre de quaternion et cryptographie à base d’isogénies*, Ecole polytechnique.

9 Scientific production

9.1 Major publications

- [1] A. Benoit, A. Bostan and J. Van Der Hoeven. ‘Quasi-optimal multiplication of linear differential operators’. In: *FOCS 2012 - IEEE 53rd Annual Symposium on Foundations of Computer Science*. New Brunswick, United States: IEEE, Oct. 2012, pp. 524–530. DOI: [10.1109/FOCS.2012.57](https://hal.archives-ouvertes.fr/hal-00685401). URL: <https://hal.archives-ouvertes.fr/hal-00685401>.
- [2] A. Bostan, M. Bousquet-Mélou and S. Melczer. ‘Counting walks with large steps in an orthant’. In: *Journal of the European Mathematical Society* (2020). URL: <https://hal.archives-ouvertes.fr/hal-01802706>.
- [3] A. Bostan, X. Caruso, G. Christol and P. Dumas. ‘Fast Coefficient Computation for Algebraic Power Series in Positive Characteristic’. In: *ANTS-XIII - Thirteenth Algorithmic Number Theory Symposium*. Ed. by R. Scheidler and J. Sorenson. Vol. 2. Proceedings of the Thirteenth Algorithmic Number Theory Symposium (ANTS-XIII) 1. Madison, United States: Mathematical Sciences Publishers, July 2018, pp. 119–135. DOI: [10.2140/obs.2019.2-1](https://hal.archives-ouvertes.fr/hal-01816375). URL: <https://hal.archives-ouvertes.fr/hal-01816375>.
- [4] A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy and E. Schost. *Algorithmes Efficaces en Calcul Formel*. Voir la page du livre à l’adresse <https://hal.archives-ouvertes.fr/AECF/>. published by the Authors, 2017. URL: <https://hal.inria.fr/hal-01431717>.
- [5] A. Bostan, F. Chyzak, P. Lairez and B. Salvy. ‘Generalized Hermite Reduction, Creative Telescoping and Definite Integration of D-Finite Functions’. In: *ISSAC 2018 - International Symposium on Symbolic and Algebraic Computation*. New York, United States, July 2018, pp. 1–8. DOI: [10.1145/3208976.3208992](https://hal.inria.fr/hal-01788619). URL: <https://hal.inria.fr/hal-01788619>.
- [6] A. Bostan, F. Chyzak, M. Van Hoeij, M. Kauers and L. Pech. ‘Hypergeometric Expressions for Generating Functions of Walks with Small Steps in the Quarter Plane’. In: *European Journal of Combinatorics* 61 (2017), pp. 242–275. DOI: [10.1016/j.ejc.2016.10.010](https://hal.inria.fr/hal-01332175). URL: <https://hal.inria.fr/hal-01332175>.
- [7] P. Bürgisser, F. Cucker and P. Lairez. ‘Computing the Homology of Basic Semialgebraic Sets in Weak Exponential Time’. In: *Journal of the ACM (JACM)* 66.1 (Dec. 2018), pp. 1–30. DOI: [10.1145/3275242](https://hal.archives-ouvertes.fr/hal-01545657). URL: <https://hal.archives-ouvertes.fr/hal-01545657>.
- [8] F. Chyzak, T. Dreyfus, P. Dumas and M. Mezzarobba. ‘Computing solutions of linear Mahler equations’. In: *Mathematics of Computation* 87 (July 2018), pp. 2977–3021. DOI: [10.1090/mcom/3359](https://hal.inria.fr/hal-01418653). URL: <https://hal.inria.fr/hal-01418653>.

- [9] F. Chyzak, A. Mahboubi, T. Sibut-Pinote and E. Tassi. ‘A Computer-Algebra-Based Formal Proof of the Irrationality of $\zeta(3)$ ’. In: *ITP - 5th International Conference on Interactive Theorem Proving*. Vienna, Austria, 2014. URL: <https://hal.inria.fr/hal-00984057>.
- [10] P. Lairez. ‘Computing periods of rational integrals’. In: *Mathematics of Computation* 85 (Nov. 2016), pp. 1719–1752. DOI: [10.1090/mcom/3054](https://doi.org/10.1090/mcom/3054). URL: <https://hal.inria.fr/hal-00981114>.

9.2 Publications of the year

International journals

- [11] V. Andrejić, A. Bostan and M. Tatarevic. ‘Improved algorithms for left factorial residues’. In: *Information Processing Letters* 167 (2021), p. 3. DOI: [10.1016/j.ipl.2020.106078](https://doi.org/10.1016/j.ipl.2020.106078). URL: <https://hal.archives-ouvertes.fr/hal-02411741>.
- [12] A. Bostan. ‘A short proof of a non-vanishing result by Conca, Krattenthaler and Watanabe’. In: *Rendiconti del Seminario Matematico della Università di Padova* (2021). URL: <https://hal.inria.fr/hal-03216337>.
- [13] A. Bostan, M. Bousquet-Mélou and S. Melczer. ‘Counting walks with large steps in an orthant’. In: *Journal of the European Mathematical Society* 23.7 (2021), pp. 2221–2297. URL: <https://hal.archives-ouvertes.fr/hal-01802706>.
- [14] A. Bostan, F. Chamizo and M. Sundqvist. ‘On an Integral Identity’. In: *American Mathematical Monthly* 128.8 (2021), pp. 737–743. URL: <https://hal.inria.fr/hal-03084594>.
- [15] A. Bostan, T. Rivoal and B. Salvy. ‘Explicit degree bounds for right factors of linear differential operators’. In: *Bulletin of the London Mathematical Society* 53.1 (1st Feb. 2021), pp. 53–62. DOI: [10.1112/blms.12396](https://doi.org/10.1112/blms.12396). URL: <https://hal.archives-ouvertes.fr/hal-02154679>.
- [16] A. Bostan and S. Yurkevich. ‘A hypergeometric proof that Iso is bijective’. In: *Proceedings of the American Mathematical Society* (2021). URL: <https://hal.archives-ouvertes.fr/hal-03426008>.
- [17] A. Bostan and S. Yurkevich. ‘Fast Computation of the N -th Term of a q -Holonomic Sequence and Applications’. In: *Journal of Symbolic Computation* (2021). URL: <https://hal.inria.fr/hal-03084680>.
- [18] A. Bostan and S. Yurkevich. ‘On a Class of Hypergeometric Diagonals’. In: *Proceedings of the American Mathematical Society* (2021). URL: <https://hal.inria.fr/hal-03084672>.
- [19] C. Boutillier, K. Raschel and A. Bostan. ‘Martin boundary of killed random walks on isoradial graphs’. In: *Potential Analysis* (2021). URL: <https://hal.archives-ouvertes.fr/hal-02422417>.
- [20] G. Fayolle and R. Iasnogorodski. ‘Conditions for some non stationary random walks in the quarter plane to be singular or of genus 0’. In: *Markov Processes And Related Fields* 27.1 (Mar. 2021), p. 12. URL: <https://hal.inria.fr/hal-03008556>.
- [21] A. Goyer. ‘A Sage package for the symbolic-numeric factorization of linear differential operators’. In: *ACM Communications in Computer Algebra* 55.2 (June 2021), pp. 44–48. DOI: [10.1145/3493492.3493496](https://doi.org/10.1145/3493492.3493496). URL: <https://hal.inria.fr/hal-03414073>.

International peer-reviewed conferences

- [22] A. Bostan. ‘Computer Algebra in the Service of Enumerative Combinatorics’. In: *International Symposium on Symbolic and Algebraic Computation (ISSAC)*. Proceedings of ISSAC’21. Saint Petersburg, Russia: ACM Press, 2021, pp. 1–8. DOI: [10.1145/3452143.3465507](https://doi.org/10.1145/3452143.3465507). URL: <https://hal.archives-ouvertes.fr/hal-03427873>.
- [23] A. Bostan, M. Kauers and T. Verron. ‘The generating function of Kreweras walks with interacting boundaries is not algebraic’. In: *FPSAC’21 - Formal power series and algebraic combinatorics*. Vol. 85B. Sémin. Lothar. Combin. Art. 78. Ramat Gan, Israel, 5th July 2021, p. 12. URL: <https://hal.inria.fr/hal-03084659>.

- [24] A. Bostan and R. Mori. ‘A Simple and Fast Algorithm for Computing the N -th Term of a Linearly Recurrent Sequence’. In: SOSA’21 (SIAM Symposium on Simplicity in Algorithms). Alexandria, United States, 11th Jan. 2021. URL: <https://hal.inria.fr/hal-02917827>.
- [25] P. Lairez and M. Safey El Din. ‘Computing the dimension of real algebraic sets’. In: ISSAC 2021 - 46th International Symposium on Symbolic and Algebraic Computation. Saint-Petersbourg, Russia, 2021, pp. 257–264. DOI: [10.1145/3452143.3465551](https://doi.org/10.1145/3452143.3465551). URL: <https://hal.inria.fr/hal-03232068>.
- [26] R. Pagès. ‘Computing Characteristic Polynomials of p -Curvatures in Average Polynomial Time’. In: ISSAC 2021 - International Symposium on Symbolic and Algebraic Computation. Saint-Petersbourg / Virtual, Russia: ACM, 2021, pp. 329–336. DOI: [10.1145/3452143.3465524](https://doi.org/10.1145/3452143.3465524). URL: <https://hal.archives-ouvertes.fr/hal-03270585>.

Scientific books

- [27] A. Bostan and K. Raschel. *Transcendence in Algebra, Combinatorics, Geometry and Number Theory*. Vol. 373. Springer Proceedings in Mathematics & Statistics. Springer, 31st Oct. 2021, 560 pages. URL: <https://hal.archives-ouvertes.fr/hal-03423170>.

Scientific book chapters

- [28] S. Yurkevich. ‘Diagonal Representation of Algebraic Power Series: A Glimpse Behind the Scenes’. In: *Transcendence in Algebra, Combinatorics, Geometry and Number Theory*. Vol. TRANS19. Springer Proceedings in Mathematics & Statistics 373. Braşov, Romania: Springer, 3rd Nov. 2021, pp. 309–339. DOI: [10.1007/978-3-030-84304-5_13](https://doi.org/10.1007/978-3-030-84304-5_13). URL: <https://hal.archives-ouvertes.fr/hal-03150958>.

Reports & preprints

- [29] G. Alsmeyer, A. Bostan, K. Raschel and T. Simon. *Persistence for a class of order-one autoregressive processes and Mallows-Riordan polynomials*. 6th Dec. 2021. URL: <https://hal.inria.fr/hal-03469594>.
- [30] J. Berthomieu, A. Bostan, A. Ferguson and M. Safey El Din. *Gröbner bases and critical values: The asymptotic combinatorics of determinantal systems*. 30th Apr. 2021. URL: <https://hal.sorbonne-universite.fr/hal-03214157>.
- [31] A. Bostan, J. Tirrell, B. W. Westbury and Y. Zhang. *On some combinatorial sequences associated to invariant theory*. 26th Oct. 2021. URL: <https://hal.archives-ouvertes.fr/hal-03423136>.
- [32] A. Bostan and S. Yurkevich. *On the q -analogue of Pólya’s Theorem*. 11th Nov. 2021. URL: <https://hal.archives-ouvertes.fr/hal-03425997>.
- [33] G. Fayolle. *A note on the connection between product-form Jackson networks and counting lattice walks in the quarter plane*. 2021. URL: <https://hal.inria.fr/hal-02415746>.
- [34] G. Fayolle, S. Franceschi and K. Raschel. *On the stationary distribution of reflected Brownian motion in a non-convex wedge*. Mar. 2021. URL: <https://hal.archives-ouvertes.fr/hal-03150317>.
- [35] J. Steininger and S. Yurkevich. *An algorithmic approach to Rupert’s problem*. 27th Dec. 2021. URL: <https://hal.inria.fr/hal-03504867>.
- [36] S. Yurkevich. *The art of algorithmic guessing in gfun*. 13th Nov. 2021. URL: <https://hal.inria.fr/hal-03427179>.

9.3 Cited publications

- [37] Y. Abdelaziz, C. Koutschan and J.-M. Maillard. ‘On Christol’s conjecture’. In: *J. Phys. A* 53.20 (2020), 205201, 16 pages. DOI: [10.1088/1751-8121/ab82dc](https://doi.org/10.1088/1751-8121/ab82dc). URL: <https://doi.org/10.1088/1751-8121/ab82dc>.

- [38] M. Abramowitz and I. A. Stegun, eds. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Reprint of the 1972 edition. New York: Dover, 1992, pp. xiv+1046.
- [39] M. Armand, B. Grégoire, A. Spiwack and L. Théry. ‘Extending Coq with Imperative Features and its Application to SAT Verification’. In: *Interactive Theorem Proving, international Conference, ITP 2010, Edinburgh, Scotland, July 11–14, 2010, Proceedings*. Lecture Notes in Computer Science. Springer, 2010.
- [40] B. Beckermann and G. Labahn. ‘A uniform approach for the fast computation of matrix-type Padé approximants’. In: *SIAM J. Matrix Anal. Appl.* 15.3 (1994), pp. 804–823.
- [41] A. Benoit, F. Chyzak, A. Darrasse, S. Gerhold, M. Mezzarobba and B. Salvy. ‘The Dynamic Dictionary of Mathematical Functions (DDMF)’. In: *The Third International Congress on Mathematical Software (ICMS 2010)*. Ed. by K. Fukuda, J. van der Hoeven, M. Joswig and N. Takayama. Vol. 6327. Lecture Notes in Computer Science. 2010, pp. 35–41.
- [42] M. Boespflug, M. Dénès and B. Grégoire. ‘Full reduction at full throttle’. In: *First International Conference on Certified Programs and Proofs, Taiwan, December 7–9*. Lecture Notes in Computer Science. Springer, 2011.
- [43] S. Boldo, C. Lelay and G. Melquiond. ‘Improving Real Analysis in Coq: A User-Friendly Approach to Integrals and Derivatives’. In: *Certified Programs and Proofs*. Ed. by C. Hawblitzel and D. Miller. Vol. 7679. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 289–304. URL: http://dx.doi.org/10.1007/978-3-642-35308-6_22.
- [44] S. Boldo and G. Melquiond. ‘Flocq: A Unified Library for Proving Floating-point Algorithms in Coq’. In: *Proceedings of the 20th IEEE Symposium on Computer Arithmetic*. Tübingen, Germany, July 2011, pp. 243–252.
- [45] A. Bostan. ‘Algorithmes rapides pour les polynômes, séries formelles et matrices’. In: *Actes des Journées Nationales de Calcul Formel*. Les cours du CIRM, tome 1, numéro 2. Luminy, France, 2010, pp. 75–262. URL: http://ccirm.cedram.org:80/ccirm-bin/fitem?id=CCIRM_2010__1_2_75_0.
- [46] A. Bostan, S. Boukraa, S. Hassani, J.-M. Maillard, J.-A. Weil and N. Zenine. ‘Globally nilpotent differential operators and the square Ising model’. In: *J. Phys. A: Math. Theor.* 42.12 (2009), p. 50. URL: <http://dx.doi.org/10.1088/1751-8113/42/12/125206>.
- [47] A. Bostan, S. Chen, F. Chyzak and Z. Li. ‘Complexity of creative telescoping for bivariate rational functions’. In: *ISSAC’10: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*. New York, NY, USA: ACM, 2010, pp. 203–210. URL: <http://doi.acm.org/10.1145/1837934.1837975>.
- [48] A. Bostan, F. Chyzak, M. van Hoeij and L. Pech. ‘Explicit formula for the generating series of diagonal 3D rook paths’. In: *Sém. Loth. Comb.* B66a (2011), p. 27. URL: <http://www.emis.de/journals/SLC/wpapers/s66bochhope.html>.
- [49] A. Bostan, F. Chyzak, G. Lecerf, B. Salvy and É. Schost. ‘Differential equations for algebraic functions’. In: *ISSAC’07: Proceedings of the 2007 international symposium on Symbolic and algebraic computation*. Ed. by C. W. Brown. ACM Press, 2007, pp. 25–32. URL: <http://dx.doi.org/10.1145/1277548.1277553>.
- [50] A. Bostan and M. Kauers. ‘The complete generating function for Gessel walks is algebraic’. In: *Proceedings of the American Mathematical Society* 138.9 (2010). With an appendix by Mark van Hoeij, pp. 3063–3078.
- [51] F. Chyzak. ‘An extension of Zeilberger’s fast algorithm to general holonomic functions’. In: *Discrete Math.* 217.1-3 (2000). Formal power series and algebraic combinatorics (Vienna, 1997), pp. 115–134.
- [52] F. Chyzak, M. Kauers and B. Salvy. ‘A Non-Holonomic Systems Approach to Special Function Identities’. In: *ISSAC’09: Proceedings of the Twenty-Second International Symposium on Symbolic and Algebraic Computation*. Ed. by J. May. 2009, pp. 111–118. URL: <http://dx.doi.org/10.1145/1576702.1576720>.

- [53] F. Chyzak and B. Salvy. ‘Non-commutative elimination in Ore algebras proves multivariate identities’. In: *J. Symbolic Comput.* 26.2 (1998), pp. 187–227.
- [54] T. Coquand and G. P. Huet. ‘The Calculus of Constructions’. In: *Inf. Comput.* 76.2/3 (1988), pp. 95–120. URL: [http://dx.doi.org/10.1016/0890-5401\(88\)90005-3](http://dx.doi.org/10.1016/0890-5401(88)90005-3).
- [55] T. Coquand and C. Paulin-Mohring. ‘Inductively defined types’. In: *Proceedings of Colog’88*. Ed. by P. Martin-Löf and G. Mints. Vol. 417. Lecture Notes in Computer Science. Springer-Verlag, 1990.
- [56] D. Delahaye and M. Mayero. ‘Dealing with algebraic expressions over a field in Coq using Maple’. In: *J. Symbolic Comput.* 39.5 (2005). Special issue on the integration of automated reasoning and computer algebra systems, pp. 569–592. URL: <http://dx.doi.org/10.1016/j.jsc.2004.12.004>.
- [57] G. Fayolle, R. Iasnogorodski and V. Malyshev. *Random walks in the quarter plane*. Springer International Publishing, 2017. DOI: [10.1007/978-3-319-50930-3](https://doi.org/10.1007/978-3-319-50930-3).
- [58] F. Garillot, G. Gonthier, A. Mahboubi and L. Rideau. ‘Packaging Mathematical Structures’. In: *Theorem Proving in Higher-Order Logics*. Ed. by S. Berghofer, T. Nipkow, C. Urban and M. Wenzel. Vol. 5674. Lecture Notes in Computer Science. Springer, 2009, pp. 327–342.
- [59] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. 2nd. New York: Cambridge University Press, 2003, pp. xiv+785.
- [60] G. Gonthier. ‘Formal proofs—the four-colour theorem’. In: *Notices of the AMS* 55.11 (2008), pp. 1382–1393.
- [61] G. Gonthier and A. Mahboubi. ‘An introduction to small scale reflection in Coq’. In: *Journal of Formalized Reasoning* 3.2 (2010), pp. 95–152.
- [62] G. Gonthier, A. Mahboubi and E. Tassi. *A Small Scale Reflection Extension for the Coq system*. Anglais. Rapport de recherche RR-6455. INRIA, 2008. URL: <http://hal.inria.fr/inria-00258384>.
- [63] G. Gonthier and E. Tassi. ‘A language of patterns for subterm selection’. In: *ITP*. Vol. 7406. LNCS. 2012, pp. 361–376.
- [64] B. Grégoire and A. Mahboubi. ‘Proving Equalities in a Commutative Ring Done Right in Coq’. In: *Theorem Proving in Higher Order Logics, 18th International Conference, TPHOLs 2005, Oxford, UK, August 22-25, 2005, Proceedings*. Vol. 3603. Lecture Notes in Computer Science. Springer, 2005, pp. 98–113.
- [65] T. Hales. ‘Formal proof’. In: *Notices of the AMS* 55.11 (2008), pp. 1370–1380.
- [66] J. Harrison. ‘A HOL Theory of Euclidean space’. In: *Theorem Proving in Higher Order Logics, 18th International Conference, TPHOLs 2005*. Ed. by J. Hurd and T. Melham. Vol. 3603. Lecture Notes in Computer Science. Oxford, UK: Springer-Verlag, 2005, pp. 114–129.
- [67] J. Harrison. ‘A Machine-Checked Theory of Floating Point Arithmetic’. In: *Theorem Proving in Higher Order Logics: 12th International Conference, TPHOLs’99*. Ed. by Y. Bertot, G. Dowek, A. Hirschowitz, C. Paulin and L. Théry. Vol. 1690. Lecture Notes in Computer Science. Nice, France: Springer-Verlag, 1999, pp. 113–130.
- [68] J. Harrison. ‘Formalizing an analytic proof of the prime number theorem’. In: *Journal of Automated Reasoning* 43 (2009). Dedicated to Mike Gordon on the occasion of his 60th birthday, pp. 243–261.
- [69] J. Harrison. *Theorem proving with the real numbers*. CPHC/BCS distinguished dissertations. Springer, 1998.
- [70] J. Harrison and L. Théry. ‘A Skeptic’s Approach to Combining HOL and Maple’. In: *J. Autom. Reason.* 21.3 (Dec. 1998), pp. 279–294. URL: <http://dx.doi.org/10.1023/A:1006023127567>.
- [71] E. Johansson. *Another Mathematica bug*. Article on personal blog. 2009. URL: <http://fredrik-j.blogspot.fr/2009/07/another-mathematica-bug.html>.
- [72] C. Koutschan. ‘A fast approach to creative telescoping’. In: *Math. Comput. Sci.* 4.2-3 (2010), pp. 259–266. DOI: [10.1007/s11786-010-0055-0](https://doi.org/10.1007/s11786-010-0055-0). URL: <http://dx.doi.org/10.1007/s11786-010-0055-0>.

- [73] A. Mahboubi. ‘Implementing the cylindrical algebraic decomposition within the Coq system’. In: *Mathematical Structures in Computer Science* 17.1 (2007), pp. 99–127.
- [74] *Computer Algebra Errors*. Article in mathematics blog *MathOverflow*. URL: <http://mathoverflow.net/questions/11517/computer-algebra-errors>.
- [75] R. Matuszewski and P. Rudnicki. ‘Mizar: the first 30 years’. In: *Mechanized Mathematics and Its Applications* 4 (2005).
- [76] M. Mayero. ‘Problèmes critiques et preuves formelles’. Habilitation à Diriger des Recherches. Université Paris 13, 2012.
- [77] M. Mezzarobba. ‘NumGfun: a package for numerical and analytic computation and D-finite functions’. In: *ISSAC 2010—Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*. New York: ACM, 2010, pp. 139–146. DOI: [10.1145/1837934.1837965](https://doi.org/10.1145/1837934.1837965). URL: <http://dx.doi.org/10.1145/1837934.1837965>.
- [78] F. W. J. Olver, D. W. Lozier, R. F. Boisvert and C. W. Clark, eds. *NIST Handbook of mathematical functions*. Cambridge University Press, 2010.
- [79] P. Paule and M. Schorn. ‘A Mathematica version of Zeilberger’s algorithm for proving binomial coefficient identities’. In: *J. Symbolic Comput.* 20.5-6 (1995). Symbolic computation in combinatorics Δ_1 (Ithaca, NY, 1993), pp. 673–698. DOI: [10.1006/jsco.1995.1071](https://doi.org/10.1006/jsco.1995.1071). URL: <http://dx.doi.org/10.1006/jsco.1995.1071>.
- [80] B. Petersen. *Maple*. Personal web site.
- [81] P. Rudnicki and A. Trybulec. ‘On the Integrity of a Repository of Formalized Mathematics’. In: *Proceedings of the Second International Conference on Mathematical Knowledge Management*. MKM ’03. London, UK: Springer-Verlag, 2003, pp. 162–174. URL: <http://dl.acm.org/citation.cfm?id=648071.748518>.
- [82] B. Salvy and P. Zimmermann. ‘Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable’. In: *ACM Trans. Math. Software* 20.2 (1994), pp. 163–177.
- [83] N. J. A. Sloane and S. Plouffe. *The Encyclopedia of Integer Sequences*. Academic Press, San Diego, 1995.
- [84] The Coq Development Team. *The Coq Proof Assistant: Reference Manual*. URL: <http://coq.inria.fr/doc/>.
- [85] The Mathematical Component Team. *A Formalization of the Odd Order Theorem using the Coq proof assistant*. 2012. URL: <http://www.msr-inria.fr/projects/mathematical-components/>.
- [86] L. Théry. ‘A Machine-Checked Implementation of Buchberger’s Algorithm’. In: *J. Autom. Reasoning* 26.2 (2001), pp. 107–137. URL: <http://dx.doi.org/10.1023/A:1026518331905>.
- [87] K. Wegschaider. ‘Computer generated proofs of binomial multi-sum identities’. Diplomarbeit. RISC, J. Kepler University, May 1997, p. 99.
- [88] S. Wolfram. *Mathematica: A system for doing mathematics by computer (2nd ed.)* Addison-Wesley, 1992.
- [89] T. Yu and J. Chen. *On the Uniqueness of Clifford Torus with Prescribed Isoperimetric Ratio*. Technical Report [arXiv:2003.13116](https://arxiv.org/abs/2003.13116) [math.DG]. 2020. arXiv: [2003.13116](https://arxiv.org/abs/2003.13116) [math.DG].
- [90] D. Zeilberger. ‘A holonomic systems approach to special functions identities’. In: *J. Comput. Appl. Math.* 32.3 (1990), pp. 321–368.
- [91] D. Zeilberger. *Opinion 94: The Human Obsession With “Formal Proofs” is a Waste of the Computer’s Time, and, Even More Regretfully, of Humans’ Time*. 2009. URL: <http://www.math.rutgers.edu/~zeilberg/Opinion94.html>.
- [92] D. Zeilberger. ‘The method of creative telescoping’. In: *J. Symbolic Comput.* 11.3 (1991), pp. 195–204.