Validated symbolic-numeric algorithms and practical applications in aerospace

Mioara Maria Joldes

Algorithmes symboliques-numeriques validés et applications au domaine spatial

Validated symbolic-numeric algorithms and practical applications in aerospace

Mémoire d'habilitation à diriger les recherches

présenté le 26 juin 2019 à

Université de Toulouse Paul Sabatier

par

Mioara Joldes

Chargée de recherche au CNRS

après avis des rapporteurs :

AMPARO GIL, Professeur, Universidad de Cantabria

PHILIPPE LANGLOIS, Professeur, Université de Perpignan Via Domitia

MIHAI PUTINAR, Professeur, University of California at Santa Barbara

devant le jury formé de :

DENIS ARZELIER, Directeur de recherche CNRS, LAAS-CNRS

AMPARO GIL, Professeur, Universidad de Cantabria, Santander

MANUEL KAUERS, Professeur, Johannes Kepler University, Linz

PHILIPPE LANGLOIS, Professeur, Université de Perpignan Via Domitia

MIHAI PUTINAR, Professeur, University of California at Santa Barbara

PING TAK PETER TANG, Senior Principal Engineer, Intel Corporation

MASSIMILIANO VASILE, Professeur, University of Strathclyde, Glasgow

Acknowledgements

From the Shakuhachi flute's calm to the tempestuous "Monsieur, vous êtes un con !", from the edge of obsolescence LaTeX books to Bourdieu, from endless debates on world best pastries to LMIs use and limits, from tic-toc-Matlab to simplify-Maple, from recurrences to recurrent issues, from collision probabilities to completely remaking the whole world while drinking (a wide-edge) cup of coffee, Denis inspired, enlightened and helped me starting day one, in such a genuine, original and generous way, that still may be completely incomprehensible for the pragmatic and cynic side of some (even myself). Knowing how hard it is to teach morality to a Carpathian hyena (but also how stubborned some can be when not installing Ubuntu as their main operating system...), I hope we will continue working together, it has been a pleasure and an honor to have you as my parrain. Merci beaucoup, Denis!

I would like to express my gratitude to the reviewers of this manuscript: Amparo Gil, Philippe Langlois and Mihai Putinar. I thank them for their great work and useful suggestions, for their understanding and availability regarding the tight schedule imposed. In particular, my thanks go to Mihai: a tiny part of his and Jean-Bernard's work is already a great inspiration for some very recent of my research attempts. Thank you Mihai for managing to enclose such complicated concepts in few simple words, a pinch of history and a global vision of so, so much mathematics.

I want to thank Manuel Kauers, Peter Tang and Massimiliano Vasile, for having accepted to participate in my jury, for making a place for this event in their busy schedules, for our discussions and I hope we will have the chance to collaborate more in the future.

My ability to supervise research works could be genuinely assessed by those who supported their "sefa tiranica" for at least three years. It was a great honor, challenge and most of the time total fun, to supervise Valentina, Paulo and Florent. I want to thank them for their hard work, brilliant minds and such motivating and refreshing attitudes. I learned so many things from them! So far, in my research life, this was the greatest part. I would also like to thank Olivier, Norbi, Bea, Damien, Marco Felice and Ana-Maria for having spent some time as interns. Thank you all, for having put a chunk of your trust, time, stubbornness and dedication in my hands (just hoping I did not do too much damage to it).

Moreover, I would like to thank my collaborators and co-authors who helped me with their hard work and great ideas and to the members of MAC team for having made my work there an unforgettable experience. I also thank Cathy and Anaïs for having taken care of all the administrative matters related to this defense.

Special thanks to my favorite Lyonnais: Nicolas, Jean-Michel and Bruno, for their endless support, fast-relax, great restaurants, good wine, and the interesting blend of computer arithmetic, computer algebra and approximation, that I enjoy so much discussing with them. Extra special mention to Nicolas for a million (+1) reasons. Also, Warwick, thanks for being a great friend and for sharing bright ideas! I would also like to also thank Sophie and Juan-Carlos for their interesting collaboration and projects and also, for listening to bunches of crazy formulas each time we meet at CNES.

Last but not least, I am so lucky and glad to have Bogdan and my family, who endure, motivate and trust the stressed-and-grumpy-me, each time this work overwhelmed me both by its rigid scientific nature or by the daily dilemma, I quote: "Do you want to be an eagle, or do you want to be a shitbird?". Time will tell. I would like to put down a final word for *Mosu@Soho*, there will always be a small thread running for you in the background of my busy, serious, modern and pragmatic-wannabe mind.

Contents

Introduction

This document surveys the framework, the goals and the main outcomes of my research in the last few years, together with new perspectives. The obtained results mainly correspond to my research plan proposed in my CNRS entrance file, back in 2012. They are outlined by insisting on the underlying ideas, using simple examples and algorithms, without imposing to the reader to go over the technical proofs of the corresponding articles. For completeness, complementary references are pointed out and all my research works are either appended to the text or available online in open access platforms[1].

Research Context. The general context of my research is the field of *rigorous computing* (sometimes called *validated computing* as well) which uses numerical computations, yet is able to provide rigorous mathematical statements about the obtained result, such as sure and reasonably tight, error bounds. The goal is to bridge the gap between pure mathematics and numerical computing. Obviously, this implies sometimes walking in equilibrium on a thin rope. Take for instance the obvious question: **can we trust the numerics?**

On one end of the spectrum, and from a "(too) traditional pure mathematics" point of view, the short answer is NO. The long answer is *Maybe, but one has to be careful, because...*

Usually, numbers are stored and manipulated in *finite precision* (floating-point being among most common formats) and they represent only a finite subset of the real axis. For each basic computation (addition, multiplication) a rounding error may occur. Then, most numerical methods introduce also so called *method approximation errors*. Finally, the solved mathematical problem is only a simplified (sometimes linearized) model of the real world, so *model disturbances, uncertainties, nonlinearities* have to be considered. Even for rather simple queries, like values of definite integrals, there is no symbolic *closed-formula* for the result. If well-known software (e.g. Matlab, Mathematica, Sage) provide numerical results, however one is often unaware of mentions like the following in Mathematica Book [227]: *when Mathematica does a numerical integral, the only information it has about your integrand is a sequence of numerical values for it. If you give a sufficiently pathological integrand, Mathematica may simply give you the wrong answer.*

Hence, these numerical computations alone do not constitute a proof of correctness of the result obtained in general.

On the other end of the spectrum, in the wild world of real applications, the results and performances of numerical computer algorithms become more and more striking regarding both their efficiency, large scale, and the current ubiquitous tendency of employing *AI* algorithms in every field.

In the middle, there lie the computer-aided mathematics or computer-assisted proofs, which have seen the advent in the last 20 years [213, 88, 90], and whose trend is getting more and more important [60]. Examples where numeric results accuracy has to be guaranteed, arise in attitude and orbit control systems (AOCS) of spacecraft or surgical robots, or when studying chaotic dynamical systems like strange attractors or complicated astrodynamics like long-term stability of the solar system [122]. Traditional validated computing [214] methods are based on arbitrary precision libraries, interval arithmetic, or formal proofs. Interval arithmetic [152] is the simplest set arithmetic, which always returns an interval guaranteed to contain the correct result. Such methods were used in computer-assisted proofs in dynamical systems [213, 38, 67, 216], optimization for uncertain chemical systems [155]. However, so far, they are employed on a case-by-case basis, and obtaining both efficient approximations and effective error bounds, for generic classes of numerical problems, remains a difficult task. Some of my results on this topic are provided in what follows. Let us mention that while some of them have been formally

[1] They can be found on http://hal.archives-ouvertes.fr/

proved in joint works, this manuscript does not deal with the field of formal proof assistants. To lure the reader into the *validated computing* field, what can be better than a story about satellites? So here it starts.

From validated numerics to spacecraft trajectories and back. Recently, I worked on evaluating an integral appearing in a critical space project: since the number of space debris has drastically increased, adequate mitigation and collision avoidance strategies are vital for active satellites [J5]. While relative debris–satellite positions and velocities are only approximately known, the collision risk has to be both fast and reliably assessed. If the evaluated risk is sufficiently high, a collision avoidance maneuver is decided, but each such maneuver reduces the remaining satellite fuel and thus its active in-orbit life. Yet, a wrong computation which underestimates the risk, could result in the satellite loss (such cases actually occurred in the past e.g., 2009, Feb. 10, a collision occurred between Iridium 33 and Cosmos 2251 satellites, although the predicted minimum distance of close approach was of 584 m [44]).
The majority of previous solutions [5, 48, 171] could not be used without further approximations and were unable to guarantee the accuracy requirements. This is because usually either numerical integration schemes or truncated power series were used, but no rigorous proof regarding the method's convergence rate was given. The truncation orders or discretization steps were fixed *by trial and error* or by comparing against other numerical tools which might offer higher accuracy. In contrast, our solution, readily implemented by CNES (French Space Agency) exploits both the power of computer algebra and numerical evaluation tools which result in a method that is not only reliable (the number of guaranteed correct digits is user-input) but also faster than quadrature schemes.

Beside this particular example, I designed validated methods employed in the domain of mathematical library design [J3, J6, C1], dynamical systems [J7] or applications to robust space missions [J5, C8, J2, J1].

Research themes and structure. The structure of my research was naturally built starting with my PhD thesis which was centered on validated computations for mathematical library design [R3] and made use of some computer algebra techniques. Subsequently, during my post-doctorate, I focused more on high performance computer arithmetic and *a posteriori* validation techniques for proving certain properties of chaotic dynamical systems. Finally, starting with January 2013, I joined MAC –Methods and Algorithms in Control– Team, whose main objectives include providing constructive theoretical conditions for characterizing solutions to various control and optimization problems, while producing effective computational algorithms. In this context, my focus is mainly on developing symbolic-numeric objects and validated algorithms for improving both the numerical reliability and the speed of optimal control methods applied in aerospace.

More specifically, my research is structured in 3 interconnected layers:

(i) At the numerics level, we are interested in improving the quality (in terms of accuracy, speed, reliability of software, etc.) of the arithmetic available on computers. In particular, we focus on high-precision arithmetic algorithms, using as basic building blocks the available operators for floating-point arithmetic. We are also interested in problems related to the efficient and reliable implementation and evaluation in fixed-precision of elementary and special functions.

(ii) At the symbolic-numeric level, we focus on Rigorous Polynomial Approximations (RPAs), which are formed by a polynomial approximation and a rigorous error bound. We consider efficient algorithms for constructing and manipulating RPAs with the purpose of validating solutions using them. We make use of an important class of functions which are *solutions of linear differential equations with polynomial coefficients* also called *differentially finite or D-finite functions* for short. These functions have very interesting symbolic and numeric properties that allow, for instance, for efficient approximation algorithms based on Taylor, Chebyshev and other series expansions.

(iii) We apply these previously described tools in dynamical systems, optimal control and the aerospace domain.

Document outline. In what follows, we overview several results published during January 2013-February 2019. My exhaustive publications list is recalled at the end of this manuscript: references I co-authored are referenced with a specific prefix: B for book, J for journal, C for proceedings in (international) conferences, NC for national conference and R for research report, preprint or other unpublished documents.

- Chapter 1 mainly focuses on computer arithmetic results. Namely, several fast and certified algorithms which implement high-precision arithmetic operations are presented. They use as basic building blocks, the floating-point arithmetic operators (addition, multiplication, division, square root) available in hardware (single or double precision) in common processors nowadays. They are collected in an arithmetic library CAMPARY (CudA Multiple Precision ARithmetic librarY), especially tuned both for CPU or GPU computations, which have a highly parallel computing structure. We analyze its performance on two applications: one concerns the long term iteration of a classical chaotic dynamical system, the Hénon map; the other considers the speedup obtained by using CAMPARY instead of other high-precision libraries in an semidefinite programming (SDP) solver on ill-conditionned instances. The results surveyed in this chapter were published in [B1, J6, J3, J7, C6, C9, C5, C14, C11].

- Chapter 2 considers the framework of symbolic-numeric computations. We present several techniques which mix structural properties, coming from the symbolic field of Linear Ordinary Differential Equations (LODEs) with polynomial coefficients, with efficient numerical routines coming from optimization or approximation theory. Moreover, as mentioned above, efficient algorithms for RPAs are described in several contexts, with a systematic analysis on their operations complexity. These include the efficient evaluation of certain integrals, or the computation of validated solutions of LODEs based on truncated Chebyshev series expansions [J2, J4, J5, C10]. The more general setting of *multivariate D-finite functions* is exploited in an inverse problem involving measures with holonomic densities and support with real algebraic boundary [C2].

- The theoretical tools developed above are inspired by problems coming from optimal control and aerospace. In Chapter 3, we present two such problems and how the solutions based on our tools improve on classical ones. Firstly, the problem of the efficient and reliable orbital collision risk assessment and mitigation is discussed [J5, C12, C13, R2]. In particular, we give a new method [J5] to compute the orbital collision probability between two spherical objects involved in a short-term encounter, under Gaussian uncertainty. Secondly, we focus on efficient and validated algorithms for impulsive spacecraft rendezvous [C8, J2, C3]. This general aim of meeting two spacecraft, under several position, time, or fuel minimization constraints, is usually formulated as an optimal control problem. In the specific case of fixed-time minimum-fuel rendezvous, with a linear impulsive setting, we proposed a solution via an efficient numerical iterative algorithm. This comes from the formulation of a semi-infinite convex optimization problem (SICP), using a relaxation scheme and duality theory in normed linear spaces. The obtained solution is *a posteriori* certified based on RPAs.

  Incidentally, we observed that the SICP-based formulation also provides a solution to a computer arithmetic problem concerning the mathematical function implementation in machine. This very recent result [C1] closes the circle of this overview.

- Chapter 4 puts in perspective the obtained results, in the context of a continuous blend of ideas and techniques corresponding to the previously described items (i) - (iii). It structures my next research goals and overviews some complementary references which can help in achieving them. These ideas are also intended to be part of a grant proposal at some future time.

Finally, a CV incorporates an overview of my teaching, developed software, students supervision, conference organization, research grants/projects I am part of.

Chapter 1

Fast and Certified Multiple-Precision Arithmetic

Computer arithmetic is devoted to the study of arithmetic algorithms and the implementation of arithmetic operations and functions, either in software or hardware. Improvements are sought in terms of accuracy, speed or reliability for the arithmetic available on computers, processors, dedicated or embedded chips.

A part of my research concerned the development of high-precision arithmetic algorithms, using as basic building blocks the available arithmetic operators (addition, multiplication, division, square root) for fixed-precision floating-point arithmetic. The idea of creating a high-performance multiple-precision arithmetic library originated from our studies, with W. Tucker (Prof. Uppsala University, Sweden), on some chaotic dynamical systems. Subsequently, with J.-M. Muller (CNRS Researcher, LIP, Lyon, France) and our student V. Popescu, who defended her PhD thesis [176] in July 2017, we made important progress on this project:

- We *doubled* the available precision by representing a real number as the unevaluated sum of two floating-point numbers. We revisited and proposed new algorithms, insisting on their correctness proofs together with effective error bounds as well as on their regular and sufficiently simple structure [J3]. As a by-product, these algorithms are amenable to formalization, which is an on-going work of L. Rideau (Inria Researcher, Sophia-Antipolis, Nice, France).

- For multiple-precision, numbers are represented as the unevaluated sum of several (more than two) floating-point numbers. In several contributions [J6, C9, C14, C6], we proposed new algorithms for arithmetic operations designed to fit different needs a user might have: either very tight error bounds on the results (some of which were also formally proved), or "quick-and-dirty" results. In some of these works we also collaborated with O. Marty (student intern), S. Collange (researcher at Inria Rennes) and S. Boldo (researcher at Inria, Orsay, France). These algorithms were collected in an open-source library [C11].

- This library was put to test on two applications: one in the field of chaotic dynamical systems [J7], the other on high-accuracy semidefinite programming [C5].

- Part of the above results were also published in the second edition of the Handbook of FP arithmetic [B1].

In what follows, some preliminary notions of floating-point arithmetic are given, before entering the actual results description in Section 1.2.

1.1 A few Floating-Point notions

Definition 1.1.1 (Floating-point number)**.** A binary *floating-point* (FP) number of precision p is a number of the form $M \cdot 2^{e-p+1}$, where M is an integer of absolute value less than or equal to $2^p - 1$ and e is an integer such that $e_{\min} \leqslant e \leqslant e_{\max}$, where the extremal exponents $e_{\min}$ and $e_{\max}$ are constants of the floating-point format being considered, and with the additional requirement that, unless $e = e_{\min}$, one has $2^{p-1} \leqslant |M|$.

An FP number is called *subnormal* when $e = e_{\min}$ and $|M| \leqslant 2^{p-1} - 1$, otherwise it is called *normal*.

Currently, most floating-point calculations are done in single precision (also called binary32) or double precision (also called binary64) arithmetic. In binary32 arithmetic, $p = 24$, $e_{\min} = -126$, and $e_{\max} = 127$, while in binary64 arithmetic, $p = 53$, $e_{\min} = -1022$, and $e_{\max} = 1023$. Most available processors offer very fast implementations of FP arithmetic in these two formats, and comply with the IEEE 754-2008 standard for FP arithmetic [100]. The IEEE 754-2008 standard defines five *rounding functions* (round downwards, upwards, towards zero, to the nearest ties to even, and to the nearest ties to away). When an arithmetic operation is performed, the result must be that which would be obtained by performing the operation with *infinite precision* and then applying the rounding function. Such an operation is said to be *correctly rounded*.

In such a case, when approximating a nonzero real number $x \in \mathbb{R}$ by $\mathrm{RN}(x)$, with RN being the round to nearest rounding mode, the relative error satisfies:

$$\left| \frac{x - \mathrm{RN}(x)}{x} \right| \leqslant 2^{-p},$$

assuming no underflow[1] /overflow occurs. When $\mathrm{RN}(x) = x = 0$, the relative error is considered to be 0.

When expressing errors of *nearly atomic* functions (arithmetic operations, elementary functions, small polynomials, sums, dots products, etc.) it is advisable and frequently more accurate to do it in terms of *the weight of the last bit of the significand*, which is defined in [154]:

Definition 1.1.2 (Goldberg's definition, extended to reals)**.** If $|x| \in [2^{e_x}, 2^{e_x+1})$, then the *unit in the last place* of x is

$$\mathrm{ulp}(x) = 2^{\max(e_x, e_{\min}) - p + 1}.$$

Roughly speaking, from the above definitions *one has 53 correct bits or 15 correct decimal digits, when rounding a real number to* binary64*, when no overflow/underflow occurs.* Formally, this is expressed by the following notion, which is widely used in numerical analysis [96]:

Definition 1.1.3 (Unit roundoff)**.** The *unit roundoff* $\mathbf{u}$ of a precision-p, binary FP system is

$$\mathbf{u} = \begin{cases} \frac{1}{2}\,\mathrm{ulp}(1) & = & 2^{-p} & \text{in round-to-nearest mode,} \\ \mathrm{ulp}(1) & = & 2^{1-p} & \text{in directed rounding modes.} \end{cases}$$

Note that the distance between 1 and its FP successor is $\mathrm{ulp}(1) = 2^{1-p}$, which is also called *machine epsilon*. In particular, this is what the Matlab function `eps` returns. Hence, this coincides with the definition of unit roundoff for directed rounded modes, but not for rounding to nearest.

However, several computing problems require higher precision (also called *multiple precision*), up to a few hundred bits. Examples include problems in the field of chaotic dynamical systems (like the long-term stability of the solar system [122], long-term iteration of the Lorenz attractor [12], the study of strange attractors such as the Hénon attractor [J7]), ill-posed semi-definite positive optimization problems that appear in quantum chemistry or quantum information [200]. We also mention the use of higher precision in computational geometry, where several of the techniques we use were introduced for the first time [178].

[1] Let us say, as does the IEEE 754 standard, that an operation underflows when the result is subnormal and inexact.

1.2 Extending the precision

There exist multiple-precision libraries that allow the manipulation of very high-precision numbers, but their generality (they are able to handle numbers with millions of digits) typically comes at the expense of performance. The reader can for instance refer to [34] for state-of-the-art algorithms in this case, and to MPFR Library for efficient implementations [74].

To harness the availability and efficiency of the hardware implementations of the standard, our approach consisted in representing higher precision numbers as **floating-point expansions**. These are unevaluated sums of several floating-point numbers of different magnitudes. Such a representation is possible thanks to the availability of *error-free transforms*, namely algorithms that allow to compute the error of a FP addition or multiplication exactly, taking the rounding mode into account.

More specifically, the sum of two floating-point numbers can be represented *exactly* (in the sense of dyadic numbers) as a floating-point number which is the correct rounding of the sum, plus another floating-point number corresponding to the remainder. Under certain assumptions, this decomposition can be computed at a very low cost. For example, when RN is round-to-nearest ties to even and $|a| \geqslant |b|$, the following simple algorithm (called "Fast2Sum" in the literature [62] [B1, Chap. 4]), returns the FP number s nearest $a + b$ *and the error of that FP operation*, namely $t = (a + b) - s$:

Algorithm 1 The Fast2Sum algorithm.

Input: $a \geqslant b$
$s \leftarrow \mathrm{RN}(a + b)$
$z \leftarrow \mathrm{RN}(s - a)$
$t \leftarrow \mathrm{RN}(b - z)$

It is thus possible, in this case, to represent and store the exact sum, even in the presence of roundings at the floating-point level. A slightly more complicated algorithm (2Sum), due to Knuth [111] and Møller [150], deals with the case where $|a|$ is not necessarily larger than $|b|$.

If an FMA operator is available[2], then similarly, an algorithm called 2ProdFMA [B1, Chap.4.4] returns the FP number t nearest ab, and the error e of that FP multiplication, namely $ab - t$.

Algorithm 2 The 2ProdFMA algorithm.

$t \leftarrow \mathrm{RN}(ab)$
$e \leftarrow \mathrm{RN}(ab - t)$

Since these algorithms return their result as an unevaluated sum (x_h, x_ℓ) of floating-point numbers (with $|x_\ell|$ much smaller than $|x_h|$): more precisely, $|x_\ell| \leqslant \frac{1}{2}\,\mathrm{ulp}(x_h)$), a natural idea is to extend the precision based on such unevaluated sums of FP numbers. This kind of representation is called double-double (DD) when two terms are considered, triple-double (TD) for three terms, quad-double (QD) for four, and so forth. The general case is known under the name of FP expansion.

An important distinction has to be made between double-double arithmetic and conventional *quad* FP arithmetic (binary128). Double-double arithmetic is not standardized and lacks many *nice and clean* properties of binary128, like clearly defined rounding modes, for instance (see Table 1.1 for a summary of differences) and due to this, Kahan [106] qualifies double-double arithmetic as an *attractive nuisance except for the BLAS* and even compares it to an unfenced backyard swimming pool. Indeed, although extensive work has been done in this area (see for instance [B1, Chap. 14] and references therein), many algorithms have been published without a proof, or with error bounds that are not completely explicit (the error is "less than a small integer times u^2").

The attractive advantage is that operations with FP expansions use only hardware implemented and highly optimized FP operations (in binary32 or binary64).

[2] A FMA operator evaluates an expression of the form $xy + t$ with one final rounding only.

Table 1.1 – Main differences between the double-double format (made up with binary64 floating-point numbers) and quad-precision (binary128).

	double-double	quad-precision
Precision	$\geqslant 107$ bits "wobbling"	113 bits
Exponent range	-1022 to 1023 (11 bits)	-16382 to 16383 (15 bits)
Rounding modes	N/A	RN, RU, RD, RZ

One cannot suppress all the drawbacks mentioned by Kahan: clearly, having in hardware a *genuine* floating-point arithmetic with twice the precision would be a better option. And yet, if rigorously proven and reasonably tight error bounds are provided, expert programmers can rely on multiple-word arithmetic for extending the precision of computations when the available floating-point arithmetic does not suffice. This is often done via so-called *compensated algorithms*. For instance the above presented 2Sum, Fast2Sum or 2 ProdFMA represent the simplest instances of such algorithms. For operations with more than two FP numbers, compensated algorithms for summation or inner product (as well as precise error bounds) have recently been subject of intensive research, see for instance [104, 189, 233] or [B1, Chap. 5] for a complete account.

A classical successful example comes from the implementation of elementary functions in fixed precision in so-called mathematical libraries (libms), like glibc, Sun libmcr, Intel©libm or CRlibm [59].

Example of double-double operations in libms. Roughly speaking if the input y of a function, say $\sin(y)$, is given with 15 decimal digits of precision, then the result is expected to also have 15 digits of precision. More specifically, some developers of libms aim for correctly rounded $\sin$ in double-precision. To achieve this, usually, one firstly performs a so-called argument reduction. This allows for the input range to be sufficiently small, so that polynomial approximations are efficient. Such polynomials can be evaluated using only basic arithmetic operations like addition and multiplication. But if these operations are all performed in standard double-precision, it is very difficult to guarantee an intermediary extended accuracy that will allow for a final correctly rounded result in double-precision. Let us explain this in more detail in the following Example 1.2.1, taken from the actual $\sin$ function implementation in CRLibm [59, B1].

Example 1.2.1. For sine function evaluation, the reduced argument x is obtained by subtracting from the floating-point input y an integer multiple of $\pi/256$. As a consequence, $x \in [-\pi/512, \pi/512] \subset [-2^{-7}, 2^{-7}]$. Then, one needs to compute the value of the odd polynomial:

$$p(x) = x + x^3 \cdot (s_3 + x^2 \cdot (s_5 + x^2 \cdot s_7)),$$

which is a polynomial close to the Taylor approximation of the sine function. The coefficients s_1, s_3 and s_5 are represented in binary64 precision arithmetic: $s_3 = -6004799503160661/2^{55}$, $s_5 = 4803839602528529/2^{59}$, $s_7 = -3660068268593165/2^{64}$.

However, since x is an irrational number, the implementation of the range reduction needs to return a number more accurate than a binary64, such that the intermediary output accuracy for $p(x)$ allows for subsequent correct rounding of $\sin(x)$. The CRlibm [59] solution is to consider a *double-double* representation for $x = x_h + x_l$.

As a numerical example, let $y = 0.5$, and the corresponding reduced argument $x = 1/2 - 41\pi/256$. This is approximated in double-double as the unevaluated sum $x_h + x_l$, with $x_h = -7253486725817229/2^{61}$ and $x_l = -508039184604813/2^{112}$.

If one computes directly $p(x_h + x_l)$ with the following Horner scheme and binary64 precision:

$$p_{eval}(x_h + x_l) = (x_h + x_l) + (x_h + x_l)^3 \cdot (s_3 + (x_h + x_l)^2 \cdot (s_5 + (x_h + x_l)^2 \cdot s_7)),$$

one obtains a poor accuracy. Note that with this order of operations, the floating-point addition $x_h + x_l$ returns x_h, so the information held by x_l is lost. The other part of the Horner evaluation also has a much smaller magnitude than y_h, since $|y| \leqslant 2^{-7}$, which gives $\left|y^3\right| \leqslant 2^{-21}$. The following evaluation leads to a much more accurate algorithm, since the leftmost addition is performed with an extended precision, namely the above mentioned Fast2Sum algorithm:

$$\begin{aligned} s &= x_l + (x_h \cdot x_h \cdot x_h \cdot (s_3 + (x_h \cdot x_h \cdot (s_5 + (x_h \cdot x_h \cdot s_7))))), \\ p'_{eval}(x_h + x_l) &= \text{Fast2Sum}(x_h, s). \end{aligned}$$

For our numerical example, one obtains $p'_{eval} = -7253474763108583/2^{61} + 82031/2^{79}$. This allows for 72 bits of accuracy in the evaluation of p compared with 54 for the first evaluation scheme. Note that for both evaluation schemes only standard binary64 operations are used: the second one performs 2 more additions than the first one (by executing the Fast2Sum algorithm) and yet, it allows for an accuracy extension by 33%.

This shows that it is possible to compute very accurate values, even in the presence of roundings at the floating-point level, by using only standard precision floating-point arithmetic operations.

Hence, our goal was to extend these ideas and obtain **new algorithms for floating-point expansions, provide a very efficient implementation and prove tight error bounds**. For several fixed extended precisions (e.g., 2 doubles, 4 doubles), we aimed for performances similar to those of the native format. For that, existing algorithms were improved to be sufficiently simple and regular. This was needed for facilitating their implementation on highly parallel architectures, but also for allowing their formal proof at reasonable cost, since proofs of these algorithms get very tricky. This work resulted in the *CAMPARY: CudA Multiple Precision ARithmetic librarY* project. A brief summary of the obtained results [B1, J3, J6, J7, C5, C6, C9, C11, C14] is given in what follows.

1.3 Double-word Arithmetic

Double-word arithmetic (also known as *double-double*, when the underlying FP format is binary64) consists in representing a real number as the unevaluated sum of two FP numbers.

Definition 1.3.1 (Double-word)**.** A *double-word* number x is the unevaluated sum $x_h + x_\ell$ of two floating-point numbers x_h and x_ℓ such that

$$x_h = \text{RN}(x).$$

We provided a rigorous error analysis of existing algorithms for double-word arithmetic (addition, multiplication, division), introduced a new one for multiplying two double-word numbers, suggested an improvement of the algorithms used in the QD library for dividing a double-word number by an FP number and for dividing two double-word numbers. We have also suggested a new algorithm for dividing two double-word numbers when an FMA instruction is available. Table 1.2 summarizes the obtained results. For the functions for which an error bound was already published, we always obtain a significantly smaller bound, except in one case, for which the previously known bound turned out to be slightly incorrect. Our results make it possible to have more trust in double-word arithmetic. For completeness, we provide: Algorithm 3 and Algorithm 4 which perform addition in a very accurate way; Algorithms 5 and 6 which multiply a double-word with an FP (and respectively two double-words) in 6 (and respectively 9) FP operations when an FMA is available and Algorithm 7 for the division of two double-words with a good accuracy/performance compromise.

Algorithm 3 – DWPlusFP(x_h, x_ℓ, y).

1: $(s_h, s_\ell) \leftarrow$ 2Sum(x_h, y)
2: $v \leftarrow \mathrm{RN}(x_\ell + s_\ell)$
3: $(z_h, z_\ell) \leftarrow$ Fast2Sum(s_h, v)
4: **return** (z_h, z_ℓ)

Algorithm 4 – AccurateDWPlusDW$(x_h, x_\ell, y_h, y_\ell)$.

1: $(s_h, s_\ell) \leftarrow$ 2Sum(x_h, y_h)
2: $(t_h, t_\ell) \leftarrow$ 2Sum(x_ℓ, y_ℓ)
3: $c \leftarrow \mathrm{RN}(s_\ell + t_h)$
4: $(v_h, v_\ell) \leftarrow$ Fast2Sum(s_h, c)
5: $w \leftarrow \mathrm{RN}(t_\ell + v_\ell)$
6: $(z_h, z_\ell) \leftarrow$ Fast2Sum(v_h, w)
7: **return** (z_h, z_ℓ)

Algorithm 5 – DWTimesFP3(x_h, x_ℓ, y).

1: $(c_h, c_{\ell 1}) \leftarrow$ 2ProdFMA(x_h, y)
2: $c_{\ell 3} \leftarrow \mathrm{FMA}(x_\ell, y, c_{\ell 1})$
3: $(z_h, z_\ell) \leftarrow$ Fast2Sum$(c_h, c_{\ell 3})$
4: **return** (z_h, z_ℓ)

Algorithm 6 – DWTimesDW3$(x_h, x_\ell, y_h, y_\ell)$.

1: $(c_h, c_{\ell 1}) \leftarrow$ 2ProdFMA(x_h, y_h)
2: $t_{\ell 0} \leftarrow \mathrm{RN}(x_\ell \cdot y_\ell)$
3: $t_{\ell 1} \leftarrow \mathrm{FMA}(x_h, y_\ell, t_{\ell 0})$
4: $c_{\ell 2} \leftarrow \mathrm{FMA}(x_\ell, y_h, t_{\ell 1})$
5: $c_{\ell 3} \leftarrow \mathrm{RN}(c_{\ell 1} + c_{\ell 2})$
6: $(z_h, z_\ell) \leftarrow$ Fast2Sum$(c_h, c_{\ell 3})$
7: **return** (z_h, z_ℓ)

Algorithm 7 – DWDivDW3$(x_h, x_\ell, y_h, y_\ell)$.

1: $t_h \leftarrow \mathrm{RN}(1/y_h)$
2: $r_h \leftarrow \mathrm{FMA}(-y_h, t_h, 1)$ //exact operation
3: $r_\ell \leftarrow \mathrm{RN}(-y_\ell \cdot t_h)$
4: $(e_h, e_\ell) \leftarrow$ Fast2Sum(r_h, r_ℓ)
5: $(\delta_h, \delta_\ell) \leftarrow$ DWTimesFP3(e_h, e_ℓ, t_h)
6: $(m_h, m_\ell) \leftarrow$ DWPlusFP$(\delta_h, \delta_\ell, t_h)$
7: $(z_h, z_\ell) \leftarrow$ DWTimesDW3$(x_h, x_\ell, m_h, m_\ell)$
8: **return** (z_h, z_ℓ)

Operation	Algorithm	Previously known bound	Bound in [J3]	Largest relative error observed in experiments	FP Ops
DW + FP	Algorithm 3	?	$2u^2 + 5u^3$	$2u^2 - 6u^3$	10
DW + DW	[J3, Algorithm 5]	N/A	N/A	1	11
	Algorithm 4	$2u^2$ (incorrect)	$3u^2 + 13u^3$	$2.25u^2$	20
DW × FP	[J3, Algorithm 7]	$4u^2$	$1.5u^2 + 4u^3$	$1.5u^2$	10
	[J3, Algorithm 8]	?	$3u^2$	$2.517u^2$	7
	Algorithm 5	N/A	$2u^2$	$1.984u^2$	6
DW × DW	[J3, Algorithm 10]	$11u^2$	$7u^2$	$4.9916u^2$	9
	[J3, Algorithm 11]	N/A	$6u^2$	$4.9433u^2$	8
	Algorithm 6	N/A	$5u^2$	$3.936u^2$	9
DW ÷ FP	[J3, Algorithm 13]	$4u^2$	$3.5u^2$	$2.95u^2$	16
	[J3, Algorithm 14]	N/A	$3.5u^2$	$2.95u^2$	10
DW ÷ DW	[J3, Algorithm 16]	?	$15u^2 + 56u^3$	$8.465u^2$	24
	[J3, Algorithm 17]	N/A	$15u^2 + 56u^3$	$8.465u^2$	18
	Algorithm 7	N/A	$9.8u^2$	$5.922u^2$	31

Table 1.2 – Summary of the results presented in [J3], where DW stands for double-word; N/A means that the algorithm existed before, but no bound was proven, ? means that the algorithm was given for the first time in our work.

1.4 FP Expansions

Next, we focused on FP expansions with more than two FP terms i.e., numbers are represented as the unevaluated sum of more than two standard precision FP.

Definition 1.4.1 (FP Expansion)**.** A *floating-point expansion* x with n terms is the unevaluated sum of n floating-point numbers $x_0, \ldots, x_{n-1}$, in which all nonzero terms are ordered by magnitude (i.e., if y is the sequence obtained by removing all zeros in the sequence x, and if y contains m terms, then $|y_i| \geqslant |y_{i+1}|$ for all $0 \leqslant i < m-1$). Each x_i is called a *component* (or a term) of x.

In the case of two terms, the additional condition:

$$x_{i+1} \leqslant \frac{1}{2}\,\mathrm{ulp}(x_i), \tag{1.1}$$

appears naturally when expressing the rounded-to-nearest sum of two numbers and their rounding error. This requirement was generalized for more terms by Hida, Li, and Bailey [95]. While not imposing unicity, such constraint usually called *nonoverlapping representation*, ensures *compacity*: it takes fewer terms for achieving the same accuracy. Many similar notions on nonoverlapping were defined in literature firstly by Priest [178], then Shewchuk [199], followed by Bailey QD library [95].

An expansion may contain interleaving zeros, but the definitions that follow apply only to the nonzero terms of the expansion (i.e., the sequence y in Definition 1.4.1).

According to Shewchuk [199], *nonzero-overlapping* expansions, are defined as follows:

Definition 1.4.2 ($\mathcal{S}$-nonoverlapping Expansion)**.** A floating-point expansion $x_0 + x_1 + \cdots + x_{n-1}$ is *$\mathcal{S}$-nonoverlapping* (that is, nonoverlapping according to Shewchuk's definition) if for all $1 \leqslant i \leqslant n-1$, we have $e_{x_{i-1}} - e_{x_i} \geqslant p - z_{x_{i-1}}$, where $e_{x_{i-1}}$ and e_{x_i} are the exponents of x_{i-1} and x_i, respectively, and $z_{x_{i-1}}$ is the number of trailing zeros of x_{i-1}.

Note that zero is $\mathcal{S}$-nonoverlapping with any nonzero floating-point number.

For example, in a binary floating-point system of precision $p = 4$, the numbers $1.100_2 \times 2^3$ and $1.010_2 \times 2^1$ are $\mathcal{S}$-nonoverlapping , whereas they are not nonoverlapping according to (1.1).

In extreme cases, in radix 2, an $\mathcal{S}$-nonoverlapping expansion with 53 components may not contain more information than one binary64 number (it suffices to put each bit of a floating-point number in a separate component). And yet, $\mathcal{S}$-nonoverlapping expansions are of interest due to the simplicity of the related arithmetic algorithms. Let us give an important example. First, following Priest [178], we define expansions whose terms "overlap by at most $0 \leqslant d \leqslant p-2$ bits", where p is the underlying precision.

Definition 1.4.3. Consider n precision-p floating-point numbers: $x_0, x_1, \ldots, x_{n-1}$. They overlap by at most d binary digits ($0 \leqslant d < p$) if and only if for all $i, 0 \leqslant i \leqslant n-2$, there exist integers k_i, δ_i such that

$$2^{k_i} \leqslant |x_i| < 2^{k_i+1}, \tag{1.2}$$

$$2^{k_i-\delta_i} \leqslant |x_{i+1}| \leqslant 2^{k_i-\delta_i+1}, \tag{1.3}$$

$$\delta_i \geqslant p - d, \tag{1.4}$$

$$\delta_i + \delta_{i+1} \geqslant p - z_{i-1}, \tag{1.5}$$

where z_{i-1} is the number of trailing zeros at the end of x_{i-1} and for $i = 0$, $z_{-1} = 0$.

Loosely speaking, this definition states that when written in positional notation, the binary digits of any two successive nonzero terms coincide in at most d positions, and no three terms mutually coincide in any digit position.

We proved in [J6] that, under mild assumptions, VecSum(x) (where VecSum is Algorithm 8) which is simply a chain of 2Sum, applied to an expansion x with n terms, makes it $\mathcal{S}$-nonoverlapping as soon as its terms *overlap by at most* $0 \leqslant d \leqslant p-2$ *bits*. We also prove that for a better performance, when $d \leqslant p-2$, the 2Sum calls in the VecSum algorithm can be replaced by calls to Fast2Sum.

The fact that VecSum transforms an expansion whose terms overlap by at most $d \leqslant p-2$ bits into an $\mathcal{S}$-nonoverlapping expansion is important, because an $\mathcal{S}$-nonoverlapping expansion can easily be transformed into another expansion with a "stronger" nonoverlapping property, that we now define.

Algorithm 8 – VecSum$(x_0, \ldots, x_{n-1})$.

1: $s_{n-1} \leftarrow x_{n-1}$
2: **for** $i \leftarrow n-2$ **to** 0 **do**
3: $\quad (s_i, e_{i+1}) \leftarrow \text{2Sum}(x_i, s_{i+1})$
4: **end for**
5: $e_0 \leftarrow s_0$
6: **return** $e_0, \ldots, e_{n-1}$

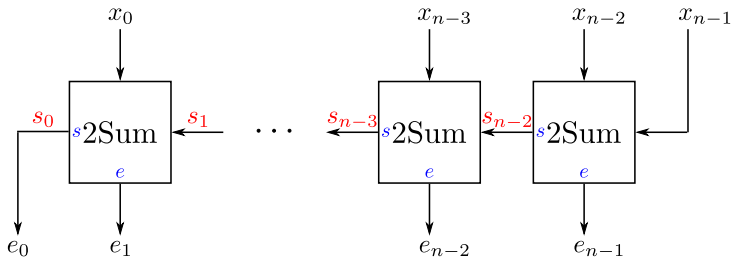


Figure 1.1 – Graphical representation of Algorithm 8. In the 2Sum calls the sum s is outputted to the left and the error e downwards.

Definition 1.4.4 (ulp-nonoverlapping Expansion [J6])**.** A floating-point expansion $x_0 + x_1 + \cdots + x_{n-1}$ is ulp-*nonoverlapping* if for all $1 \leqslant i \leqslant n-1$, $|x_i| \leqslant \text{ulp}(x_{i-1})$.

Depending on the nonoverlapping type of an expansion, when using standard floating-point formats as underlying arithmetic, the exponent range forces a constraint on the number of terms. The largest expansion can be obtained when the largest term is close to overflow and the smallest is close to underflow. This gives a maximum ulp-nonoverlapping expansion size of 40 for binary64 and 12 for binary32.

Concerning ulp-nonoverlapping expansions, we proposed several new arithmetic algorithms designed either for efficiency (*quick-and-dirty* algorithms) or reliability that is, providing tight error bounds on the results. Implemented in CAMPARY library, they are collected and proven in detail in Popescu's PhD dissertation [176] together with [C14, J6].

We focus here on a renormalization algorithm, which has a key role in restoring the nonoverlapping property after different manipulations with expansions. This algorithm was also formally proved in a joint work with S. Boldo, J.-M. Muller and V. Popescu [C6].

1.4.1 Renormalization of floating-point expansions

We have seen that an expansion that "does not overlap too much" (its terms overlap by at most $d \leqslant p-2$ bits) can easily be transformed into an $\mathcal{S}$-nonoverlapping expansion. Let us now give an algorithm, VecSumErrBranch, that transforms an $\mathcal{S}$-nonoverlapping expansion into an ulp-nonoverlapping expansion.

The VecSumErrBranch algorithm

Algorithm 9 (also represented in Figure 1.2) is a variation of the VecSum Algorithm 8, which starts from the most significant term and instead of propagating the partial sums, propagates the errors. If however, the error after a 2Sum block is zero, the sum is propagated instead. It is formally proved by Boldo et al. [C6] that this algorithm transforms an $\mathcal{S}$-nonoverlapping expansion into an ulp-nonoverlapping one and that, in all practical cases (in particular, the IEEE formats), 2Sum calls can be safely replaced by Fast2Sum.

Algorithm 9 – VecSumErrBranch$(e_0, \ldots, e_{n-1}, m)$, where m is the number of terms of the result

$j \leftarrow 0$
$\varepsilon_0 = e_0$
for $i \leftarrow 0$ **to** $n-2$ **do**
  $(r_j, \varepsilon_{i+1}) \leftarrow$ 2Sum(ε_i, e_{i+1})
  if $\varepsilon_{i+1} \neq 0$ **then**
    if $j \geqslant m-1$ **then**
      return $r_0, r_1, \ldots, r_{m-1}$ //enough output terms
    end if
    $j \leftarrow j+1$
  else
    $\varepsilon_{i+1} \leftarrow r_j$
  end if
end for
if $\varepsilon_{n-1} \neq 0$ **and** $j < m$ **then**
  $r_j \leftarrow \varepsilon_{n-1}$
end if
return $r_0, r_1, \ldots, r_{m-1}$

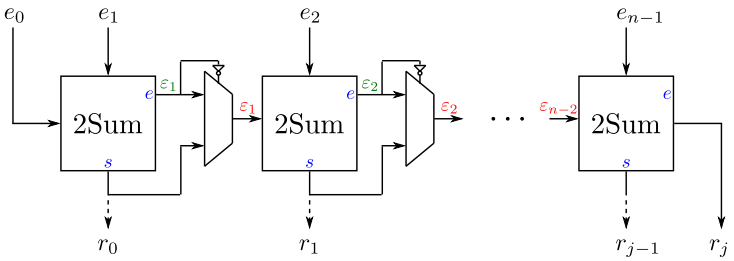


Figure 1.2 – Graphical representation of Algorithm 9. In the 2Sum calls the sum s is outputted downwards and the error e to the right.

The renormalization algorithm

By successively using VecSum and VecSumErrBranch, we can convert an array of numbers overlapping by at most $p-2$ bits into an ulp-nonoverlapping expansion. This gives Algorithm 10 below.

Algorithm 10 – Renormalize$(x_0, \ldots, x_{n-1}, m)$.

Input: a sequence $x_0, x_1, \ldots, x_{n-1}$ of floating-point numbers overlapping by at most $p-2$ bits.
Output: an ulp-nonoverlapping expansion $r_0, r_1, \ldots, r_{m-1}$.
$(e_0, e_1, \ldots, e_{n-1}) \leftarrow$ VecSum$(x_0, x_1, \ldots, x_{n-1})$
$(r_0, r_1, \ldots, r_{m-1}) \leftarrow$ VecSumErrBranch$(e_0, e_1, \ldots, e_{n-1}, m)$
return $r_0, r_1, \ldots, r_{m-1}$

Renormalization of arbitrary numbers

We have seen how to "renormalize" a sequence that "does not overlap too much". If we are given an arbitrary sequence of floating-point numbers as input, renormalization is still possible, but at a much higher cost. For Algorithm 11, given below, we have proven the following theorem in [J6].

Theorem 1.4.5. Let $x_0, x_1, \ldots, x_{n-1}$ be a sequence of n floating-point numbers that may contain interleaving 0s, and let m be an integer such that $1 \leqslant m \leqslant n-1$. Provided that no underflow/overflow occurs during the calculations, Algorithm 11 returns the first m terms of an ulp-nonoverlapping floating-point expansion $r = r_0 + \cdots + r_{n-1}$ such that $x_0 + \cdots + x_{n-1} = r$.

Algorithm 11 – Renormalize_arbitrary$(x_0, \ldots, x_{n-1}, m)$.

Input: an arbitrary sequence $x_0, x_1, \ldots, x_{n-1}$ of floating-point numbers.
Output: an ulp-nonoverlapping expansion $r_0, r_1, \ldots, r_{m-1}$.
$e_0^{(0)} \leftarrow x_0$
for $i \leftarrow 1$ **to** $n-1$ **do**
  $(e_0^{(i)}, e_1^{(i)}, \ldots, e_i^{(i)}) \leftarrow$ VecSum$(e_0^{(i-1)}, e_1^{(i-1)}, \ldots, e_{i-1}^{(i-1)}, x_i)$
end for
$(r_0, r_1, \ldots, r_{m-1}) \leftarrow$ VecSumErrBranch$(e_0^{(n-1)}, e_1^{(n-1)}, \ldots, e_{n-1}^{(n-1)}, m)$
return $r_0, r_1, \ldots, r_{m-1}$

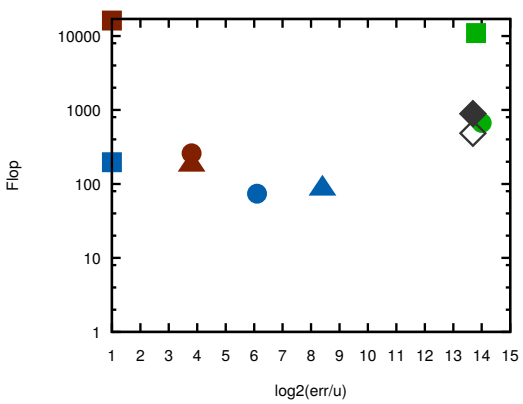
1.4.2 Arithmetic of FP expansions

Based on the above renormalization algorithm one can design other arithmetic operations. For instance, we proved in Popescu's thesis [176] that by merging two ulp-nonoverlapping expansions in decreasing order of magnitude, and renormalizing the resulting array using Algorithm 10, the output expansion s, with r terms, is ulp-nonoverlapping and satisfies the error bound:

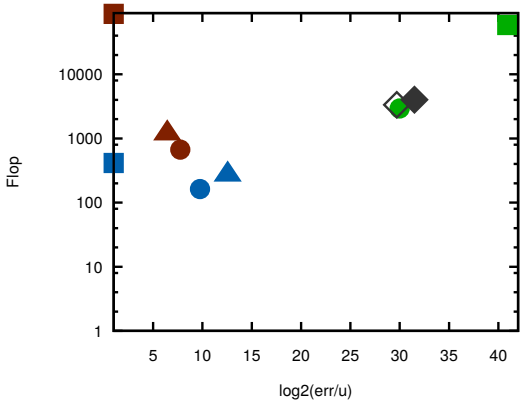
$$|x + y - s| < \frac{9}{2} \cdot 2^{-(p-1)r} \cdot (|x| + |y|), \tag{1.6}$$

as soon as the underlying precision p is at least 4, which always holds in practice.

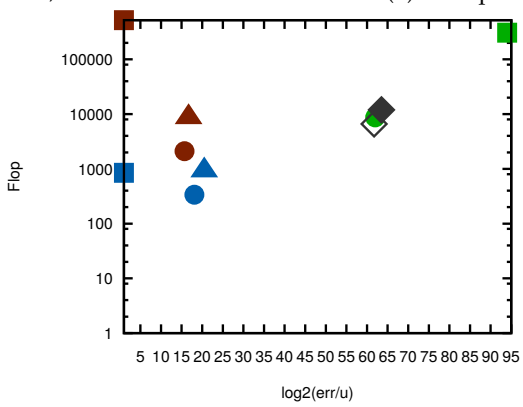
Similarly, we proposed arithmetic algorithms for multiplication, division, reciprocal and square root. We refer the reader to Chap. 3 of Popescu's thesis [176] for a detailed description all the proposed variants and proofs of error bounds. We provide in Figure 1.3 a summary of error bounds vs. operation count for some of the proposed algorithms with FP expansions of increasing sizes. One can observe that we drastically improved the efficiency compared to Priest [178] algorithms, while proving good accuracy bounds.



(a) FP expansions with 4 terms; $u = 2^{-212}$



(b) FP expansions with 8 terms; $u = 2^{-424}$



(c) FP expansions with 16 terms; $u = 2^{-848}$

Figure 1.3 – Error bounds vs. number of FP operations for several arithmetic algorithms: square indicates Priest algorithms [178] (Alg. 26, 30, 34 of [176, Chap. 3]); circle indicates our most accurate algorithms, namely Alg. 27, 31 and 36 of [176, Chap. 3]; triangle indicates other existing versions of the studied algorithms (Alg. 27, 33 of [176, Chap. 3]); blue color stands for addition, red for multiplication, green for division; full gray diamond for square-root (Alg. 37 [176, Chap. 3]) and empty gray diamond for reciprocal (Alg. 35 of [176, Chap. 3]).

1.5 Applications of CAMPARY

We present two numerical applications of CAMPARY, which make use of both higher-precision and high-performance computations. Firstly, an open question raising in a classical chaotic dynamical system, the Hénon map, is to search for periodic orbits, for values of the parameters close to the classical ones, for which the map is believed to be chaotic. This is hence a very numerically sensitive problem. Our approach [J7], based on extensive long-term numerical iterations of the map, also needs high-performance computing. The second application concerns improving the accuracy of semidefinite programming (SDP) solvers for ill-conditionned instances [C5].

Solutions for these problems are obtained much faster when numerical algorithms are implemented on highly parallel architectures, like for instance GPUs. To this end, CAMPARY is interesting because very few multiple-precision libraries can be ported to GPUs. We provided a CUDA implementation, which is an extension of the C language [162] for NVidia GPUs. Two other similar libraries are GQD [138], which supports double-double (DD) and quad-double (QD) computations, or CUMP [158]. However, these turn out to be suboptimal for our purpose: CUMP is based on the *multiple-digit* format, for which basic operations are slower than those on *multiple-terms*, while GQD is *multiple terms* but limited to 4 doubles.

1.5.1 Finding sinks for Hénon map

Hénon map [99] can be considered as one of the *classic* discrete dynamical systems, for which some long-standing questions remain open. It is a two-parameter, invertible map $h(x, y) = (1 + y - ax^2, bx)$., which, depending on a and b, can be chaotic, regular (the attractor of the map is a stable periodic orbit), or a combination of these. It is conjectured that for the classical parameters $a = 1.4$ and $b = 0.3$, the Hénon map is chaotic and supports a strange attractor [99]. This property has been observed numerically, but *the question whether the Hénon attractor is indeed chaotic (trajectories belonging to the attractor are aperiodic and sensitive to initial conditions) remains open.*

It is known [14] that there is a set of parameters (near $b = 0$) with positive Lebesgue measure for which the Hénon map has a strange (chaotic) attractor. The parameter space is believed to be densely filled with open regions, where the attractor consists of one or more stable periodic orbits (sinks). In light of this, it is probably impossible to verify that, given a specific point (a, b) in parameter space, the dynamics of the map generates a strange attractor.

On the other hand, it was more recently proven using validated numerics [78] that for several parameter values close to the classical ones, what appears to be a strange attractor (Fig. 1.4(a)) is actually a stable periodic orbit (Fig. 1.4(b)). Specifically, in Fig. 1.4, 10000 iterations of the Hénon map $h(x, y)$, with fixed parameters $a = 1.399999486944$ and $b = 0.3$ are plotted. The iterates appearing in Fig. 1.4(a) start in a point (x'_0, y'_0) and those for Fig. 1.4(b) in (x''_0, y''_0), which are chosen in the following way: $5 \cdot 10^9$ iterations are performed and skipped (not plotted) before obtaining (x'_0, y'_0); and respectively, for (b) $6 \cdot 10^9$ iterations are skipped before obtaining (x''_0, y''_0). Clearly, Fig. 1.4(a) looks like the Hénon strange attractor, while Fig. 1.4(b) is just a periodic orbit. This means that what we observe in computer simulations is actually a transient behavior to the periodic steady-state that we are actually interested in.

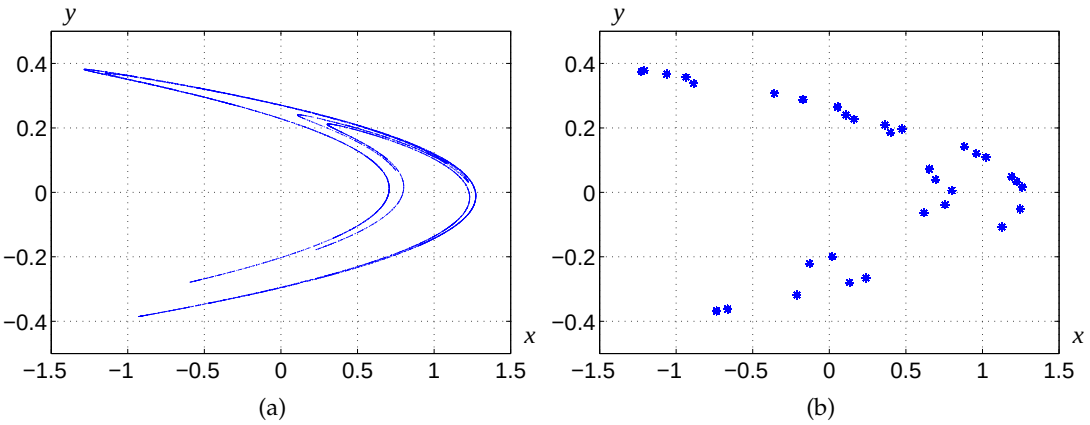


Figure 1.4 – Hénon map $h(x, y) = (1 + y - ax^2, bx)$ with $a = 1.399999486944$, $b = 0.3$; 10000 iterates are plotted after skipping (a) $5 \cdot 10^9$ and (b) $6 \cdot 10^9$ iterations.

Proving the existence of such a stable periodic orbit involves a finite (yet challenging) amount of computations, and all necessary conditions are robust (there exists an open set in the parameter space in which all conditions remain true). So, if such a sink exists, we should theoretically be able to find it using high performance computing. In order to find sinks for parameters close to the classical ones, we need to compute very long orbits for a large amount of initial points and parameters, as follows:

(i) for each considered point (a, b) in parameter space, we perform a large amount of iterations of the Hénon map h for many different initial points. The hope is that at least one of these trajectories will, after some initial transient behaviour, be attracted to what appears to be a periodic orbit. Specifically, given a fixed (a, b) together with a single initial point (x_0, y_0), the subsequent computations are governed by two integers N_t and p_{max}. First, we perform N_t iterations of the map h: $h(x_0, y_0)$, $h(h(x_0, y_0))$,..., $h^{N_t}(x_0, y_0)$. These are all discarded, except the final iterate $h^{N_t}(x_0, y_0)$, which we continue to follow for another p_{max} iterates. At this stage, we examine the piece of orbit $h^{N_t+1}(x_0, y_0)$,..., $h^{N_t+p_{max}}(x_0, y_0)$ for any close return. In other words, we attempt to find an integer $1 < k < p_{max}$ such that $\max_{i=1}^{k} \|h^{N_t+i}(x_0, y_0) - h^{N_t+i+k}(x_0, y_0)\|$ is small. If this succeeds, we may have found a period-k sink, which we verify *a posteriori* in a second step. The number N_t of transient iterations which are discarded is usually chosen by trial-and-error since it depends on hidden intrinsic properties of the dynamics of the Hénon map. In practice, the following values were employed: $N_t \sim 10^9$, $p_{max} = 5000$. Moreover, for each parameter choice, $N_i \sim 10^3$ different initial points are used. Finally, the entire procedure is repeated for $N_p \sim 10^6$ parameters near $(1.4, 0.3)$.

(ii) Rigorous numerics (particularly an interval Newton operator [152, 159]) are used to validate/falsify the existence of any sink found in the previous step. This step is not detailed further here, since similar techniques will be discussed in Chapter 2.

With this process, and using only binary64 computations, we found 57 parameters which present stable periodic orbits in 2.94 hours on 2 Nvidia GeForce Tesla C2075 GPU with 448 cores, 1.15GHz. A 21.5x speedup was obtained by our CUDA C implementation vs. a C implementation with OpenMP on Intel(R) Core(TM) i7 CPU 3820, 3.6GHz, 4 cores, 8 threads. This computation confirmed the results obtained in [78]. When increasing the precision, two orbits given in [79] and Table 1.3, were obtained using our GPU implementation. Compared with GQD, our implementation was 1.6 (and respectively 2.8) times faster for double-double (and respectively quad-double) computations.

P	d	r
41	$4.73e{-}10$	$3.31e{-}17$
47	$1.47e{-}10$	$1.17e{-}20$

Table 1.3 – Hénon map sinks found using CAMPARY on GPU [79]. P is the period, d is the distance to the point $(1.4, 0.3)$, r is the minimum immediate basin radius.

The closest to the classical parameters sink (of period 115) known to-date [79] is given by parameters:

$$a = 1.39999999999999999999968839903277301984563091568983, \text{ and}$$

$$b = 0.29999999999999999999944845519288458244332946957783,$$

and for which the periodic window is $d = 6.335e{-}22$ [79]. Hence, the periodic windows are very narrow and the transient time to corresponding sinks can be extremely long, which implies that "it is practically impossible to observe such sinks in simulations" [79]. This result draws a line on this search for sinks, for this particular system. On the one hand it proved that our arithmetic library is efficient, but on the theoretical side it also implied that there is no point in trying to get closer in parameter space. We continue to work with W. Tucker on different dynamical systems, as further explained in Chapter 4.

1.5.2 Performance assessment of CAMPARY with SDPA

Semidefinite programming (SDP) is an important branch of convex optimization and can be seen as a natural generalization of linear programming to the cone of symmetric matrices with non-negative eigenvalues, i.e. positive semidefinite matrices. However, certain SDP instances are ill-posed and need more accuracy than that provided by the standard double-precision. Moreover, these problems are large-scale and could benefit from parallelization on specialized architectures such as GPUs. Examples of such problems appear in the high-accuracy computation of *kissing numbers*, i.e. the maximal number of non-overlapping unit spheres that simultaneously can touch a central unit sphere [149]; bounds from binary codes; control theory and structural design optimization (e.g., the wing of Airbus A380) [61]; quantum information and physics [200].

Numerical inaccuracies when solving SDP with finite precision appear for two main reasons. On the one hand, strong duality does not always hold. In practice, the method of choice for SDP solving is based on interior-point algorithm, like the primal-dual path-following interior-point method (PDIPM) [151]. This algorithm is considered in literature as theoretically mature and is widely accepted and implemented in most state-of-the-art SDP solvers like SDPA [231], CSDP [22], SeDuMi [205], SDPT3 [215]. However, problems which do not have an interior feasible point induce numerical instability and may result in inaccurate calculations or non-convergence. The SPECTRA package [94] proposes to solve such problems with exact rational arithmetic, but the instances treated are small and this package does not aim to be a concurrent of general numerical solvers.

On the other hand, even for problems which have interior feasible solutions, numerical inaccuracies may appear when solving with finite precision due to large condition numbers (higher than 10^{16}, for example) which appear when solving linear equations. This happens, as explained in [156], when approaching optimal solutions, since the semidefinite matrices appearing in the primal and dual instances become singular in practice (due to the complementary slackness and optimality condition $X^*Y^* = 0$). In this second case, having an efficient underlying multiple-precision arithmetic is crucial to detect (at least numerically) when the convergence issue arises simply from numerical errors due to lack of precision.

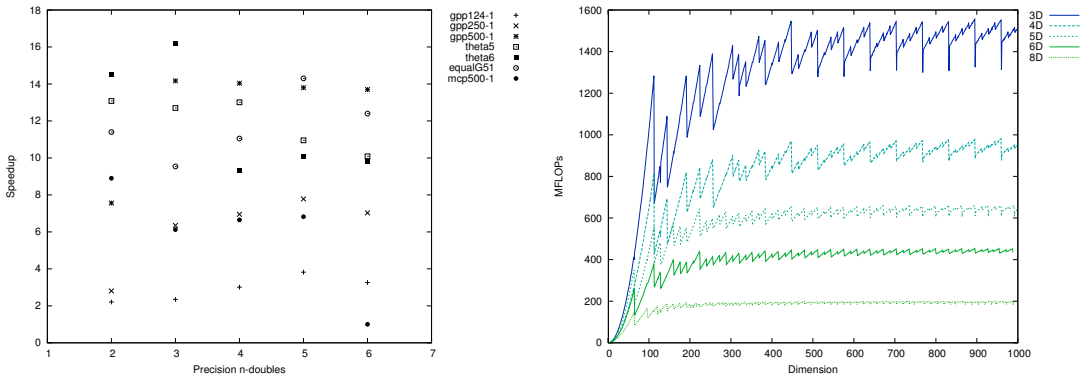
This resulted in a recent increased interest in providing both higher-precision (also called *multiple precision*) and high-performance SDP libraries. We implemented and evaluated the performance of our previously presented arithmetic algorithms of CAMPARY in this context.

To achieve this, CAMPARY was plugged-in for both SDPA CPU and GPU-tuned implementations. Several multiple precision arithmetic libraries like GMP and QD were already ported inside SDPA [156]. We compared and contrast both the numerical accuracy and performance of SDPA-GMP, -QD and -DD, which employ other multiple-precision arithmetic libraries against SDPA-CAMPARY. We showed

that CAMPARY is a very good trade-off for accuracy and speed when solving ill-conditioned SDP problems. For that, an important contribution was a multiple precision GPU compatible general matrix multiplication routine *RGEMM* that can be used in SDPA. This routine runs at up to 83% of the theoretical GPU peak-performance and allows for an average speedup of one order of magnitude for SDP instances run in multiple precision with SDPA-CAMPARY and GPU support compared to SDPA-CAMPARY on CPU only, as showed in Figure 1.5a.[3]

Concerning our GPU implementation, performance results for n-double RGEMM are shown in Figure 1.5b. It is important to note that the RGEMM implementation is quite efficient with: 83% of theoretical peak performance for DD, respectively 43% for TD, 50% for 4D, 57% for 5D, 61% for 6D, 57% for 8D, while other implementations [207] of DGEMM (matrix multiplication in double precision) attain $58 - 70\%$ of the theoretical peak performance. Our DD implementation is slower by $\sim 10\%$ than the implementation in [157], which can be explained by the generality of our code.

We conclude by comparing in Table 1.4 the performance obtained on the CPU for a classical problem in coding theory: finding the largest set of binary words with n letters, such that the Hamming distance between two words is at least d. This is reformulated as a maximum stable set problem, which is solved with SDP by Schrijver [195] and Laurent [128]. Instances for such problems are taken from [61]. The comparison is done between SDPA-DD, SDPA-GMP (run with 106 bits of precision) and SDPA-CAMPARY-DD. The accuracy of the obtained results is similar, while our library has better timings. Some instances do not converge when DD precision is used. We also include the results obtained with the SDPA-CAMPARY with triple-double (TD) precision, which has a better performance comparing to SDPA-GMP with 106 bits of precision.



(a) (b)

Figure 1.5 – (a) Speedup of SDPA-CAMPARY for n-double with GPU vs CPU, on several problems from SDPLIB. Maximum speedup was 16.2; (b) Performance of RGEMM with CAMPARY for n-double on GPU. Maximum performance was 1.6GFlops for TD, 976MFlops for QD, 660MFlops for 5D, 453MFlops for 6D, 200MFlops for 8D.

1.6 Conclusion

Nowadays, very efficient arithmetic operations with double-precision floating-point numbers compliant with the IEEE-754 standard are available on most recent computers. However, when more than double-precision/binary64 (53 bits) is required, especially in the HPC context, few multiple-precision arithmetic libraries exist, and the trade-off between performance versus reliability is still a challenge. In this sense, we summarized some of our recent results obtained during the master/PhD thesis of my student V.

[3] Benchmarks were performed on well-known ill-conditioned examples from SDPLIB [23] and [61]. GPU tests were performed on a GPU NVIDIA(R) Tesla(TM) C2075, with 448 cores, 1.15 GHz, 32KB of register, 64KB shared memory/L1 cache set by default to 48KB for shared memory and 16KB for L1 cache. CPU tests use an Intel(R) Xeon(R) CPU E3-1270 v3 @ 3.50GHz processor, with Haswell micro-architecture which supports hardware implemented FMA instructions.

Problem	SDPA-DD	SDPA-CAMPARY-DD	SDPA-CAMPARY-TD	SDPA-GMP
Laurent_A(19,6)	optimal: $-2.4414745686616550e-03$			
iteration	92	94	71	73
time (s)	4.3	3.1	18.65	29.16
Laurent_A(26,10)	optimal: $-1.3215201241629400e-05$			
iteration	80	80	123	125
time (s)	12.8	8.68	109.54	173.42
Laurent_A(28,8)	optimal: $-1.1977477306795422e-04$			
iteration	93	100	76	113
time (s)	47.8	36.85	219.46	541.19
Laurent_A(48,15)	optimal: $-2.229e-09$			
iteration	134	134	165	145
time (s)	2204.61	1569.48	14691.92	21695.08
Laurent_A(50,15)	optimal: $-1.9712e-09$			
iteration	142	142	191	154
time (s)	3463.2	2421.86	25773.96	35173.79
Laurent_A(50,23)	optimal: $-2.5985e-13$			
iteration	124	124	155	140
time (s)	342.73	221.32	2333.74	3426.17
Schrijver_A(19,6)	optimal: $-1.279036270018091 0e+03$			
iteration	40	40	66	95
time (s)	1.59	1.14	14.65	32.21
Schrijver_A(26,10)	optimal: $-8.858571428571388 0e+02$			
iteration	54	54	127	108
time (s)	7.75	5.2	100.73	134.48
Schrijver_A(28,8)	optimal: $-3.215079582579291 3e+04$			
iteration	45	45	69	97
time (s)	21.05	15.06	182.25	422.78
Schrijver_A(37,15)	optimal: $-1.400699999999988 6e+03$			
iteration	58	58	132	116
time (s)	54.86	36.35	683.07	988.21
Schrijver_A(40,15)*	optimal: $-1.9e+04$			
iteration	23	23	23	23
time (s)	53.99	35.99	285.3	471.870
Schrijver_A(48,15)*	optimal: $-2.56e+06$			
iteration	27	27	27	27
time (s)	432.13	307.88	2260.24	3862.29
Schrijver_A(50,15)**	optimal: $-7.6e+06$			
iteration	29	29	29	29
time (s)	694.07	471.57	3695.95	677.830
Schrijver_A(50,23)**	optimal: $-5.2e+03$			
iteration	29	29	29	29
time (s)	76.55	47.84	413.31	6370.97

Table 1.4 – The optimal value, iterations and time for solving some ill-posed problems for binary codes by SDPA-DD, -CAMPARY-DD, -CAMPARY-TD, and -GMP-DD. $*$ problems that converge to more than two digits only when using quad-double precision. $**$ problems that converge to more than two digits with precision higher that quad-double. The digits written with blue were obtained only when triple-double precision was employed.

Popescu. We proposed to represent and compute with multiple-precision numbers via unevaluated sums of standard machine precision floating-point numbers, so-called **floating-point expansions**. This approach allows to directly benefit from the available and efficient hardware implementation of the IEEE-754 standard. We improved or designed several new algorithms for performing basic arithmetic operations using this extended format. For all the algorithms, we gave rigorous correctness and error bound proofs as well as an implementation in our multiple-precision arithmetic library, CAMPARY. Our work focused not only on the arithmetic details and technicalities, but also on the applications of CAMPARY. Related future works are discussed in Chapter 4.

Chapter 2

Symbolic-Numeric Computations

In the last 15 years, the increasing demands in both speed and reliability in scientific and engineering computing, contributed to the merging of symbolic and numeric computations, which were coexisting, but traditionnally separated research branches in computational mathematics [221, 118]. In this framework, I worked on techniques mixing structural properties (coming from the symbolic field) of Linear Ordinary Differential Equations (LODEs) with polynomial coefficients, with efficient numerical routines coming from optimization or approximation theory. Moreover, an important aspect pertaining to this last area, is not only to compute *approximations*, but also *enclosures* of the approximation errors, which give an effective quality measurement of the computation. All in all, the goal is to provide very efficient algorithms (with proven theoretical complexity) which provide accurate and reliable approximate solutions together with effective (approximation and rounding) error bounds. In this context, three contributions are presented in what follows:

1. We proposed a new accurate, reliable and efficient method to evaluate 2D Gaussians on disks [J5]. This result is employed for the computation of orbital collision probability between two spherical space objects involved in a short-term encounter as detailed in Chapter 3.

2. Motivated by this preliminary work and its links with the algebraic structure of moments of Gaussian measures supported on semi-algebraic sets, we extended a study of Lasserre and Putinar [126] to inverse problems involving measures with holonomic densities and support with real algebraic boundary. In the framework of holonomic distributions (i.e. they satisfy a holonomic system of linear partial or ordinary differential equations with polynomial coefficients), our results exploit the linear recurrence structure of corresponding moments [C2].

3. The two previous contributions make use of algebraic properties of power series solutions of D-finite/holonomic linear differential (systems) of equations. A related aspect, which is important in approximation problems, is to consider other orthogonal series expansions. We focused on the efficient computation of truncated Chebyshev series (together with rigorous approximation error bounds) for D-finite functions in two recent articles [J4, J2].

These contributions are in collaboration with researchers working in Computer Algebra: B. Salvy (Inria researcher, LIP, Lyon) and his former students A. Benoit (Mathematics teacher, A. Dumas High School, St-Cloud) and M. Mezzarobba (CNRS Researcher, Paris), as well as in Optimization and Control at LAAS Laboratory: D. Arzelier and J.-B. Lasserre (CNRS researchers, LAAS, Toulouse) and A. Rondepierre (Maître de conférences, INSA, Toulouse). My PhD student F. Bréhard, made very important contributions to the works listed in items 2. and 3. He is to defend his PhD Thesis in July 2019, and is co-supervised with N. Brisebarre and D. Pous (CNRS Researchers in Lyon). The subject discussed in item 3. is also joint research with N. Brisebarre.

It is important to remark that the theoretical tools developed for these works were inspired by practical applications coming either from the conception of mathematical libraries (which corresponds to my PhD background) or from the aerospace domain (which corresponds to some of my current research activities as a member of MAC team) and which will be detailed in Chapter 3.

Computation and complexity model Our numerical algorithms rely on *floating-point* arithmetics, either in standard double precision, or in arbitrary precision when needed. In the later case, `GNU-MPFR` library [74] is used. For *validated* computations, we make use of *interval arithmetics* via the `MPFI` library [185].Complexity results are given in the uniform complexity model: all basic arithmetic operations (addition, subtraction, multiplication, division and square root), either in rational arithmetic (for symbolic algorithms), floating-point or interval arithmetic, induce a unit cost of time. In particular, we do not investigate the incidence of the precision parameter on the global time complexity.

To provide some common ground for our results, a few introductory notions concerning (univariate) D-finite functions and (multivariate) holonomic functions are given in what follows. Great existing surveys on these topics include that of B. Salvy [191], F. Chyzak [53] or C. Koutschan [114].

2.1 Differential finiteness

Rougly speaking, for the definitions in this section, the *numbers involved* are rational. Formally, one can consider a field $\mathbb{K}$, which is a real finite computable extension of $\mathbb{Q}$.

2.1.1 Univariate case

D-finite (differentially finite) functions are functions satisfying a linear differential equation with polynomial coefficients. A simple example is the exponential function $\exp$ which satisfies $\exp' - \exp = 0$. To specify a D-finite function y, a linear homogeneous differential equation of order r with polynomial coefficients

$$L \cdot y = a_r y^{(r)} + a_{r-1} y^{(r-1)} + \cdots + a_0 y = 0, \quad a_i \in \mathbb{K}[x], \tag{2.1}$$

together with r initial values

$$y^{(i)}(0) = \ell_i, \qquad 0 \leqslant i \leqslant r-1, \tag{2.2}$$

is considered such that y is its unique solution.

This specification can be seen as a data structure, because many mathematical properties of y can be inferred directly from this equation. Such a data structure can represent the vast majority of functions commonly used in mathematics and physics, which turn out to be *D-finite functions*, e.g. $\exp$, $\sin$, $\cos$, their hyperbolic counterpart and their functional inverses (arc trigonometric and arc hyperbolic functions), Bessel, Airy functions, etc. Instead of adopting a closed-formula representation for each such function, one can consider only the LODE satisfied by the function and suitable initial conditions, which is also a *finite* data-structure. This allowed for the development of a uniform theoretic and algorithmic treatment of these functions, an idea that has led to many applications in recent years in the context of Symbolic Computation [232, 191, 192, 24, 27, 16, 28].

In particular, such functions can be expanded in power series whose coefficients are P-recursive, i.e. they satisfy a linear recurrence with polynomial terms in the index variable [202]. For example, let $\exp(z) = \sum_{n=0}^{+\infty} c_n z^n$, then the recurrence satisfied by the coefficients c_n is $(n+1) \cdot c_{n+1} = c_n,\ c_0 = 1$.

Indeed, this class of functions marks an ideal spot at the junction between symbolic and numeric computation.

On the approximation side, D-finite functions possess strong analytic properties that can be exploited in the production of approximations and corresponding error bounds. Power series approximations with tight bounds for this class of functions can then be obtained: this is based on efficient algorithms for computing the nth coefficient of the power series [52],[25, Chap.15] together with algorithms which produce majorant series, whose speed of convergence is controlled [218, 148]. This also entails efficient numerical evaluations of D-finite functions outside of the disk of convergence of the initial power series, via so-called analytic continuation [218, 148, 25]. M. Mezzarobba `ore_algebra_analytic` package [146, 147] provides a SageMath [203] implementation for these algorithms.

Important properties of D-finite functions include closure under addition, product, Hadamard product, or Laplace/Borel transform. Moreover for algebraic y (there exists a non-zero polynomial P s.t. $P(x, y) = 0$), and D-finite f, the composition $f \circ y$ is D-finite. These properties allow (among others) for proving special functions identities. For instance, one way to prove that two power series are equal

is to show that they are both solutions of a common linear differential equation, with the same initial conditions. Thus the computation is reduced to finitely many operations, as shown in the example below.

Example 2.1.1 (Identity proof with D-finite functions)**.** Let us prove the following identity:

$$\arcsin(x)^2 = \sum_{k\geqslant 0} \frac{k!}{\left(\frac{1}{2}\right)\dots\left(k+\frac{1}{2}\right)} \frac{x^{2k+2}}{2k+2}. \tag{2.3}$$

For that, one can proceed as follows:

- Consider the LODE for $y = \arcsin$: $(1-x^2)y'' - xy' = 0, \quad y(0) = 0, \quad y'(0) = 1$.
- Let $h = y^2$, then, by successive derivations obtain an LODE satisfied by h:

$$\begin{aligned} h' &= 2yy', \\ h'' &= 2y'^2 + 2yy'' = 2y'^2 + \frac{2x}{1-x^2}yy', \\ h''' &= 4y'y'' + \frac{2x}{1-x^2}(y'^2 + yy'') + \left(\frac{2}{1-x^2} + \frac{4x^2}{(1-x^2)^2}\right)yy', \\ &= \left(\frac{4x+2}{1-x^2} + \frac{6x^2}{(1-x^2)^2}\right)yy' + \frac{2x}{1-x^2}y'^2. \end{aligned}$$

- The vectors h, h', h'', h''' are linear combination of 3 vectors y^2, yy', y'^2. One can then compute a linear relation,

$$(1-x^2)h''' - 3xh'' - h' = 0.$$

- From this relation, one obtains the linear recurrence satisfied by the power series coefficients of h: $(n+1)(n+2)(n+3)h_{n+3} - (n+1)^3 h_{n+1} = 0$, with given initial conditions $h(0) = 0, \quad h'(0) = 0, \quad h''(0) = 2$. It then suffices to check that Equation (2.3) satisfies this recurrence. Otherwise, one could also directly solve the recurrence in this case, and obtain Equation (2.3).

Many (much more complicated) combinatoric and special functions identities can be proved algorithmically with D-finite functions and are beyond the scope of this report. We mention the fact that the degrees of the polynomial coefficients of the involved LODE can be quite high. A striking example in this context is the recent proof that the complete generating function for Gessel walks is algebraic [29], where for instance a candidate differential operator is of order 11 and (bivariate) polynomial coefficients of degrees up to 96 (and respectively 78) and with integer coefficients up to 61 digits.

2.1.2 Multivariate case

Let us now briefly discuss how D-finiteness can be generalized to the multivariate case. For that, note that Equation (2.1) reads in operator form $L \cdot y = 0$ and that one can obtain other operators which satisfy (2.1) by multiplying it with x or by differentiating it. So it is natural to represent such *classes of operators* in the *Weil algebra* $\mathfrak{D}_1 := \mathbb{K}[x]\langle\partial_x\rangle$, generated by $\{x, \partial_x\}$ and quotiented by the relation: $\partial_x x = x\partial_x + 1$, which represents the commutation rule coming from Leibniz law: $(xy(x))' = xy'(x) + y(x)$. Instead of a single operator L, we thus consider all the operators of $\mathfrak{D}_1$ which satisfy $L \cdot y = 0$. This is called the *annihilator* $\mathfrak{Ann}(y)$, which is a left ideal of $\mathfrak{D}_1$:

$$\mathfrak{Ann}(y) := \{L \in \mathfrak{D}_1 \mid Ly = 0\}.$$

Similarly, one can introduce a discrete analogue for the Weil algebra $\mathfrak{R}_1 := \mathbb{K}[\alpha]\langle S_\alpha\rangle$ (the polynomial coefficients are in α), which is a *shift algebra* for operators on P-recursive sequences, quotiented by the commutation relation $S_\alpha \alpha = \alpha S_\alpha + S_\alpha$. It can be seen that the action of a differential operator $L \in \mathfrak{D}_1$ on the generating function $\sum u_\alpha x^\alpha$ of a sequence $(u_\alpha)_{\alpha\in\mathbb{N}}$ corresponds to the action of an operator $S \in \mathfrak{R}_1$

on $(u_\alpha)_{\alpha\in\mathbb{N}}$, via the relations $\partial_x = (\alpha+1)S_\alpha$ and $x = S_\alpha^{-1}$ (we can always get rid of the negative powers of shift operators by multiplying through). This provides a formal correspondence between D-finite differential equations and P-recursive power series coefficients, which is further exploited in Section 2.3.

By analogy with the univariate case, we proceed to the multivariate one. From now on, we fix some notations related to multivariable and multiindices: let n be a positive integer for the ambient space $\mathbb{R}^n$, when there is no ambiguity with respect to the univariate case, $\mathbb{K}[x]$ is the ring of polynomials in the variables $x = (x_1, \dots, x_n)$ and let $\mathbb{K}[x]_d$ be the vector space of polynomials of total degree at most d. For every d, let $\mathbb{N}^n_d := \{\alpha \in \mathbb{N}^n : |\alpha| \leqslant d\}$, where $|\alpha| = \sum_i \alpha_i$. In a multivariate setting, we denote $x^\beta = x_1^{\beta_1} \dots x_n^{\beta_n}$ and $\partial_x^\alpha = \partial_{x_1}^{\alpha_1} \dots \partial_{x_n}^{\alpha_n}$ for $\alpha, \beta \in \mathbb{N}^n$. The derivative $\frac{\partial p}{\partial x_i}$ is denoted p_{x_i}.

To represent linear partial differential operators which annihilate a multivariate function $f : \mathbb{K}^n \to \mathbb{K}$, we consider first:

(i) The ring of differential operators with polynomial coefficients (the *n-th Weyl algebra*)

$$\mathfrak{D}_n := \mathbb{K}[x_1, \dots, x_n]\langle \partial_{x_1}, \dots, \partial_{x_n}\rangle,$$

generated by $\{x_1, \dots, x_n, \partial_{x_1}, \dots, \partial_{x_n}\}$ and quotiented by the relations:

$$\partial_{x_i} x_j = \begin{cases} x_i\partial_{x_i} + 1, & i = j, \\ x_j\partial_{x_i}, & i \neq j, \end{cases} \qquad x_i x_j = x_j x_i, \quad \partial_{x_i}\partial_{x_j} = \partial_{x_j}\partial_{x_i}.$$

We have that $\{x^\beta \partial_x^\alpha, \alpha, \beta \in \mathbb{N}^n\}$ is a basis of $\mathfrak{D}_n$ as a $\mathbb{K}$-vector space. If $L = \sum_{\alpha,\beta} c_{\alpha,\beta} x^\beta \partial_x^\alpha$, its *order* is the largest value of $|\alpha|$ such that there exists β with $c_{\alpha,\beta} \neq 0$.

Differential operators in $\mathfrak{D}_n$ naturally act on smooth functions f via $\partial_{x_i} f = f_{x_i} := \frac{\partial f}{\partial x_i}$. The *annihilator* $\mathfrak{Ann}(f)$ is a left ideal of $\mathfrak{D}_n$:

$$\mathfrak{Ann}(f) := \{L \in \mathfrak{D}_n \mid Lf = 0\}.$$

In the multivariate setting, a generalization of D-finiteness is the notion of *holonomicity*. Namely, a multivariate function f is called *holonomic* if there exists such an *annihilator* $\mathfrak{Ann}(f)$ in $\mathfrak{D}_n$, which is *holonomic*. This notion has its origin in D-module theory and we provide a formal definition for completeness.

Definition 2.1.2. Let $\mathfrak{I}$ be a left ideal of $\mathfrak{D}_n$. For $L \in \mathfrak{D}_n$, let $[L]_\mathfrak{I}$ denote the class of L in the quotient $\mathfrak{D}_n/\mathfrak{I}$. For $s \geqslant 0$, define

$$\left(\mathfrak{D}_n \diagup \mathfrak{I}\right)_s = \mathrm{Span}_\mathbb{K}\left\{[x^\beta\partial_x^\alpha]_\mathfrak{I}, |\alpha| + |\beta| \leqslant s\right\}.$$

Then there exists a polynomial $b(s) \in \mathbb{K}[s]$ such that $\dim_\mathbb{K}(\mathfrak{D}_n/\mathfrak{I})_s = b(s)$ for s large enough. The degree of $b(s)$ is called the *Bernstein dimension* of $\mathfrak{D}_n/\mathfrak{I}$. The left ideal $\mathfrak{I}$ is called *holonomic* if the Bernstein dimension of $\mathfrak{D}_n/\mathfrak{I}$ is equal to n.

When dealing with partial differential equations instead of operators, a *holonomic* system is a maximally overdetermined system, in the sense that there are as many linear partial differential equations with polynomial coefficients as possible.

Similarly, $\mathfrak{R}_n := \mathbb{K}[\alpha_1, \dots, \alpha_n]\langle S_{\alpha_1}, \dots, S_{\alpha_n}\rangle$ is the set of difference operators with polynomial coefficients in α, acting on sequences $u = (u(\gamma_1, \dots, \gamma_n))_{\gamma\in\mathbb{N}^n}$ via

$$\begin{aligned}(\alpha_i u)(\gamma_1, \dots, \gamma_n) &= \gamma_i u(\gamma_1, \dots, \gamma_n),\\ (S_{\alpha_i} u)(\gamma_1, \dots, \gamma_n) &= u(\gamma_1, \dots, \gamma_i + 1, \dots, \gamma_n), \quad \gamma \in \mathbb{N}^n.\end{aligned}$$

The annihilator $\mathfrak{Ann}(u) = \{R \in \mathfrak{R}_n \mid R\, u = 0\}$ is the set of recurrence relations satisfied by u, which is holonomic when its generating series is holonomic [53].

Using the theory of D-modules, similar closure properties (under addition, multiplication, definite integration or antiderivative with respect to a variable) to the univariate case can be proved. Unfortunately, the existent algorithms, which execute these closure properties using the representation with holonomic ideals, are not as efficient as those developed for another ring $\mathfrak{D}_n^*$ of differential operators defined below, which proves better suited algorithmically for certain classes of functions:

(ii) The ring of differential operators with rational fraction coefficients

$$\mathfrak{D}_n^* := \mathbb{K}(x_1, \ldots, x_n)\langle \partial_1, \ldots, \partial_n \rangle,$$

(rational differential Ore algebra) where the commutation rules of $\mathfrak{D}_n$ are extended by

$$\partial_{x_i} q(x) = q(x)\partial_{x_i} + \frac{\partial q(x)}{\partial x_i}, \qquad q(x) \in \mathbb{K}(x_1, \ldots, x_n).$$

One can also see $\mathfrak{Ann}(f)$ as a left ideal of $\mathfrak{D}_n^*$, and the quotient $\mathfrak{D}_n^*/\mathfrak{Ann}(f)$ as a $\mathbb{K}(x_1, \ldots, x_n)$-vector space. A smooth function f is called D-finite if $\mathfrak{D}_n^*/\mathfrak{Ann}(f)$ has finite dimension. Equivalently, its iterated derivatives $\{\partial_x^\alpha f, \alpha \in \mathbb{N}^n\}$ form a finite-dimensional vector space over rational fractions.

Both **(i)** and **(ii)** can be seen as so-called Ore algebras, where one can represent in a unified framework differential, difference (or mixed) operators [53, 115]. An Ore polynomial ring is obtained by applying Ore extensions to some base ring $\mathbf{A}$. An Ore extension adds a new symbol, say ∂ to the base ring: this gives a skew polynomial ring, whose elements are polynomials in the new symbol with coefficients in the base ring. The addition in this skew ring is the usual one, while the multiplication is defined by a specific commutation rule:

$$\partial a = \sigma(a)\partial + \delta(a), \quad \text{for all } a \in \mathbf{A},$$

where σ and δ are linear maps on the base ring, with specific properties (mainly they fulfill a skew Leibniz law). Note that in contrast to the Weyl algebra the non-commutativity, is now between the variables of the polynomial ring and its coefficients. Concrete rings of differential/difference operators are obtained depending on how ∂, σ, and δ act on functions. In **(i)** and **(ii)** (consider the univariate case for simplicity), $\sigma(f) = f$, $\delta(f) = \partial_x f$ and $\partial f = \delta(f)$. It is important to note that for **(i)** the base ring is $\mathbb{K}[x]$, while for **(ii)**, the base ring is also a field (of rational fractions) $\mathbb{K}(x)$.

In classical settings involving analytic functions, the distinction between **(i)** and **(ii)** is subtle[1] and often from an algorithmic point of view, it suffices to stick with the more suited rational Ore algebra, where rational coefficients allow to divide out polynomial contents and closure properties (sum, product, algebraic substitution) can be executed in a simpler manner. We refer to [53, 115, 163] for a more comprehensive presentation.

However, this distinction is essential when considering "generalized functions", for instance distributions. In this case, $\mathfrak{Ann}(f)$ can only be seen as a left ideal of $\mathfrak{D}_n$ and $\mathfrak{D}_n/\mathfrak{Ann}(f)$ as a $\mathbb{K}$-vector space. For example, the univariate Dirac distribution, defined by $\langle \delta, f \rangle = f(0)$, is annihilated (as a distribution) by x, since $\langle x\delta, f \rangle = \langle \delta, xf \rangle = 0$. However, a left ideal of $\mathfrak{D}_1^*$ containing x is necessarily $\mathfrak{D}_1^*$, but 1 annihilating δ would imply $\delta = 0$. Hence in this setting, which will be adopted in Section 2.3, the relevant notion is *holonomicity*.

Algorithms and software. An important algorithmic aspect is that the well-known notion of Gröbner bases was generalized to this non-commutative setting (see for example [80, 53, 115] and references therein). This is the building block of algorithms on executing closure operations [53, 115, 80, 206, 163]. Several software packages implement very efficient algorithms for manipulating such functions (especially in the case of rational Ore algebras), for instance `algolib` which contains in particular the packages `gfun` [192], `NumGfun` [145], `Ore_algebra` and `Mgfun` [53] in Maple; `HolonomicFunctions` [114] in Mathematica.

We now provide a more detailed review of related topics that we addressed in this area.

2.2 Efficient evaluation of Gaussians on disks

We became interested in the efficient and reliable computation of the following integral:

[1] A left ideal II of **(ii)** is D-finite iif its restriction to left ideals in **(i)** (i.e. $II \cap I$, where I is a left ideal in **(i)**) is holonomic [115, Thm.2.22].

$$\mathcal{P}_c = \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\bar{\mathcal{B}}((0,0),R)} \exp\left(-\frac{1}{2}\left(\frac{(x-x_m)^2}{\sigma_x^2} + \frac{(y-y_m)^2}{\sigma_y^2}\right)\right) \mathrm{d}x\mathrm{d}y, \tag{2.4}$$

which is a 2D integral over a disk of radius R centered at the origin of a Gaussian function with no cross-terms, for given parameters (x_m, y_m) and (σ_x, σ_y). As mentioned in the introduction, this integral appears when estimating the collision risk between two space objects, whose position has Gaussian-distributed uncertainty and under certain assumptions of the so-called short-encounter model. While several methods have been developed in the literature to compute this integral [71, 171, 3, 47], we provided a new efficient and reliable method which is based on a numerically stable and efficient power series evaluation.

2.2.1 Our approach

As a keen reader would expect, the employed series is obtained by the use of D-finite functions. Moreover, the evaluation scheme proposed also relies on analytical properties of our series. Particularly, this integral is analytically computed as the product of an exponential term with a convergent power series:

$$\mathcal{P}_c = \exp\left(-\frac{R^2}{2\sigma_y^2}\right) \sum_{k=0}^{+\infty} \beta_k R^{2k}, \tag{2.5}$$

where $(\beta_k)_{k\geqslant 0}$ is a positive P-recursive sequence, suitable for accurate numerical evaluation. Moreover, explicit bounds on the tail of this series are provided. In brief, the steps employed to obtain this formula together with the explicit recurrence for $(\beta_k)_{k\geqslant 0}$ are:

Step 1. Firstly, a method introduced by Lasserre and Zeron in [127] to integrate Gaussian functions over Euclidean balls is used. Specifically, Equation (2.4) is seen as a function $g(z)$, with $z = R^2$, whose Laplace transform $\mathcal{L}_g(\lambda) = \int_0^\infty \exp(-\lambda z) g(z) \mathrm{d}z$ can be computed in closed-form:

$$\mathcal{L}_g(\lambda) = \frac{\exp\left(-\lambda\left(\frac{x_m^2}{2\lambda\sigma_x^2+1} + \frac{y_m^2}{2\lambda\sigma_y^2+1}\right)\right)}{\lambda\sqrt{2\lambda\sigma_x^2+1}\sqrt{2\lambda\sigma_y^2+1}}, \text{ for all } \lambda \in \mathbb{C}, \text{ such that } \mathrm{Re}(\lambda) > 0. \tag{2.6}$$

This is then expanded in a power series at ∞. Classical results from [224] or [223, Chap. 2.14] allow for a term by term application of the inverse Laplace transform – also known as the Borel Transform of the sequence of coefficients of $\mathcal{L}_g$. This leads to a power series for the initial integral:

$$g(z) = \sum_{k=0}^{+\infty} \alpha_k z^k. \tag{2.7}$$

Step 2. In [127] however, no insight is given on how to obtain the coefficients $(\alpha_k)_{k\geqslant 0}$, besides the obvious way of computing the first ones based on explicit derivatives of $\mathcal{L}_g$. So, the second step consists in finding a simple form for these coefficients. For that, it suffices to remark that g is D-finite and hence, algorithmically obtain the recurrence formula satisfied by (α_k).

Step 3. Nevertheless, from a numerical point of view, the direct evaluation in finite precision of the series obtained for $g(z)$ can be difficult: although the power series expansion of $g(z)$ is convergent, the evaluation of the sum in finite precision arithmetic is prone to high cancellation [51, 84]. This means that if consecutive terms are close in magnitude, but of different signs, their sum in finite precision arithmetic contains very few correct significant digits. This makes the power series evaluation impractical for large values of z. This phenomenon was studied for various common functions e.g., $z \mapsto \exp(-z)$, the error function erf or the Airy function [51], but a workaround exists on a case by case basis, as exemplified below.

Example 2.2.1 (Cancellation in finite precision power series evaluation)**.** Consider the power series expansion $\exp(-x) = \sum_{i=0}^{\infty} \frac{(-1)^i x^i}{i!}$ and suppose one evaluates a truncation of this series at $x = 20$. Numerically, this gives:

$$\exp(-20) \simeq 1 - 20 \ldots + 1.66 \cdot 10^7 - 1.23 \cdot 10^7 + \ldots + 1.19 \cdot 10^{-8} - 3.45 \cdot 10^{-9} \ldots$$

One observes that the terms of the sum are alternating in sign and their maximum magnitude is very high compared to the actual value to be computed $\exp(-20) \simeq 2.06 \cdot 10^{-9}$. When evaluating in finite precision, the precision loss can be estimated by $\log \frac{\max_i |g_i z^i|}{|g(z)|}$, which in this case amounts to more than 53 bits that is, no correct digit is obtained when evaluating using the binary64 (double) FP format.

However, experts in numerical functions implementations will never use this series for evaluation. Instead, the series for $\exp(x)$ is used, which evaluates without cancellation (all the terms are positive) and then the reciprocal $1/\exp(x)$ is performed.

Similarly to this intuitive example, such a preconditioning was used in order to obtain a series suitable for low cancellation numerical evaluation for other functions [84, 51]. Instead of g, the function $f = hg$ is considered, where the so-called preconditioner h is D-finite too. This product is also D-finite by closure properties of D-finite functions. By carefully choosing the preconditioner, both h and f can be efficiently numerically evaluated. The heuristic for h is based on a method presented in [84] which uses several complex analysis results and properties obtained for g. In brief, we prove that $h(z) = \exp(pz)$ is a good choice for several values of p. For completeness, an overview of this method is presented in Section 2.2.2. Moreover, the choice $p = \frac{1}{2\sigma_y^2}$ leads to a simple proof of positivity of the series coefficients of f. Finally, explicit truncation error bounds for this series were proved, which also allowed for an *a priori* computation of the truncation order, when a pre-specified accuracy on the results is required.

These steps allow for building Algorithms 12 and 13, explained below and one obtains the following result:

Proposition 2.2.2. Algorithm 13 is correct that is, given $\delta > 0$ it returns an interval $\left[\underline{\mathcal{P}_c}, \overline{\mathcal{P}_c}\right]$ enclosing the evaluation of $\mathcal{P}_c \in \left[\underline{\mathcal{P}_c}, \overline{\mathcal{P}_c}\right]$, such that $\overline{\mathcal{P}_c} - \underline{\mathcal{P}_c} \leqslant \delta$. The algorithm runs in $O(\log_2(\delta^{-1}))$ i.e. linearly with respect to the number of required correct digits.

The proof is given in [J5] and here, we underline its main ingredients. Firstly, it makes use of Algorithm 12. The initial conditions of the recurrence satisfied by the terms of the power series of f are given in lines $2-5$ of Algorithm 12, while the loop in lines $6-8$ computes the truncated sum of a total of N positive terms by unrolling the recurrence. In line 13, one divides by the preconditionner to retrieve the original searched value of g. To obtain the recurrence unrolled in Algorithm 12, it suffices to use the algorithmic properties of D-finite functions: first, the Laplace transform of f, which corresponds to a translation in the Laplace domain: $\mathcal{L}_f(\lambda) = \mathcal{L}_g(\lambda - p)$ (for $\mathcal{L}_g$ given in Eq. 2.6) is D-finite. This can be checked by hand or obtained by the command `holexprtodiffeq` of Gfun [192]. Then, the coefficients of the series are obtained for instance with `diffeqtoreq` command of `Gfun`, followed by a Borel transform (which implies division by $n!$, hence the obtained sequence is still P-recursive).

Moreover, this is combined with the explicit truncation bounds in Algorithm 13. They correspond to closed-form evaluation of simple minorant/majorant series (whose coefficients are in geometric progression), which can be obtained in this specific case. Otherwise, majorant series tools like those of [147] could be used. Based on these closed-forms, an *a priori* sufficient bound of the number of terms, required for a specific accuracy, can in turn be computed. These quantities correspond to: l_n for lower bound on truncation, u_n for upper bound on truncation and n for a sufficient bound on the number of terms.

Before going further, it may be interesting to analyze a little bit further the "precondinionning method" employed above.

Algorithm 12 Approximate evaluation of $\mathcal{P}_c$.

Input: Parameters: σ_x, σ_y, x_m, y_m, R; Number of terms: N.
Output: $\tilde{\mathcal{P}}_c$ – truncated series evaluation of $\mathcal{P}_c$.

$p = \frac{1}{2\sigma_y^2}$; $\varphi = 1 - \dfrac{\sigma_y^2}{\sigma_x^2}$; $\omega_x = \dfrac{x_m^2}{4\sigma_x^4}$; $\omega_y = \dfrac{y_m^2}{4\sigma_y^4}$; $\alpha_0 = \dfrac{1}{2\sigma_x\sigma_y}\exp\left(-\dfrac{1}{2}\left(\dfrac{x_m^2}{\sigma_x^2} + \dfrac{y_m^2}{\sigma_y^2}\right)\right)$;

$c_0 = \alpha_0 R^2$;

$c_1 = \frac{\alpha_0 R^4}{2}\left(p\left(\frac{\varphi}{2}+1\right) + \omega_x + \omega_y\right)$;

$c_2 = \frac{\alpha_0 R^6}{12}\left(\left(p\left(\frac{\varphi}{2}+1\right) + \omega_x + \omega_y\right)^2 + p^2\left(\frac{\varphi^2}{2}+1\right) + 2p\varphi\omega_x\right)$;

$$\begin{aligned} c_3 &= \frac{\alpha_0 R^8}{144}\left(\left(p\left(\frac{\varphi}{2}+1\right) + \omega_x + \omega_y\right)^3 + 3\left(p\left(\frac{\varphi}{2}+1\right) + \omega_x + \omega_y\right)\left(p^2\left(\frac{\varphi^2}{2}+1\right) + 2p\varphi\omega_x\right)\right. \\ &+ \left. 2\left(p^3\left(\frac{\varphi^3}{2}+1\right) + 3p^2\varphi^2\omega_x\right)\right); \end{aligned}$$

for $k \leftarrow 0$ to $N-5$ **do**

$$\begin{aligned} c_{k+4} &= -\frac{R^8 p^3 \varphi^2 \omega_y}{(k+2)(k+3)(k+4)^2(k+5)} c_k + \frac{R^6 p^2 \varphi\left(p\varphi\left(k+\frac{5}{2}\right) + 2\omega_y\left(\frac{\varphi}{2}+1\right)\right)}{(k+3)(k+4)^2(k+5)} c_{k+1} \\ &- \frac{R^4 p\left(p\varphi\left(\frac{\varphi}{2}+1\right)(2k+5) + \varphi\left(2\omega_y + \frac{3p}{2}\right) + \omega_x + \omega_y\right)}{(k+4)^2(k+5)} c_{k+2} \\ &+ \frac{R^2\left(p(2\varphi+1)(k+3) + p\left(\frac{\varphi}{2}+1\right) + \omega_x + \omega_y\right)}{(k+4)(k+5)} c_{k+3} \end{aligned}$$

end for

$s \leftarrow 0$

for $k \leftarrow 0$ to $N-1$ **do**

$s \leftarrow s + c_k$;

end for

$\tilde{\mathcal{P}}_c \leftarrow \exp\left(-pR^2\right) s$;

return $\tilde{\mathcal{P}}_c$.

Algorithm 13 Evaluation of $\mathcal{P}_c$ with guaranteed accuracy.

Input: Parameters: σ_x, σ_y, x_m, y_m, R; Threshold δ.
Output: $\left[\underline{\mathcal{P}}_c, \overline{\mathcal{P}}_c\right]$ such that $\underline{\mathcal{P}}_c \leqslant \mathcal{P}_c \leqslant \overline{\mathcal{P}}_c$ and $\overline{\mathcal{P}}_c - \underline{\mathcal{P}}_c \leqslant \delta$.

$p = \frac{1}{2\sigma_y^2}$; $\varphi = 1 - \dfrac{\sigma_y^2}{\sigma_x^2}$; $\omega_x = \dfrac{x_m^2}{4\sigma_x^4}$; $\omega_y = \dfrac{y_m^2}{4\sigma_y^4}$; $\alpha_0 = \dfrac{1}{2\sigma_x\sigma_y}\exp\left(-\dfrac{1}{2}\left(\dfrac{x_m^2}{\sigma_x^2} + \dfrac{y_m^2}{\sigma_y^2}\right)\right)$;

$l_0 = \alpha_0 \dfrac{1 - \exp(-pR^2)}{p}$;

$u_0 = \alpha_0 \dfrac{\exp\left(p\left(\frac{\varphi}{2} + \frac{\omega_x+\omega_y}{p}\right)R^2\right) - \exp(-pR^2)}{p\left(1 + \frac{\varphi}{2} + \frac{\omega_x+\omega_y}{p}\right)}$;

if $u_0 - l_0 \leqslant \delta$ **then**

return $[l_0, u_0]$;

else

$N_1 = 2\left\lceil epR^2\left(1 + \frac{\varphi}{2} + \frac{\omega_x+\omega_y}{p}\right)\right\rceil$;

$N_2 = \left\lceil \log_2 \dfrac{\alpha_0 \exp\left(pR^2\left(\frac{\varphi}{2} + \frac{\omega_x+\omega_y}{p}\right)\right)}{\delta p\sqrt{2\pi N_1}\left(1 + \frac{\varphi}{2} + \frac{\omega_x+\omega_y}{p}\right)} \right\rceil$;

$n = \max\{N_1, N_2\} - 1$

$\tilde{\mathcal{P}}_c \leftarrow$ Algorithm 12 (σ_x, σ_y, x_m, y_m, R, n);

$l_n = \dfrac{\alpha_0 \exp(-pR^2)(pR^2)^{n+1}}{p(n+1)!}$;

$u_n = \dfrac{\alpha_0 \exp\left(p\left(\frac{\varphi}{2} + \frac{\omega_x+\omega_y}{p}\right)R^2\right)\left(p\left(1 + \frac{\varphi}{2} + \frac{\omega_x+\omega_y}{p}\right)R^2\right)^{n+1}}{p\left(1 + \frac{\varphi}{2} + \frac{\omega_x+\omega_y}{p}\right)(n+1)!}$;

$\underline{\mathcal{P}}_c = \tilde{\mathcal{P}}_c + l_n$;

$\overline{\mathcal{P}}_c = \tilde{\mathcal{P}}_c + u_n$;

return $\left[\underline{\mathcal{P}}_c, \overline{\mathcal{P}}_c\right]$.

end if

2.2.2 Some remarks about function evaluation without cancellation

We consider a Taylor series of the form

$$g(\xi) = \sum_{i=0}^{\infty} g_i \xi^i, g_i \sim (-1)^i \lambda \frac{\alpha^i}{i!^\kappa}, \tag{2.8}$$

where $\lambda, \alpha, \kappa > 0$. For large $\xi > 0$, the computation in finite precision arithmetic of such a sum is prone to *cancellation*. This is because the terms $\left|g_i \xi^i\right|$ are first growing, before the series starts to converge when $i^\kappa \geqslant \alpha\xi$. When $i^\kappa \simeq \alpha\xi$ the terms $\left|g_i \xi^i\right|$ usually get much larger than $|g(\xi)|$. So the leading bits cancel out while the lower-order bits which actually contribute to the first significant bits of the actual result get lost in the roundoff errors. So, we are interested in minimizing the ratio

$$d_g(\xi) = \log \frac{\max_i \left|g_i \xi^i\right|}{|g(\xi)|}. \tag{2.9}$$

When evaluating an entire function on some complex sector, Gawronski, Müller and Reinhard [84] provided an estimation of this ratio based on the asymptotic behaviour of this entire function. Under certain assumptions [84], one can estimate that for $\xi = re^{i\theta}$, and large values of r, $\log \left|g(re^{i\theta})\right| \sim h(\theta) r^\rho$, which gives the estimate:

$$d_g(re^{i\theta}) \sim r^\rho (\sigma_g - h_g(\theta)), \tag{2.10}$$

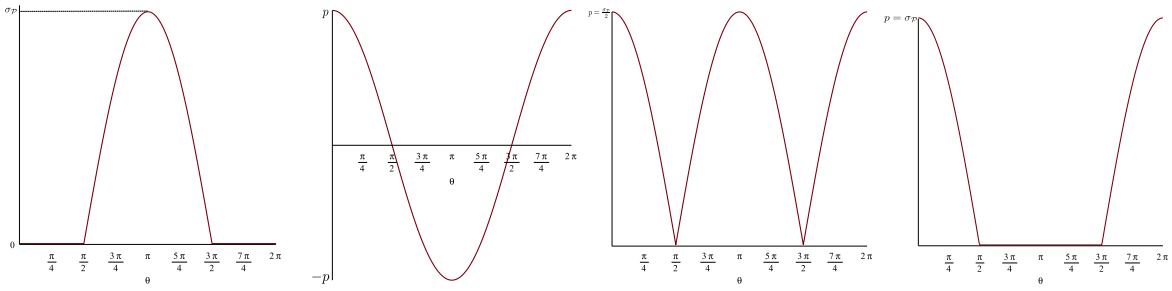
where:

- ρ is the order of g (roughly speaking ρ is the infimum of all m s.t. $g(\xi) = O(\exp(|\xi|^m)), r \to \infty$);
- h_g is the indicator function of g, which shows the growth along a ray; $h_g(\theta) = \limsup_{n\to\infty} \frac{\log|g(re^{i\theta})|}{r^\rho}$;
- σ_g is the type of g, which satisfies: $\max_{\theta\in[0,2\pi]} h_g(\theta) = \sigma_g$.

Hence, the heuristic to minimize the cancellation is to ensure that $\sigma_g - h_g(\theta)$ is small on the complex sector corresponding to θ. We use a simple instance of their method, since we focus only on positive real line evaluation and one can prove that in our case g is an entire function of order $\rho = 1$, also called *entire function of exponential type (EFET)*. In such a case, the indicator of g can explicitly be obtained with Polya theorem [133, Chapter 9], which relates the growth of the EFET function g along a ray, to the location of the singularities of its Inverse Borel transform.

The indicator function h_g is showed in Figure 2.1 (a). One can see that in this case, $d_g(r) = \sigma_g > 0$ and thus, the sum is not optimally conditioned to be evaluated on the real axis. The idea is to multiply g by some *preconditioner* function, such that both the preconditioner and their product are *very well conditioned* for evaluation at the considered sector (positive real line in our case). One possible choice for the preconditionner is $e^{p\xi}$ for which the indicator function is given in Figure 2.1(b) for $p > 0$. From the computation of the indicator function of the product, one can observe that it is well-conditioned for $\sigma_g \geqslant \sigma_g/2$. We give for example the indicator function obtained when $p = \frac{\sigma_g}{2}$ in Figure 2.1(c) and respectively $p = \sigma_g$ in Figure 2.1(d). For this problem, $p = \sigma_g$ is considered. Other choices are of course possible for preconditionners and currently, there is no established technique in the literature to assess which one is better. There are also "obvious bad choices": take for instance $\sin(\xi)$ which is entire of order 1 and indicator function $h_{\sin} = \omega\,|\sin(\theta)|$, $\omega > 0$. For real line evaluation, this will never give an optimal conditioning in zero.

Moreover, there is no guarantee that the obtained terms for the series are positive (since this estimate works asymptotically). In Chapter 4, future research directions on this topic are given. Let us now check some interesting numerical results.



(a) (b) (c) (d)

Figure 2.1 – Indicator functions of (a) probability g; (b) precondition $\xi \mapsto e^{p\xi}$; (c) $\xi \mapsto e^{\frac{\sigma_g}{2}\xi} g(\xi)$;

(d) $\xi \mapsto e^{\sigma_g \xi} g(\xi)$.

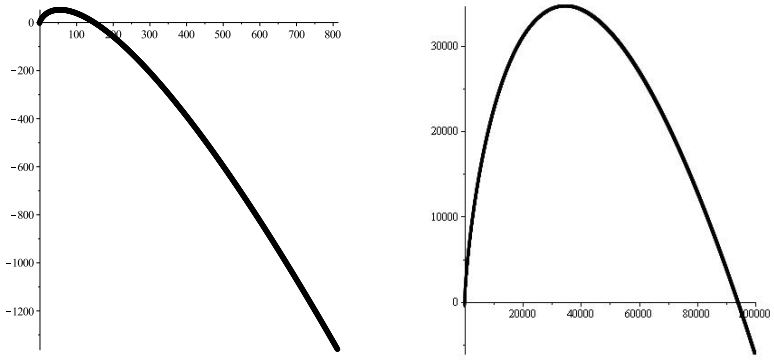
2.2.3 Challenging examples

Numerical results on relevant practical instances concerning collision probabilities between debris and satellites are postponed to Chap 3. Here we focus on two examples, recorded in Table 2.1, which are interesting from a theoretical viewpoint and which were considered as very challenging for accurate computing [6].

Case #	Input parameters (m)				
	σ_x	σ_y	R	x_m	y_m
3	114.25	1.41	15	0.15	3.88
5	177.8	0.038	10	2.12	-1.22

Table 2.1 – Two examples of [5].

Let us denote by $\eta_N = -\log \frac{u_N - l_N}{l_N + \widehat{\mathcal{P}}_c}$ the number of actual correct significant digits that can be certified when computing with N terms. For this example, our algorithm needs for instance 800 terms of the series in order to obtain $\eta_{800} = 30$ significant digits for Case 3 and respectively 121000 terms to obtain $\eta_{121000} = 20$ significant digits in the Case 5.



(a) Case 3 (b) Case 5

Figure 2.2 – Log-Magnitude of the terms $\log(c_i)$ in Algorithm 12 function of i, for Table 2.1 cases.

This is due mainly to two reasons: firstly, even though no cancellation occurs, the first terms of the series are very small and in order to compute the largest one (before they start to decrease again due to convergence), many terms need to be computed by unrolling the recurrence. For instance, the

log-magnitude of these terms $\log(c_i)$ is plotted in Figure 2.2 for these two cases. Secondly, the lower and upper bounds l_N and u_N are under/overestimating the actual tail of the series (although they are asymptotically tight). It would be thus interesting to provide alternatives to this method for this kind phenomenon proper to entire series. This will be discussed further in my research perspectives (Chap. 4).

2.2.4 Extensions and discussion

One observes that the integral (2.4) can be generalized to a higher (fixed) dimension n and it comes to computing the mass (i.e. the moment of order zero) of the restriction of an n-dimensional Gaussian measure to an n-dimensional disk with respect to the Euclidean norm.

A first question is whether similar computation methods can be developed in higher dimensions. For each fixed n, one can apply exactly the same procedure: Laplace transform, computing recurrences and initial conditions in the Laplace plane together with adapted evaluation strategies. Currently, we only provided a $3D$ preliminary result in [C13], which shows that the order of recurrences increases as well as the computational complexity related to D-finite algorithms (since more parameters are involved). A further study in this sense would be interesting (cf. Chapter 4).

On a different perspective, one can inquire about what happens for higher order moments, say

$$m_\alpha = \int_{D(0,R)} x^\alpha \mathrm{d}\mu_g(x), \quad \alpha \in \mathbb{N}^n, \tag{2.11}$$

where in the multivariate notation, $x = (x_1, \ldots, x_n)$, $x^\alpha = x_1^{\alpha_1} \ldots x_n^{\alpha_n}$, μ_g is the multivariate Gaussian measure and $D(0,R) = \{x \in \mathbb{R}^n : |\sum_i x_i^2| \leqslant R^2\}$.

Several questions may be asked:

1. Is there a similar way to efficiently compute m_α for a fixed index α?
2. Is (m_α) P-recursive, in which case, does an efficient algorithm exist for computing a system of linear recurrences satisfied by $(m_\alpha)_{\alpha\in\mathbb{N}^n}$?
3. More generally, if the density of μ_g is holonomic (i.e. it satisfies a holonomic system of linear partial or ordinary differential equations with polynomial coefficients), what can be said about the sequence $(m_\alpha)_{\alpha\in\mathbb{N}^n}$?
4. Finally, what happens when considering a more general support $G \subset \mathbb{R}^n$, which is a bounded open set of Euclidean space, whose boundary ∂G is algebraic (∂G is contained in the real zero set of finitely many polynomials), hence address the same issues for:

$$m_\alpha = \int_G x^\alpha \mathrm{d}\mu_g(x), \quad \alpha \in \mathbb{N}^n. \tag{2.12}$$

It is interesting to see that these or related questions have been recently studied from different points of view. Without being exhaustive, we note the versatile optimization approach of Lasserre [123], which formulates a linear programming problem on an appropriate space of measures, which is then solved via a hierarchy of semidefinite programs involving moments of the measures (also known as the Lasserre hierarchy) and thus proposes a numerical scheme to solve various optimization problems. In particular, in [124] this approach is applied for numerically computing Gaussian and exponential measures of semi-algebraic sets and thus proposes a numerical solution for question 1 above.

From the symbolic computation perspective, creative telescoping techniques [53, 115, 26, 163] can be employed to answer the questions 2–4. However, the numerical computation of initial conditions for the recurrences obtained as well as efficient numerical evaluation techniques are still subject of research and will be developed in Chapter 4.

In the sequel, we focus more on inverse problems related to items 2–4. Namely, in the framework of holonomic systems, we revisit and extend an approach of Lasserre and Putinar [126], concerning the inverse problem of support and/or density reconstruction when only a finite number of moments are given.

2.3 Direct and inverse moment problems with holonomic functions

Many reconstruction algorithms from moments of algebraic data were developed in optimization, analysis or statistics. Lasserre and Putinar [126] proposed an exact reconstruction algorithm for the algebraic support of the Lebesgue measure, or of measures with density equal to the exponential of a known polynomial. Their approach relies on linear recurrences for the moments, obtained using Stokes theorem. A natural question is whether this result can be generalized, as mentioned in [126]: *the analogy to the well understood moment rigidity of the Gaussian distribution is striking, although the constructive aspects of this finite determinateness remain too theoretical in general.* Motivated by this remark, in [C2], we extended this study to measures with holonomic densities and support with real algebraic boundary. Our approach relies on the algorithmic framework available for holonomic systems as presented in Section 2.1.2, as well as a generalized Stokes formula [126].

From the algebraic point of view, related works include well-studied algebraic methods for reconstruction problems involving measures with finitely many atoms (sums of Dirac) cf. [153] and references therein. In this case, their moment generating functions are solutions of systems of partial differential equations with constant coefficients, and the moments satisfy multi-index linear recurrences with constant coefficients. We also note that the reconstruction problem is solved in the univariate case, for piecewise D-finite densities in [13].

In our algorithms, holonomic distributions are involved and are briefly recalled in Section 2.3.1. Before stating more technical theorems and algorithms, we recall two introductory univariate examples from [C2]. In what follows, the indicator function of a set G, is denoted by $\mathbb{1}_G$.

Example 2.3.1 (Direct problem for erf-like function)**.** We are interested in computing a recurrence for the moments $m_i = \int_{-1}^{1} x^i e^{-x^2} \mathrm{d}x$. The idea is to include $\mathbb{1}_{[-1,1]}$ in the integral, and consider the distribution u corresponding to $\mathbb{1}_{[-1,1]}(x)e^{-x^2}$. Although not differentiable as a function, u satisfies as a distribution:

$$(1-x^2)(\partial_x + 2x)u = 0.$$

Now, note the Lagrange Identity [101] which is related to integration by parts: for a linear differential operator with polynomial coefficients, $L = c_r\partial_x^r + \ldots + c_0$, its adjoint is defined as $L^* = (-1)^r\partial_x^r c_r + \ldots + c_0$ and the following holds:

$$\varphi L(f) - L^*(\varphi)f = \partial_x(\mathcal{L}_L(f,\varphi)), \tag{2.13}$$

for any function φ and f, with an explicit linear differential operator $\mathcal{L}_L$.

Integrating for the test function x^i, using (2.13) and noticing that its right hand side vanishes after integration, one has:

$$\int_{-1}^{1} e^{-x^2}(\partial_x + 2x)^*((1-x^2)x^i)\mathrm{d}x = 0,$$

which directly provides the recurrence

$$im_{i-1} - (i+4)m_{i+1} + 2m_{i+3} = 0.$$

Example 2.3.2 (Univariate support and density reconstruction)**.** Consider the problem of reconstructing the parameters ξ_1, ξ_2 and p_2, p_1, p_0 provided that the first N moments $\{m_i, 0 \leqslant i \leqslant N\}$ are known:

$$m_i = \int_{\xi_1}^{\xi_2} x^i e^{p_2x^2 + p_1x + p_0}\mathrm{d}x. \tag{2.14}$$

Like in the previous example, $u = \mathbb{1}_{[\xi_1,\xi_2]}e^{p_2x^2+p_1x+p_0}$ satisfies:

$$(x-\xi_1)(x-\xi_2)(\partial_x - 2p_2x - p_1)u = 0.$$

Denote by $\hat{L} := g(x)\partial_x + h(x)$ the operator to be reconstructed such that $\hat{L}f = 0$, with $g(x) = x^2 + g_1 x + g_0$ and $h(x) = \sum_{i=0}^{3} h_i x^i$. Integrating and using Lagrange identity, one has:

$$\int_{-\infty}^{\infty} \left(g(x)\partial_x - h(x))(x^i)\right) u \mathrm{d}x = 0. \tag{2.15}$$

This gives, for each $i \geqslant 0$:

$$i m_{i+1} + i g_1 m_i + i g_0 m_{i-1} - h_3 m_{i+3} - h_2 m_{i+2} - h_1 m_{i+1} - h_0 m_i = 0. \tag{2.16}$$

Hence, the coefficients of g and h are solution of the above infinite linear system. If g is recovered, p (except for the coefficient p_0) could also be recovered from the division h/g. Finally the constant coefficient p_0 could also be recovered from the Equation (2.14), with $i = 0$.

Let us focus now on the general multivariate case and formalize the ideas presented in the above examples.

2.3.1 Holonomic distributions and their moments

Introduced by Schwartz [196], *distributions* generalize functions and measures.

Definition 2.3.3 (Test functions and distributions)**.** Let $\mathcal{E} = \mathcal{C}^\infty(\mathbb{R}^n)$ be the set of smooth functions over $\mathbb{R}^n$, equipped with the *compact-open topology*: $\varphi_k \to \varphi$ in $\mathcal{E}$ if $\partial_x^\alpha \varphi_k$ converges uniformly to $\partial_x^\alpha \varphi$ over every compact set, for each $\alpha \in \mathbb{N}^n$.

Its topological dual $\mathcal{E}'$ is the set of *compactly supported distributions* (or simply *distributions* in this work) i.e. linear forms $T : \mathcal{E} \to \mathbb{R}$ such that:

- There exists a minimal compact set $K \subseteq \mathbb{R}^n$ (the *support* of T) such that $\langle T, \varphi\rangle = 0$ whenever φ vanishes over K.
- $\langle T, \varphi_k\rangle \to 0$ whenever $\varphi_k \to 0$ in $\mathcal{E}$.

$\mathcal{E}'$ has a canonical $\mathfrak{D}_n$-module structure:

$$\langle L\,T, \varphi\rangle := \langle T, L^*\varphi\rangle, \qquad L \in \mathfrak{D}_n,\ T \in \mathcal{E}',\ \varphi \in \mathcal{E}, \tag{2.17}$$

where the *adjoint operator* L^* is defined by

$$x_i^* = x_i, \qquad \partial_{x_i}^* = -\partial_{x_i}, \quad \text{and} \quad (L_1 L_2)^* = L_2^* L_1^*.$$

Definition 2.3.4 (Holonomic distribution)**.** A distribution $T \in \mathcal{E}'$ is holonomic if its annihilator is a holonomic ideal of $\mathfrak{D}_n$:

$$\mathfrak{Ann}(T) := \{L \in \mathfrak{D}_n \mid L\,T = 0 \text{ as a distribution}\}.$$

A measure supported on a set G, with density $f \in \mathcal{E}$, is represented by the distribution $f\mathbb{1}_G$, with $\langle f\mathbb{1}_G, \varphi\rangle = \int_G \varphi(x) f(x) \mathrm{d}x$.

We make the following assumption on $G \subseteq \mathbb{R}^n$, which is necessary for the correctness proofs in [C2]:

Assumption 2.3.5. G is a compact n-dimensional semi-algebraic set. In particular, the following holds:

(1) G is an n-dimensional compact manifold such that its boundary can be decomposed as $\partial G = Z \cup Z'$, with Z a finite union of $(n-1)$-dimensional manifolds and Z' a negligible set w.r.t the $(n-1)$-dimensional Hausdorff measure.

(2) the ideal of polynomials vanishing over ∂G is radical and principal i.e., generated by a single square-free polynomial g. In particular, the family $\{g, g_{x_1}, \ldots, g_{x_n}\}$ is coprime, implying that the set of singular points $\{x \mid g(x) = 0 \text{ and } \nabla g(x) = 0\}$ is negligible in ∂G.

Definition 2.3.6 (Moments of a compactly supported distribution)**.** The moments of a distribution $T \in \mathcal{E}'$ are:

$$m_\alpha(T) := \langle T, x^\alpha \rangle, \qquad \alpha \in \mathbb{N}^n. \tag{2.18}$$

Note that if $T = f\mathbb{1}_G$ with G compact and $f \in \mathcal{E}$, then $m_\alpha(f\mathbb{1}_G)$ coincides with the moments defined in Equation (2.12).

A convenient way to deal with moments of a distribution is the *Fourier transform* (also called *characteristic function*).

Definition 2.3.7. The Fourier transform of a distribution $T \in \mathcal{E}'$ is the analytic function $\mathscr{F}\{T\}$ of $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$ defined by:

$$\mathscr{F}\{T\}(z) = \sum_{\alpha \in \mathbb{N}^n} m_\alpha(T) \frac{(-\boldsymbol{I}\, z)^\alpha}{\alpha!} = \langle T, e^{-\boldsymbol{I}\, x \cdot z} \rangle, \qquad z \in \mathbb{R}^n.$$

Proposition 2.3.8. Let $T \in \mathcal{E}'$ and $L = \sum_\beta q_\beta(x)\partial_x^\beta \in \mathfrak{D}_n$.

(i) The Fourier transform of $L\,T$ is related to that of T by

$$\mathscr{F}\{L\,T\} = L^{\mathscr{F}}\mathscr{F}\{T\}, \qquad \text{with}$$
$$L^{\mathscr{F}} = L\left[\begin{array}{ll} x_i & \mapsto \boldsymbol{I}\,\partial_{z_i} \\ \partial_{x_i} & \mapsto \boldsymbol{I}\, z_i \end{array}\right] = \sum_\beta q_\beta(\boldsymbol{I}\,\partial_z)(\boldsymbol{I}\, z)^\beta. \tag{2.19}$$

(ii) The moments of $L\,T$ are related to those of T by

$$(m_\alpha(L\,T)) = L^{\mathscr{M}}(m_\alpha(T)), \qquad \text{with}$$
$$L^{\mathscr{M}} = L\left[\begin{array}{ll} x_i & \mapsto S_{\alpha_i} \\ \partial_{x_i} & \mapsto -\alpha_i S_{\alpha_i}^{-1} \end{array}\right] = \sum_\beta (-1)^{|\beta|} q_\beta(S_\alpha)\left(\prod_{i=1}^n \left(\alpha_i S_{\alpha_i}^{-1}\right)^{\beta_i}\right), \tag{2.20}$$

Proposition 2.3.9. Let $T \in \mathcal{E}'$. An operator $L \in \mathfrak{D}_n$ satisfies

$$\langle T, L^* x^\alpha \rangle = 0, \qquad \text{for all } \alpha \in \mathbb{N}^n, \tag{2.21}$$

if and only if $L \in \mathfrak{Ann}(T)$.

Finally, the following proposition provides differential equations for measures supported on semi-algebraic sets. Its proof is given in [C2] and is based on Lagrange's identity and Stokes' theorem.

Proposition 2.3.10. [C2, Prop.3.3] Let G and g as in Assumption 2.3.5, $f \in \mathcal{E}$, $L \in \mathfrak{Ann}(f)$ of order r. Then $g^r L \in \mathfrak{Ann}(f\mathbb{1}_G)$.

Hence, Proposition 2.3.10 gives an easy way to construct operators in $\mathfrak{Ann}(f\mathbb{1}_G)$ from operators in $\mathfrak{Ann}(f)$. Indeed, given a Gröbner basis $\{L_1, \ldots, L_k\}$ of $\mathfrak{Ann}(f)$, and $g \in \mathbb{R}[x]$ vanishing over ∂G, each operator $g^{r_i}L_i$ (with r_i the order of L_i) annihilates $f\mathbb{1}_G$ as a distribution. Therefore, each operator $R_i := (g^{r_i}L_i)^{\mathscr{M}}$ gives a valid recurrence for the sequence of moments (m_α). Together with Proposition 2.3.8 and in particular Equation (2.20), this provides a method for computing linear recurrences for the moments. The advantage is that when the coefficients of the polynomial g are given as parameters, the obtained recurrences remain linear with respect to them.

However, from the fact that f is holonomic, one can not directly guarantee that the ideal generated by $\{g^{r_1}L_1, \ldots, g^{r_k}L_k\}$ is holonomic. Similarly, we are not able to prove (or refute) that $\{R_1, \ldots, R_k\}$ is holonomic in general. Nevertheless, one can apply a Gröbner basis algorithm, which will possibly return such a basis. This heuristic is proposed in Algorithm 14.

We proved that this algorithm returns a Gröbner basis of a holonomic ideal, in the particular case of an exponential-polynomial density (including the Lebesgue measure), and a smooth boundary,

Algorithm 14 RECURRENCESMOMENTS($n, g, \{L_1, \ldots, L_k\}$)

Input: Gröbner basis $\{L_1, \ldots, L_k\}$ for $\mathfrak{Ann}(f)$, g.
Output: Gröbner basis for $\mathfrak{Ann}(m_\alpha)$.

1: $R_i \leftarrow (g^{r_i} L_i)^{\mathscr{M}}$, as in (2.20), with r_i the order of L_i, for $i \in [1\,..\,k]$
2: **return** GröbnerBasis($\{R_1, \ldots, R_k\}$,$\mathfrak{R}_n$)

extending [163, Prop. 4]. Note that creative telescoping algorithms of Oaku [163], are proven to compute a holonomic system, but when the coefficients of the polynomial g are given as parameters, the obtained recurrences are not linear with respect to them.

However, having a Gröbner basis is not mandatory for the reconstruction problems addressed in the next section and the recurrences obtained as above turn out to be sufficient and constitute the basic brick of our reconstruction method.

2.3.2 Reconstruction methods

The general reconstruction problem considered is the following:

Problem 1 (General Inverse Problem)**.** Let $\mu_f = f\mathbb{1}_G \mathrm{d}x$ be a measure supported on a compact semi-algebraic set G, with *holonomic* f. Given a finite number of moments m_α, $|\alpha| \leqslant N$, recover a polynomial $g \in \mathbb{K}[x]$ vanishing on the algebraic boundary of G and the coefficients of a holonomic system satisfied by f.

The strategy proposed is:

- Take an *ansatz* $L' = \sum_{\beta \in A} q_\beta(x)\partial_x^\beta$, for a specified finite set $A \subset \mathbb{N}^n$ and polynomials $q_\beta(x)$ with specified degrees d_β.
- Let $R = L'^{\mathscr{M}}$ (see Equation (2.20)). Solve a finite-dimensional linear system in the unknown coefficients of the polynomials q_β:

$$(R\,m(f\mathbb{1}_G))_\alpha = 0, \qquad |\alpha| \leqslant N. \tag{2.22}$$

  This requires the knowledge of moments $m_\alpha(f\mathbb{1}_G)$ with $|\alpha| \leqslant N + \max_{\beta \in A}\{d_\beta - |\beta|\}$ (see Equation (2.20)).
- From the solution L' of (2.22), extract a polynomial $\tilde{g}$ vanishing on ∂G and an operator $\tilde{L} \in \mathfrak{Ann}(f)$.

Note that the solution of the system (2.22) corresponds to a truncation of the infinite system (2.21), since $\langle f\mathbb{1}_G, L'^* x^\alpha \rangle = 0$, for $|\alpha| \leqslant N$. Hence, one is interested in obtaining bounds $\hat{N}$ on N, such that any solution of (2.22) is also solution of (2.21). Such an *a priori* uniform bound depending only on A and d_β does not exist in general.

Remark 2.3.11. As noted in [13], for general holonomic operators L with $r > 1$, the number N of required moments, in order to correctly recover the parameters, might depend also on specific coefficients of L. An example is the nth Legendre polynomial, whose first n moments (taken over $[-1, 1]$) vanish, while $L_n = \partial_x((1 - x^2)\partial_x) + n(n + 1)$. Hence the reconstruction of μ_f depends also on the parameter n, which is part of the definition of L_n. On the contrary, for *exponential-polynomial* case, we show that N depends only on the degrees of the polynomials involved.

Another issue detailed in [C2] is that L' may not be factorized as $\tilde{g}(x)^r \tilde{L}$ with $\tilde{g}$ vanishing on ∂G and $\tilde{L}f = 0$.

These issues can be solved when f is exponential-polynomial:

Problem 2 (Exp-Poly Inverse Problem). Let a measure $\mu_f = f\mathbb{1}_G \mathrm{d}x$, supported on a compact semi-algebraic set G, whose algebraic boundary is included in the zero set of a polynomial $g \in \mathbb{K}[x]_d$. Let $f = \exp(p)$, with $p \in \mathbb{K}[x]_s$. Given s, d, and a finite number of moments m_α, $|\alpha| \leqslant N$, recover the coefficients of both g and p.

Algorithm 15 and Theorem 2.3.12, proved in [C2], show that this reconstruction problem boils down to solving a linear system of $3d + s - 1$ equations, involving moments up to degree $|\alpha| \leqslant 4d + 2(s-1)$.

Algorithm 15 RECONSTRUCTEXPPOLY$(n, d, s, N, (m_\alpha)_{|\alpha| \leqslant N+d+s-1})$

Input: $n \geqslant 2$, degrees $d, s \geqslant 0$, moments m_α for $|\alpha| \leqslant N + d + s - 1$.
Output: $\tilde{g}, \tilde{p} \in \mathbb{K}[x]$ with $\deg(\tilde{g}) \leqslant d$ and $\deg(\tilde{p}) \leqslant s$.

▷ *Find* $L_i' \in \mathfrak{Ann}(f\mathbb{1}_G)$
1: $h_0 \leftarrow \sum_{|\gamma| \leqslant d} h_{0\gamma} x^\gamma$ and $h_i \leftarrow \sum_{|\gamma| \leqslant d+s-1} h_{i\gamma} x^\gamma$ for $i \in [1\,..\,n]$, with symbolic coefficients $h_{i\gamma}$
2: $L_i' \leftarrow h_0 \partial_{x_i} - h_i$ for $i \in [1\,..\,n]$
3: Find a nontrivial solution $\{h_{i\gamma}\}$ of the linear system:

$$(L_i' \overset{\mathscr{M}}{\cdot} m)_\alpha = 0, \qquad i \in [1\,..\,n],\ |\alpha| \leqslant N$$

▷ *Reconstruct* $\tilde{g}$ *and* $\tilde{p}$
4: $\tilde{g} \leftarrow h_0$ and $\tilde{p}_i \leftarrow h_i/\tilde{g}$ for $i \in [1\,..\,n]$
5: $\tilde{p} \leftarrow \sum_{i=1}^{n} \int_0^{x_i} \tilde{p}_i(0, \ldots, 0, t_i, x_{i+1}, \ldots, x_n)\mathrm{d}t_i$
6: **return** $(\tilde{g}, \tilde{p})$

Theorem 2.3.12. [C2, Thm. 4.1] Let $f(x) = \exp(p(x))$ with $\deg p = s$, and G, g with $\deg g = d$ be as in Assumption 2.3.5. If $N \geqslant \hat{N} = 3d + s - 1$, then RECONSTRUCTEXPPOLY$(n, d, s, N, (m_\alpha))$ returns $\tilde{g} = \lambda g$ with $\lambda \in \mathbb{K}^*$, and $\tilde{p} = p - p(0)$. This requires moments up to degree $4d + 2(s-1)$.

Moreover, if $g \geqslant 0$ over G, $\hat{N}$ can be only $2d + s - 1$, requiring moments up to degree $3d + 2(s-1)$.

The general holonomic case is addressed in two steps: firstly, for recovering the density, it can be proved that N is finite, but no *a priori* bound for it is known; secondly, once the density is known, a stronger result is proved for the support reconstruction, since an explicit uniform bound on the number of required moments is given. The corresponding theorems proved in [C2] are summarized below. For the support reconstruction, the following assumption is made. Roughly speaking, the differential system must not be singular over the Zariski closure of ∂G, except for a zero-measure set.

Assumption 2.3.13. The pair $\{g, q_{i,r}\}$ is coprime for each $i \in [1\,..\,n]$.

Theorem 2.3.14. [C2, Thm. 4.4] Let $i \in [1\,..\,n]$, f analytic, $G, g \in \mathbb{K}[x]_d$ satisfying Assumption 2.3.5, and

$$L = \sum_{j=0}^{r} q_j(x) \partial_{x_i}^j \in \mathfrak{Ann}(f) \cap \mathbb{K}[x]\langle \partial_{x_i} \rangle,$$

of minimal order r, with q_r of minimal degree. Then, Algorithm RECONSTRUCTDENSITY$(n, i, r, s, N, (m_\alpha))$ returns $\tilde{L} = \lambda L$ with $\lambda \in \mathbb{K}^*$ for $s \geqslant dr + \max\{\deg(q_j)\}$ and N large enough.

Theorem 2.3.15. [C2, Thm. 4.5] Let analytic f be annihilated by the order r rectangular system $\{L_1, \ldots, L_n\}$, G be as in Assumption 2.3.5 with $g \in \mathbb{K}[x]$ of degree d, and assume also Assumption 2.3.13. Then, for $N \geqslant \hat{N} := (2r-1)d + (d-1)b + s$, with $b = r \mod 2$ and $s = \max\{q_{i,r}\}$, RECONSTRUCTSUPPORT$(n, d, r, \{L_i\}, N, (m_\alpha))$ returns $\tilde{g} = \lambda g$ with $\lambda \in \mathbb{K}^*$.

In particular, this proves that when the density is known, the support can be reconstructed using moments up to degree $(3r-1)d + (d-1)b + s + \max_{ij}\{\deg(q_{i,j}) - j\}$.

In the algorithms proposed, "exact computations" are assumed, that is, both the polynomial coefficients and the given moments m_α lie in a computable finite extension of $\mathbb{Q}$. The practical case of approximately known numerical moments was for the moment only briefly analyzed. A final example in this sense is presented (some others are available in [C2]), while a thorough robustness analysis remains to be done.

Algorithm 16 RECONSTRUCTDENSITY$(n, i, r, s, N, (m_\alpha)_{|\alpha|\leqslant N+s})$

Input: $n \geqslant 2$, $i \in [1\,..\,n]$, order r, maximum degree s, moments m_α for $|\alpha| \leqslant N+s$.

Output: $\tilde{L} = \sum_{j=0}^{r} \tilde{q}_j(x)\partial_{x_j}^j$ with $\deg(\tilde{q}_j) \leqslant s$.

▷ *Find* $L' \in \mathfrak{Ann}(f\mathbb{1}_G) \cap \mathbb{K}[x]\langle\partial_{x_i}\rangle$

1: $h_j \leftarrow \sum_{|\gamma|\leqslant s} h_{j\gamma} x^\gamma$ for $j \in [0\,..\,r]$ with symbolic coefficients $h_{j\gamma}$

2: $L' \leftarrow \sum_{j=0}^{r} h_j(x)\partial_{x_i}^j$

3: Find a nontrivial solution $\{h_{j\gamma}\}$ of the linear system:

$$(L'^{\mathscr{M}} m)_\alpha = 0, \qquad |\alpha| \leqslant N$$

▷ *Extract minimal* $L \in \mathfrak{Ann}(f) \cap \mathbb{K}[x]\langle\partial_{x_i}\rangle$

4: $\ell \leftarrow \mathrm{GCD}(h_0, \ldots, h_r)$ and $\tilde{q}_j \leftarrow h_j/\ell$ for $j \in [1\,..\,n]$.

5: **return** $\tilde{L} = \sum_{j=0}^{r} \tilde{q}_j(x)\partial_{x_j}^j$

Algorithm 17 RECONSTRUCTSUPPORT$(n, d, r, \{L_i\}_{i=1}^n, N, (m_\alpha))$

Input: $n \geqslant 2$, degree d, order r, rectangular system $\{L_1, \ldots, L_n\}$ of order r, moments m_α for $|\alpha| \leqslant N + dr + \max_{ij}\{\deg(q_{i,j}) - j\}$.

Output: polynomial $\tilde{g}(x) \in \mathbb{K}[x]_d$ vanishing over ∂G.

1: $h \leftarrow \sum_{|\gamma|\leqslant dr} h_\gamma x^\gamma$ with symbolic coefficients h_γ

2: Find a nontrivial solution $\{h_\gamma\}$ of the linear system:

$$\left((hL_i)^{\mathscr{M}} m\right)_\alpha = 0, \qquad |\alpha| \leqslant N, i \in [1\,..\,n]$$

3: $\tilde{g} \leftarrow h/\mathrm{GCD}(h, h_{x_1}, \ldots, h_{x_n})$

4: **return** $\tilde{g}$

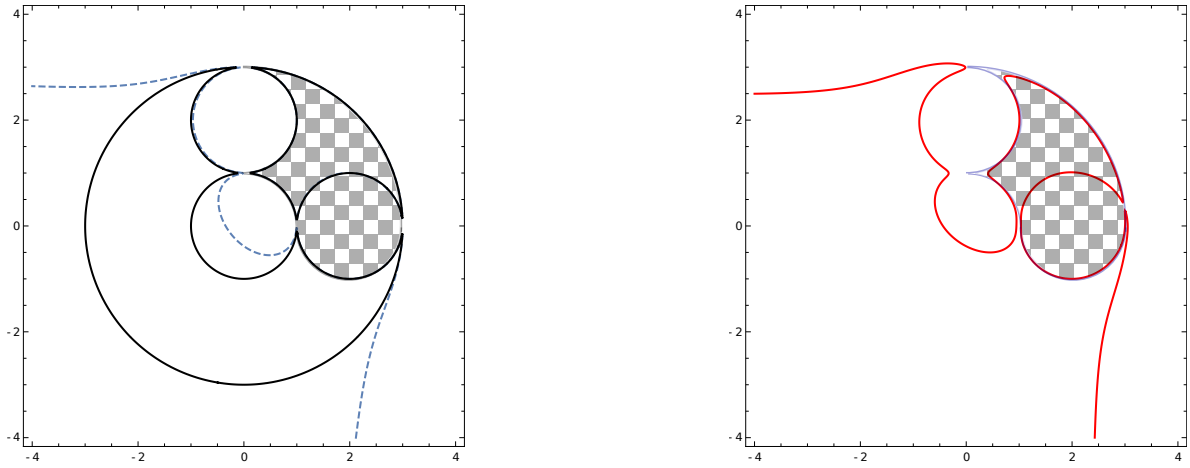
2.3.3 Example and conclusion

Our implementation[2] uses `OreAlgebra` and `OreGroebnerBasis` routines from the `Holonomic Functions` library [114]. The exactly computed moments m_{ij} (obtained from the recurrences given by Algorithm 14 together with closed-form initial conditions, when possible) are truncated to $\tilde{m}_{ij}$, s.t. $\lfloor -\log_{10} \frac{m_{i,j}-\tilde{m}_{ij}}{m_{ij}} \rfloor = \varepsilon$ i.e., ε represents the number of correct digits of $\tilde{m}_{ij}$.

In a second time, Algorithm 15 solves the inverse problem given the approximate $\tilde{m}_{ij}$. For numerically solving the resulting overdetermined systems of linear equations, we employ a Least Mean Squares method of Mathematica.

Example 2.3.16 (Algebraic Support, Lebesgue measure)**.** Consider the moments $m_{ij} = \int_G x^i y^j \mathrm{d}x\mathrm{d}y$, with respect to the Lebesgue measure, with G depicted with the checkered pattern in Figure 2.3.

[2] The corresponding code is available at `http://homepages.laas.fr/fbrehard/HolonomicMomentProblem`



(a) (b)

Figure 2.3 – (a) G in checkered pattern, together with ∂G in black. For $\varepsilon > 4$: reconstructed and original boundary cannot be distinguished at this scale; in dashed-blue, $\varepsilon = 4$, while in red (b) $\varepsilon = 2$.

(i) *Direct problem*: Given $g = (x^2+y^2-1)(x^2+y^2-9)(x^2+(y-2)^2-1)((x-2)^2+y^2-1)$, which vanishes on ∂G, and $\mathfrak{Ann}\{1\} = \{\partial_x, \partial_y\}$, Algorithm 14 returns a Gröbner basis with 9 generators and with 36 *monomials under the staircase*: $\{S_i^k S_j^l,\ k,l \in \mathbb{N}, k+l \leqslant 7\}$.

(ii) *Inverse problem*: Suppose now given a finite number of numerically computed moments $\tilde{m}_{ij}$ of the Lebesgue measure with unknown support G. The goal is to reconstruct $g = \sum_{i+j\leqslant 8} g_{ij} x^i y^j$ which vanishes on ∂G. The results of Algorithm 15 called with parameters $(2, 8, 0, 22, (\tilde{m}_{ij})_{|i+j|\leqslant 29})$ are depicted in Figure 2.3: the reconstructed boundary cannot be distinguished from the exact one at the drawing scale, when the moments $\tilde{m}_{ij}$ are given with more than 4 correct digits. When $2 \leqslant \varepsilon \leqslant 4$, the actual geometric boundary of G, can still be very well reconstructed, although the algebraic boundary is degraded.

The proposed method is very robust on the above academic examples, but a further study is needed for an efficient implementation in practical higher-dimensional cases. On the theoretical side, this study provided further insight on the question raised in [126] regarding the *finite determinateness of a measure*. To sum up, provided Assumptions 2.3.5 and 2.3.13 hold, for a measure with compact algebraic support G, with $g \in \mathbb{K}[x]_d$ vanishing on ∂G and known holonomic density f, the moments up to degree N (which only depends on d and the order of a rectangular differential system which annihilates f) determine in a constructive and robust manner the coefficients of g. Thus, this determines in turn all the other moments. When both the density and the support are unknown, a uniform bound N does not exist in general. We provided the solution for the special case of unknown *exponential-polynomial* density.

From power to orthogonal series. Broadly speaking, the above contributions exploit the algebraic properties of power series solutions of holonomic linear differential (systems) of equations. Another natural aspect is to consider other orthogonal series expansions. In my research, this was dealt with from the point of view of polynomial approximation as discussed below.

2.4 Rigorous Polynomial Approximations

When computing *approximate solutions*, one often uses polynomial approximations. This is frequently useful since such an approximation may be more compact to represent and store, and also more efficient to evaluate and manipulate. Concerning evaluation, polynomial approximations are especially important. In general, the basic functions that are implemented in hardware on a processor are limited to addition, subtraction, multiplication, and sometimes division. Moreover, division is significantly slower than multiplication. The only functions of one variable that one may evaluate using a bounded number

of additions/subtractions, multiplications and comparisons are piecewise polynomials: hence, on such systems, polynomial approximations are not only a good choice for implementing more complex mathematical functions, they are frequently the only one that makes sense. Polynomial approximations for common functions used to be tabulated in handbooks [1]. Nowadays, most computer algebra systems provide routines for obtaining polynomial approximations for such functions. However, when bounds for the approximation errors are available, they are not guaranteed to be accurate and are sometimes unreliable.

An important part of my work deals with solving this shortcoming using *Rigorous Polynomial Approximations* (RPAs): a polynomial approximation together with rigorous error bounds. More specifically, an RPA for a function f defined over an interval I is a couple (P, Δ) where P is a polynomial and Δ is an interval such that $f(x) - P(x) \in \Delta$, for all $x \in I$.

In this framework, D-finite functions play an important role, since (as mentioned in Section 2.1) for this class of functions, many recent efficient algorithms provide both polynomial approximations and rigorous error bound. When expanding in power series a given D-finite function f, efficient algorithms were developed for computing its coefficients and also for providing effective bounds on the tail inside the disk of convergence of the series. From the linear recurrence on the coefficients, one can produce majorant series whose speed of convergence is under control [218, 148] (an example for such majorant series was given in 2.2). Moreover, analytic continuation provides a generalization of this process outside the disk of convergence. Reliable approximations can be obtained on any disk containing no singularities of f, given a polygonal path starting from the origin, which avoids the (finitely many) singularities of the equation and such that on each of its vertices, translated arbitrarily precise initial conditions for the same differential equation can be computed. This process was automatically and efficiently implemented [148, 147] in computer algebra systems like Maple or Sage.

RPAs constructed on power series, can be obtained also for non D-finite functions, when such functions are represented as an expression tree composed of *basic* functions (D-finite or other functions for which RPAs can be constructed) together with arithmetic operations (addition, multiplication, division, square-root). Recursively evaluating such a tree, by overloading corresponging operators with operations on polynomials and error bounds, is known as *Taylor model algebra*. This tool was made popular by Makino and Berz and similar ideas were employed in a series of articles of several authors [140, 160, 141]. During my PhD, I already analyzed, implemented and proposed improvements for Taylor models [R3], then proposed with N. Brisebarre [C17] a new so-called Chebyshev models (CMs) algebra, using Chebyshev interpolants.

The main idea is that despite its simplicity, Taylor expansion has several drawbacks when uniformly approximating f over a given compact interval. When f is not smooth enough on the disc surrounding the considered interval, convergence cannot be ensured and one needs to suitably split the interval and provide a Taylor series for each subsegment (via analytic continuation for instance). Moreover, even when convergent, the n-th order truncated Taylor series of f is usually not the best uniform polynomial approximation of degree n over the segment under consideration. From this point of view, truncated Chebyshev series or Chebyshev interpolants prove to be a better choice and excellent accounts on this topic are given in [33, 49, 76, 143, 177, 211, 186].

Subsequently, we proposed in [J4] a method for computing CMs for *D-finite functions*. The key ingredient of our method is that the coefficients of Chebyshev series expansions also satisfy linear recurrences with polynomial coefficients (like power series do). However, these recurrences do not yield a direct computation of the coefficients, owing chiefly to the lack of initial conditions. Therefore, we used a combination of a classical method dating back to Clenshaw, revisited in light of the properties of the recurrence relations we consider, and a rigorous enclosure method for the solution of the differential equation based on a fixed point theorem. One more important contribution of this work is that the algorithms for obtaining these RPAs have a linear arithmetic complexity.

2.4.1 Chebyshev expansions of D-finite functions

Chebyshev polynomials and Chebyshev series

The Chebyshev family of polynomials is defined using the following three-term recurrence relation:

$$T_0(X) = 1, \quad T_1(X) = X, \tag{2.23}$$
$$T_{n+2}(X) = 2XT_{n+1}(X) - T_n(X), \quad n \geqslant 0, \tag{2.24}$$

which gives a basis for $\mathbb{R}[X]$. Equivalently, T_n is defined to be the only polynomial satisfying $T_n(\cos(\theta)) = \cos(n\theta)$ for all $\theta \in \mathbb{R}$. In particular, one gets that $|T_n(t)| \leqslant 1$ for all $t \in [-1, 1]$. To obtain more symmetric formulas, one can define $T_{-n} := T_n$ for all $n \geqslant 0$, which is consistent with the trigonometric definition of T_n.

Similarly to the monomial basis, we have simple formulas for multiplication and (indefinite) integration:

$$T_nT_m = \frac{1}{2}(T_{n+m} + T_{n-m}), \quad n, m \in \mathbb{Z},$$
$$\int T_n = \frac{1}{2}\left(\frac{T_{n+1}}{n+1} - \frac{T_{n-1}}{n-1}\right), \quad n \in \mathbb{Z}, \tag{2.25}$$

where $T_{n+1}/(n+1)$, resp. $T_{n-1}/(n-1)$, is 0 by convention when the denominator vanishes (that is, when $n = -1$, resp. $n = 1$). However, contrary to the monomial basis, derivation in the Chebyshev basis does not have a compact expression:

$$T_n' = n \sum_{\substack{|i|<|n| \\ i \neq n \bmod 2}} T_i = \begin{cases} n(T_{-n+1} + \cdots + T_{-1} + T_1 + \cdots + T_{n-1}), & n \text{ even}, \\ n(T_{-n+1} + \cdots + T_0 + \cdots + T_{n-1}), & n \text{ odd}. \end{cases} \tag{2.26}$$

Another important property is that Chebyshev polynomials form a family $(T_n)_{n\geqslant 0}$ of orthogonal polynomials with respect to the following inner product, defined on $L^2([-1, 1])$, the space of real-valued measurable functions over $[-1, 1]$ for which $\int_{-1}^{1} f(t)^2(1 - t^2)^{-1/2}\,\mathrm{d}t < +\infty$:

$$\langle f, g\rangle := \int_{-1}^{1} \frac{f(t)g(t)}{\sqrt{1-t^2}}\,\mathrm{d}t = \int_0^{\pi} f(\cos\theta)g(\cos\theta)\,\mathrm{d}\theta \in \mathbb{R}, \qquad f, g \in L^2.$$

One has: $\langle T_0, T_0\rangle = \pi, \quad \langle T_n, T_n\rangle = \frac{\pi}{2}$, for $n > 0$, and $\langle T_n, T_m\rangle = 0$, for $n \neq m$.

Whence, the n-th order Chebyshev coefficient of $f \in L^2$ is defined by:

$$[f]_n := \frac{1}{\pi}\langle f, T_n\rangle = \frac{1}{\pi}\int_0^{\pi} f(\cos\theta)\cos(n\theta)\,\mathrm{d}\theta, \quad n \in \mathbb{Z}. \tag{2.27}$$

Note that $[f]_{-n} = [f]_n$ for all $n \in \mathbb{Z}$ and the symmetric n-th order truncated Chebyshev series of f is defined by:

$$\mathrm{I\!I}_n \cdot f := \sum_{|i|\leqslant n} [f]_i T_i = [f]_{-n}T_{-n} + \cdots + [f]_0T_0 + \cdots + [f]_nT_n, \quad n \geqslant 0.$$

Beside convergence of $\mathrm{I\!I}_n \cdot f$ to f in $L^2([-1, 1])$ [49, Chap. 4], one also has the following result of uniform and absolute convergence [211, Thm. 3.1]:

Theorem 2.4.1. If f is Lipschitz continuous on $[-1, 1]$, it has a unique representation as a Chebyshev series,

$$f(x) = \sum_{k=-\infty}^{\infty} [f]_k T_k(x), \text{ with } [f]_{-k} = [f]_k \text{ for all } k \in \mathbb{Z},$$

which is absolutely and uniformly convergent.

This theorem shows the effectiveness of approximating by truncated Chebyshev series even when functions have low regularity. Moreover, the smoother f is, the faster its approximants converge. From [211, Thm 7.2], one has that if the νth derivative of f is of bounded variation V, then for a truncation order n, the speed of convergence is in $O(Vn^{-\nu})$. According to [211, Thm 8.2] for analytic functions, if $\rho > 0$ and f is analytic in the neighborhood of the set bounded by the Bernstein ρ-ellipse $\mathcal{E}_\rho = \{z = (\rho e^{i\theta} + \rho^{-1}e^{-i\theta})/2 \in \mathbb{C} \mid \theta \in [0, 2\pi]\}$ of foci -1 and 1, the convergence is in $O(M\rho^{-n})$, where M upper bounds $|f|$ on $\mathcal{E}_\rho$. In particular, for entire functions ($\rho = \infty$), the convergence is faster than any geometric sequence [33].

Note also that truncated Chebyshev series are near-best approximations with respect to the uniform norm, denoted by $\|\cdot\|_\infty$ in what follows, over $[-1, 1]$ on the space $\mathcal{C}^0 = \mathcal{C}^0([-1, 1])$ of continuous functions over $[-1, 1]$ [211, Thm. 16.1]:

Theorem 2.4.2. Let $n \in \mathbb{N}, n \geqslant 1$, $f \in \mathcal{C}^0$, and p_n^* denote the polynomial of degree at most n that minimizes $||f - p||_\infty$, then

$$||f - \pi_n \cdot f||_\infty \leqslant \left(4 + \frac{4}{\pi^2}\log(n+1)\right)\|f - p_n^*\|_\infty.$$

The Chebyshev Recurrence Relation From Equations (2.23) and (2.25) follows the important property that the Chebyshev coefficients of a D-finite function obey a linear recurrence with polynomial coefficients, a fact that was dealt with in a series of articles [75, 76, 169, 135, 85], and can be summarized in the framework of a suitable skew-polynomial ring (which were briefly defined in Section 2.1) as follows.

Denote by $\mathbb{Q}(n)\langle S, S^{-1}\rangle$ the skew Laurent polynomial ring over $\mathbb{Q}(n)$ in the indeterminate S, subject to the commutation rules

$$S\lambda = \lambda S \quad (\lambda \in \mathbb{Q}), \qquad Sn = (n+1)S. \tag{2.28}$$

Likewise, $\mathbb{Q}[n]\langle S, S^{-1}\rangle \subset \mathbb{Q}(n)\langle S, S^{-1}\rangle$ is the subring of noncommutative Laurent polynomials in S themselves with polynomial coefficients. The elements of $\mathbb{Q}[n]\langle S, S^{-1}\rangle$ identify naturally with linear recurrence operators through the left action of $\mathbb{Q}[n]\langle S, S^{-1}\rangle$ on sequences $(u_n)_{n\in\mathbb{Z}}$, defined by $(n \cdot u)_n = nu_n$ and $(S \cdot u)_n = u_{n+1}$. Recall that L denotes the differential operator appearing in Equation (2.1).

Theorem 2.4.3. [169, 134, 135, 183, 17] Let u, v be analytic functions on some complex neighborhood of the segment $[-1, 1]$, with Chebyshev expansions

$$u(x) = \sum_{n=-\infty}^{\infty} u_n T_n(x), \quad v(x) = \sum_{n=-\infty}^{\infty} v_n T_n(x).$$

There exist difference operators $P, Q \in \mathbb{Q}[n]\langle S, S^{-1}\rangle$ with the following properties.

1. The differential equation $L \cdot u(x) = v(x)$ holds if and only if
$$P \cdot (u_n) = Q \cdot (v_n). \tag{2.29}$$
2. The left-hand side operator P is of the form $P = \sum_{k=-s}^{s} b_k(n)S^k$ where $s = r + \max_i(\deg a_i)$ and $b_{-k}(-n) = -b_k(n)$ for all k.
3. Letting
$$\delta_r(n) = 2^r \prod_{i=-r+1}^{r-1}(n-i), \qquad I = \frac{1}{2n}(S^{-1} - S), \tag{2.30}$$
we have $Q = Q_r = \delta_r(n)I^r$ (this expression is to be interpreted as a polynomial identity in $\mathbb{Q}(n)\langle S, S^{-1}\rangle$). In particular, Q depends only on r and satisfies the same symmetry property as P.

A sloppy but perhaps more intuitive statement of the main point of Theorem 2.4.3 would be: "$(\int)^r L \cdot u = w$ if and only if $\delta_r(n)P \cdot u = w$, up to some integration constants".

While several properties of the Chebyshev recurrence (2.29), like the injectivity of Q restricted to symmetric sequences, the structure of recurrence singularities, as well as the asymptotic structure of the solutions of this recurrence are analyzed and proved in [J4], compared with power series, this recurrence does not yield a direct computation of the coefficients, due to several difficulties:

- Lack of initial conditions: unlike the first few Taylor coefficients of y, the Chebyshev coefficients that could serve as initial conditions for the recurrence are not related in any direct way to initial or boundary conditions of the differential equation. Moreover, the order $2s$ of the recurrence is usually larger than that of the differential equation, so we need to somehow "obtain more initial values for the recurrence than we naturally have at hand".
- Existence of leading and trailing singularities of the recurrence (2.29) (over which we have some control [J4, Prop. 2.7]).
- Chebyshev recurrences always admit *divergent* solution sequences, which do not correspond to the expansions of solutions of the differential equation the recurrence comes from. It can be proved, using the main asymptotic existence theorem for linear recurrences (Birkhoff-Trjitzinsky), that in general a Chebyshev recurrence (like in Theorem 2.4.3 (2)) has a basis of $2s$ "germs of solutions at infinity", half of them convergent and half divergent [J4, Corollary 3.6].

Due to the above mentioned difficulty, the numerical evaluation of Chebyshev series coefficients by (forward) recurrence unrolling is problematic, as exemplified below.

Example 2.4.4 (Numerical evaluation of forward recurrence for Chebyshev series coefficients. Part 1)**.** The Chebyshev recurrence associated to the equation $y' = y$ is

$$(P \cdot u)_n = u(n+1) + 2n\,u(n) - u(n-1) = 0.$$

In terms of the modified Bessel functions I_ν and K_ν, a basis of solutions of the recurrence is given by the sequences $(I_\nu(1))_{\nu\in\mathbb{Z}}$ and $(K_\nu(1))_{\nu\in\mathbb{Z}}$. The former is the coefficient sequence of the Chebyshev expansion of the exponential function and decreases as $\Theta(2^{-\nu}\,\nu!^{-1})$. The later satisfies $K_\nu(1) = \Theta(2^\nu\,(\nu-1)!)$. Hence, even if we had a way of obtaining very accurate initial conditions of the recurrence, forward unrolling would result in very unstable numerical computations, as practically shown in comparison with the Taylor counterpart, in Table 2.2.

Taylor series: $\exp(x) = \sum \frac{1}{n!}x^n$		Chebyshev series: $\exp(x) = \sum I_n(1)T_n(x)$	
Coeff. Recurrence: $u(n+1) = \frac{u(n)}{n+1}$		Coeff. Recurrence: $u(n+1) = -2nu(n) + u(n-1)$	
Rec. unrolling:	Accurate value:	Rec. unrolling:	Accurate value:
$u(0) = 1$	$1/0! = 1$	$u(0) = 1.266$	$I_0(1) \approx 1.266$
$u(1) = 1$	$1/1! = 1$	$u(1) = 0.565$	$I_1(1) \approx 0.565$
$u(2) = 0.5$	$1/2! = 0.5$	$u(2) \approx 0.136$	$I_2(1) \approx 0.136$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$u(50) \approx 3.28 \cdot 10^{-65}$	$1/50! \approx 3.28 \cdot 10^{-65}$	$u(50) \approx 4.450 \cdot 10^{67}$	$I_{50}(1) \approx 2.934 \cdot 10^{-80}$

Table 2.2 – Example of numerical computation (precision binary64) of power series coefficients (left) vs. Chebyshev series coefficients (right) for exp, by direct recurrence unrolling (first and 3rd column), while the accurate values are given in the second and 4th column.

To compute these solutions efficiently, despite the difficulties discussed above, a method originally due to Clenshaw [56][3] and already discussed in the Chebyshev series context in [75, 76, 183] can be

[3]The Clenshaw method we are referring to, should not be confused with the Horner-like scheme for Chebyshev polynomials known as Clenshaw's algorithm [55].

used. This is related to Miller's well-known *backward recurrence* technique [20, 226] to compute minimal ("slowest increasing") solutions of three-term recurrences. Miller's idea is to compute the coefficients $u_N, u_{N-1}, \dots, u_0$ of a linear recurrence sequence in the backward direction, starting form arbitrary "initial conditions" u_{N+1} and u_{N+2}. When N goes to infinity (u_{N+1}, u_{N+2} being chosen once and for all), the computed coefficients $u_0, \dots, u_N$ get close to those of a minimal solution with large u_0, u_1, in accordance with the intuition that "minimal solutions are the dominant ones when going backwards". This method behaves much better numerically than the standard forward recurrence. But its key feature for our purposes is that it allows to compute a minimal solution characterized by its minimality plus one normalizing condition instead of two initial values. This is exemplified below.

Example 2.4.5 (Numerical evaluation of forward recurrence for Chebyshev series coefficients. Part 2)**.** The Chebyshev recurrence

$$(P \cdot u)_n = u(n+1) + 2n\,u(n) - u(n-1) = 0$$

is unrolled backward: for instance, to compute an approximation of degree $N = 50$, canonical initial conditions are set to $u(52) = 0$ and $u(51) = 1$ and then the other coefficients are "backward" computed by $u(n-1) = u(n+1) + 2n\,u(n)$. The computation is shown in Table 2.3.

Backward unrolling:	Accurate value:	Backward unrolling and scaling:
$u(52) = 0$	$I_{52}(1) \approx 2.77 \cdot 10^{-84}$	$\frac{u(52)}{C} = 0$
$u(51) = 1$	$I_{51}(1) \approx 2.88 \cdot 10^{-82}$	$\frac{u(51)}{C} \approx -2.88 \cdot 10^{-82}$
$u(50) = -102$	$I_{50}(1) \approx 2.93 \cdot 10^{-80}$	$\frac{u(50)}{C} \approx 2.93 \cdot 10^{-80}$
$\vdots$	$\vdots$	$\vdots$
$u(2) \approx -4.72 \cdot 10^{80}$	$I_2(1) \approx 0.14$	$\frac{u(2)}{C} \approx 0.14$
$u(1) \approx 1.96 \cdot 10^{81}$	$I_1(1) \approx -0.57$	$\frac{u(1)}{C} \approx -0.57$
$u(0) \approx -4.4 \cdot 10^{81}$	$I_0(1) \approx 1.27$	$\frac{u(0)}{C} \approx 1.27$
$C = \sum_{n=-50}^{50} u(n)T_n(0) \approx -3.48 \cdot 10^{81}$		

Table 2.3 – Example of numerical computation (precision binary64) of Chebyshev series coefficients for $\exp$, with backward recurrence unrolling by setting canonical initial conditions (left) and then scaling the obtained results with the normalizing condition C, while the accurate values are given in the middle column.

Based on our study of the properties of the recurrence, we can turn Clenshaw's method into the true Algorithm 18 that applies uniformly to differential equations of arbitrary order and degree, which runs linearly in function of N. Roughly speaking, we use the idea of backward recurrence to approach the whole subspace of convergent solutions instead of a single minimal one. There remains to take care of the constraints related to the singularities of the recurrence, the symmetry condition and the initial values of the differential equation, all of which is done using linear algebra. Its correctness proof is more technical and can be found in [J4], here we highlight the main result.

Theorem 2.4.6. Algorithm 18 fails for finitely many N only. As $N \to \infty$, its output satisfies

$$\max_{n=-N}^{N} |y_n^{(N)} - y_n| = O(N^\tau e_{1,N}),$$

for some τ independent of N and $e_{1,N}$ corresponding to the formal asymptotic behavior of the *least convergent* solution of the recurrence i.e. $e_{1,N} = O(N!^\kappa \alpha^N e^{o(N)})$ (with $\kappa \leqslant 0$ and if $\kappa = 0$ then $|\alpha| < 1$).

Algorithm 18 Clenshaw-like computation of Chebyshev series coefficients for D-finite functions.

Input: a linear differential operator L of order r, and suitable boundary conditions $\lambda_1(y) = \ell_1, \ldots, \lambda_r(y) = \ell_r$, a target degree $d > s$, an integer $N \geqslant \max(d, \max\{n : b_{-s}(n) = 0\})$.
Output: an approximation $\tilde{y}(x) = \sum_{n=-d}^{d} \tilde{y}_n T_n(x)$ of the corresponding solution y of $L \cdot y = 0$.

1 compute the Chebyshev recurrence operator $P = \sum_{k=-s}^{s} b_k(n) S^k$ associated to L
2 set $\mathbf{S} = \{n \geqslant s : b_{-s}(n) = 0\}$ and $\mathbf{I} = \mathbf{S} \cup [\![N, N+s-1]\!]$
3 for n from $N+s-1$ down to 1
  4 for $i \in \mathbf{I}$
    5 if $n = i$ then set $f_{i,n-s} = 1$
    6 else if $n \in \mathbf{I}$ then set $f_{i,n-s} = 0$
    7 else compute $f_{i,n-s}$ using the relation $(P \cdot f)_n = 0$
8 using indeterminate $(\eta_i)_{i\in\mathbf{I}}$, set

$$\tilde{y}_n = \begin{cases} \sum_{i\in\mathbf{I}} \eta_i f_{i,|n|}, & |n| \leqslant N \\ \tilde{y}_n = 0, & |n| > N, \end{cases} \quad \text{and} \quad \tilde{y}(x) = \sum_{n=-N}^{N} \tilde{y}_n T_n(x)$$

9 solve for $(\eta_i)_{i\in\mathbf{I}}$ the linear system

$$\begin{cases} \lambda_k(\tilde{y}) = \ell_k, & 1 \leqslant k \leqslant r, \\ b_{-s}(n)\tilde{y}_{n-s} + \cdots + b_s(n)\tilde{y}_{n+s} = 0, & n \in [\![r, s-1]\!] \cup \mathbf{S} \end{cases} \tag{2.31}$$

10 return $\sum_{n=-d}^{d} \tilde{y}_n T_n(x)$

More numerical examples for Algorithm 18 can be found in [J4]. We chose to show the quality of these approximations by the following "effective near-minimax approximation" property [J4]:

Corollary 2.4.7. Given $d \in \mathbb{N}$, there exists N such that Algorithm 18, called with parameters L, (ℓ_k), d, and N, computes a polynomial p_d of degree at most d satisfying $\|p_d - y\|_\infty \leqslant \left(4\pi^{-2}\ln(d+1)+5\right)\|p_d^* - y\|_\infty$ in $O(\ln \|p_d - y\|_\infty^{-1})$ arithmetic operations.

Concerning the computation of an *a posteriori* validated approximation error bound, we proposed in [J4] an algorithm based on convergent Neumann series of linear operators in the Banach space of continuous functions $(\mathcal{C}^0, \|\cdot\|_\infty)$. We chose here to focus on a more recent work [J2], in collaboration with my PhD student, F. Bréhard, and N. Brisebarre (co-supervisor). It extends the complexity study of [J4] to the framework of quasi-Newton validation methods for LODEs.

2.4.2 Validated and efficient Chebyshev spectral methods for linear ordinary differential equations

More specifically, consider the following problem:

Problem 3. Let r be a positive integer, $\alpha_0, \alpha_1, \ldots, \alpha_{r-1}$ and γ continuous functions over $[-1, 1]$. Consider the LODE

$$f^{(r)}(t) + \alpha_{r-1}(t)f^{(r-1)}(t) + \cdots + \alpha_1(t)f'(t) + \alpha_0(t)f(t) = \gamma(t),\ t \in [-1, 1], \tag{2.32}$$

together with conditions uniquely characterizing the solution:

a) For an *initial value problem* (IVP), consider:

$$\mathbf{\Lambda} \cdot f := (f(t_0), f'(t_0), \ldots, f^{(r-1)}(t_0)) = (v_0, v_1, \ldots, v_{r-1}), \tag{32a}$$

for given $t_0 \in [-1, 1]$ and $(v_0, v_1, \ldots, v_{r-1}) \in \mathbb{R}^r$.

b) For a generalized *boundary value problem* (BVP), conditions are given by r linearly independent linear functionals $\lambda_i : \mathcal{C}^0 \to \mathbb{R}$:

$$\mathbf{\Lambda} \cdot f := (\lambda_0(f), \ldots, \lambda_{r-1}(f)) = (\ell_0, \ldots, \ell_{r-1}), \tag{32b}$$

for given $(\ell_0, \ldots, \ell_{r-1}) \in \mathbb{R}^r$.

Given an approximation degree $d \in \mathbb{N}$, find the coefficients of a polynomial $\tilde{f}(t) = \sum_{n=0}^{d} c_n T_n(t)$ written in Chebyshev basis (T_n), together with a tight and rigorous error bound η such that $\|f - \tilde{f}\|_\infty := \sup_{t\in[-1,1]} |f(t) - \tilde{f}(t)| \leqslant \eta$.

One can readily see that the previously proposed algorithm provides a solution for the numerical part, given *sufficiently good* polynomial approximations of the coefficients α_i. But other numerical methods could also be used (see for instance [102, 165]). Therefore, we also proposed another linear (with respect to the approximation degree) time approximation algorithm in [J2] which is based on an algorithm for almost-banded linear systems from [165], together with a classical integral reformulation of the above problem.

We focus now on the rigorous computation of approximation error bounds. Our proposed solution is less based on computer algebra algorithms for D-finite functions (although some structural properties of such functions remain essential) and more oriented towards a functional analysis approach. In particular, a so-called *a posteriori* quasi-Newton validation method is employed, which mainly relies on a fixed-point argument of a contracting map. This method was already used for nonlinear multivariate problems [229, 132, 217, 98], but these works focused on *ad-hoc* solutions for specific problems. In contrast, our study is from the computer algebra perspective: only LODEs are handled, yet with a generic algorithmic approach, as well as its complexity study.

General setting for quasi-Newton validation

Consider the equation $\mathbf{F} \cdot x = 0$ where $\mathbf{F}$ is an operator acting on a Banach space $(E, \|\cdot\|)$. A numerical method provides an approximation $\widetilde{x}$ of some exact solution x. One is interested in rigorously bounding the approximation error between x and $\widetilde{x}$. For that, a classical idea is to reformulate the problem as a fixed-point equation $\mathbf{T} \cdot x = x$ with $\mathbf{T} : E \to E$ an operator whose fixed points correspond to the zeros of $\mathbf{F}$. The distance between a given approximation and a fixed point of $\mathbf{T}$ is bounded based on the following theorem [18, Thm 2.1]:

Theorem 2.4.8. Let $(E, \|\cdot\|)$ be a Banach space, $\mathbf{T} : E \to E$ a continuous operator and $\widetilde{x} \in E$ an approximate solution of the fixed-point equation $\mathbf{T} \cdot x = x$. If there is a radius $r > 0$ such that

- $\mathbf{T} \cdot \overline{B}(\widetilde{x}, r) := \{\mathbf{T} \cdot x \mid \|x - \widetilde{x}\| \leqslant r\} \subseteq \overline{B}(\widetilde{x}, r) := \{x \mid \|x - \widetilde{x}\| \leqslant r\}$, and
- $\mathbf{T}$ is contracting over $\overline{B}(\widetilde{x}, r)$: there exists a constant $\mu \in (0, 1)$ such that for all $x_1, x_2 \in \overline{B}(\widetilde{x}, r)$, $\|\mathbf{T} \cdot x_1 - \mathbf{T} \cdot x_2\| \leqslant \mu \|x_1 - x_2\|$,

then $\mathbf{T}$ admits a unique fixed point x^* in $\overline{B}(\widetilde{x}, r)$ and we have the following enclosure of the approximation error:

$$\frac{\|\mathbf{T} \cdot \widetilde{x} - \widetilde{x}\|}{1 + \mu} \leqslant \|x^* - \widetilde{x}\| \leqslant \frac{\|\mathbf{T} \cdot \widetilde{x} - \widetilde{x}\|}{1 - \mu}.$$

One special class of such operators $\mathbf{T}$ are the Newton-like operators acting on Banach spaces (see [180, Chap.4] and references therein). Suppose that $\mathbf{F}$ is of class $\mathcal{C}^2$ over E, and suppose that $\mathbf{A} = (\mathrm{d}\mathbf{F}|_{x=\widetilde{x}})^{-1}$ exists. Then the fixed points of:

$$\mathbf{T} = \mathbf{I} - \mathbf{A} \cdot \mathbf{F} : E \to E \tag{2.33}$$

are exactly the zeros of $\mathbf{F}$ and $\mathbf{T}$ has a null derivative at $\widetilde{x}$, so that it is locally contracting around $\widetilde{x}$. Hence, if for a well-chosen $r > 0$, the hypotheses of Theorem 2.4.8 are respected, one obtains an upper bound for the approximation error $\|x^* - \widetilde{x}\|$. In general however, we cannot exactly compute $(\mathrm{d}\mathbf{F}|_{x=\widetilde{x}})^{-1}$ and $\mathbf{A}$ is only an approximation. Still, this may be sufficient to get a contracting operator $\mathbf{T}$ around $\widetilde{x}$.

When $\mathbf{T}$ is affine, it is contracting if and only if its linear part $\mathcal{D}\mathbf{T}$ has operator norm $\|\mathcal{D}\mathbf{T}\| = \mu < 1$. In particular, for an affine operator $\mathbf{T}$, being locally or globally contracting are equivalent. Therefore, the ball $\overline{B}(\widetilde{x}, r)$ can be replaced by the whole space E in Theorem 2.4.8 and the first condition becomes trivially true.

This general abstract formulation was specialized in [J2] as follows.

Setting for quasi-Newton validation of Chebyshev series

- The Banach space denoted by $(\text{Ч}^1, \|\cdot\|_{\text{Ч}^1})$, contains continuous functions f, for which the sum of absolute value of Chebyshev series coefficients is convergent:

$$\|f\|_{\text{Ч}^1} := \sum_{n\in\mathbb{Z}} |[f]_n| < +\infty.$$

  Note that if $f \in \text{Ч}^1$, then $\text{п}_n \cdot f$ converges absolutely and uniformly to f [J2, Lemma 2.3.]. Moreover, Ч^1 is a Banach algebra analogous to the Wiener algebra $A(\mathbb{T})$ of absolutely convergent Fourier series [109, §I.6]: for $f \in \text{Ч}^1$, we have $\|f\|_{\text{Ч}^1} = \|f(\cos)\|_{A(\mathbb{T})}$.

- The operators in Equation (2.33) are obtained by transforming the differential Equation (2.32) into an integral one. This is useful because derivation is not an endomorphism of Ч^1 (some functions in Ч^1 are *not* even differentiable) and also the action of derivation on Chebyshev series has worse numerical conditioning properties [87].

  Proposition 2.4.9. Let f be a function of class $\mathcal{C}^r$ over $[-1, 1]$. Then f is a solution of the linear IVP problem (32a) if and only if $\varphi = f^{(r)} \in \mathcal{C}^0$ is solution of the Volterra integral equation:

$$\varphi + \mathbf{K}\cdot\varphi = \psi \quad \text{with} \quad (\mathbf{K}\cdot\varphi)(t) = \int_{t_0}^{t} k(t,s)\varphi(s)\mathrm{d}s,\ t \in [-1,1], \tag{2.34}$$

  where:

  - the kernel $k(t, s)$ is a bivariate continuous function given by:

$$k(t,s) = \sum_{j=0}^{r-1} \alpha_j(t)\frac{(t-s)^{r-1-j}}{(r-1-j)!},\ (t,s) \in [-1,1]^2, \tag{2.35}$$

  - the right-hand side ψ is given by:

$$\psi(t) = \gamma(t) - \sum_{j=0}^{r-1} \alpha_j(t) \sum_{k=0}^{r-1-j} v_{j+k}\frac{(t-t_0)^k}{k!},\ t \in [-1,1]. \tag{2.36}$$

  By a slight abuse of terminology, we shall call r the order of the integral operator $\mathbf{K}$.

- In this case, $\mathbf{F}\cdot\varphi := \varphi + \mathbf{K}\cdot\varphi - \psi$ is affine, with linear part $\mathbf{I} + \mathbf{K}$. The quasi-Newton method requires an approximate inverse operator $\mathbf{A} \approx (\mathbf{I} + \mathbf{K})^{-1}$ such that $\|\mathbf{I} - \mathbf{A}\cdot(\mathbf{I} + \mathbf{K})\|_{\text{Ч}^1} < 1$. Of course, computing an exact inverse would solve the problem but is out of reach. Instead of that, one can proceed from the following two key ideas:

  - it is necessary to consider finite dimensional approximations of these operators and to establish convergence and invertibility properties of these operators and their truncations with respect to their action on Ч^1, together with effective truncation bounds;
  - it turns out that in many cases, these finite dimensional truncations have a very interesting sparse matrix structure, namely they are almost-banded, which allows for very efficient numerical algorithms [165].

Specifically, one can prove that $\mathbf{K}$ and $\mathbf{I}+\mathbf{K}$ are bounded linear endomorphisms of $\mathcal{H}^1$, $\mathbf{I}+\mathbf{K}$ is invertible in $\mathcal{H}^1$. Moreover, the n-th truncation (also called the n-th section in [86]) of the integral operator $\mathbf{K}$:

$$\mathbf{K}^{[n]} = \pi_n \cdot \mathbf{K} \cdot \pi_n, \tag{2.37}$$

converges to $\mathbf{K}$ for the $\|\cdot\|_{\mathcal{H}^1}$ operator norm, as $n \to \infty$ and $\mathbf{K}$ is compact.

Validation for the D-finite case

When the LODE (2.32) is D-finite, the representation matrices (obtained by studying the operator action on Chebyshev polynomials) of $\mathbf{K}$ and $(\mathbf{I}+\mathbf{K})$ (as well as their truncations $\mathbf{K}^{[n]}$ and $(\mathbf{I}+\mathbf{K})^{[n]}$) have an almost-banded structure (see Figure 2.4).

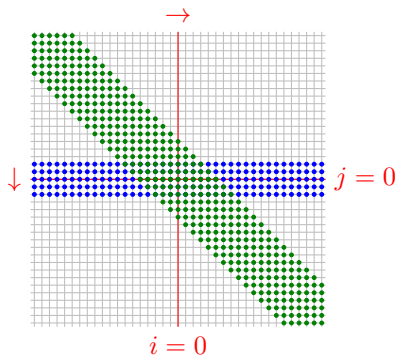


Figure 2.4 – Almost-banded structure of operator $\mathbf{K}$.

One can prove that for n large enough $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$ exists and is a good approximation of $(\mathbf{I}+\mathbf{K})^{-1}$. Since $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$ is defined by an $(n+1)$-order square matrix (its restriction over $\pi_n \cdot \mathcal{H}^1$) extended over the whole space $\mathcal{H}^1$ by the identity, we define the operator $\mathbf{A}$ over $\mathcal{H}^1$ as an $(n+1)$-order square matrix A approximating $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$ over $\pi_n \cdot \mathcal{H}^1$, extended by the identity over the whole space:

$$\mathbf{A} \cdot \varphi = A \cdot \pi_n \cdot \varphi + (\mathbf{I} - \pi_n) \cdot \varphi.$$

The first technical issue is to numerically compute (or represent) both very accurately and efficiently such a matrix A. Specifically, we aim both for a linear complexity computation with respect to n and for minimizing $\|I_{n+1} - A \cdot M\|_1$, where M is an order $n+1$ matrix representation for $\mathbf{I}+\mathbf{K}^{[n]}$. In [J2] we present two solutions: one that has a quadratic complexity w.r.t. n, but computes M^{-1} very accurately; the other uses a heuristic which states that $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$ is well approximated by almost-banded matrices. This second option may require a large value of n, but works well in practice and has a linear complexity w.r.t. n. A discussion on which of these two methods should be used in practice is presented in [J2], together with complexity estimates in both cases.

Next, one has to provide a rigorous Lipschitz constant μ (required by Theorem 2.4.8) for the Newton-like operator. We have:

$$\|\mathbf{I} - \mathbf{A} \cdot (\mathbf{I}+\mathbf{K})\|_{\mathcal{H}^1} \leqslant \|\mathbf{I} - \mathbf{A} \cdot (\mathbf{I}+\mathbf{K}^{[n]})\|_{\mathcal{H}^1} + \|\mathbf{A} \cdot (\mathbf{K} - \mathbf{K}^{[n]})\|_{\mathcal{H}^1}, \tag{2.38}$$

which can be interpreted as:

- $\|\mathbf{I} - \mathbf{A} \cdot (\mathbf{I}+\mathbf{K}^{[n]})\|_{\mathcal{H}^1}$ is the approximation error because A was (maybe) not the exact representation matrix of $(\mathbf{I}+\mathbf{K}^{[n]})^{-1}$.
- $\|\mathbf{A} \cdot (\mathbf{K} - \mathbf{K}^{[n]})\|_{\mathcal{H}^1}$ is the truncation error because $\mathbf{K}^{[n]}$ is not exactly $\mathbf{K}$.

Bounding the approximation error reduces to (almost-banded) matrix multiplication with interval arithmetic, while the truncation error concerns the *tail* of the operator $\mathbf{K}$ and needs some more technical inequalities (cf. [J2, Algorithm 6.]). To summarize, one can prove (and compute explicit upper-bounds) that the error due to *tail* diagonal coefficients $B_D(i) = \|(\mathbf{I}-\pi_n)\cdot\mathbf{K}\cdot T_i\|_{\mathbb{Q}^1}$ decreases in $O(1/i)$. Furthermore, the error due to initial coefficients multiplied by A, $B_I(i) = \|A\cdot\pi_n\cdot\mathbf{K}\cdot T_i\|_{\mathbb{Q}^1}$ decreases in $O(1/i^2)$.

Once we have obtained and validated a quasi-Newton operator $\mathbf{T}$ with a certified Lipschitz constant $\mu < 1$, the validation of a candidate solution $\widetilde{\varphi}$ of the integral Equation (2.34), reduces to computing:

$$\|\varphi^* - \widetilde{\varphi}\|_{\mathbb{Q}^1} \leqslant \frac{\|\widetilde{\varphi} + \mathbf{K}\cdot\widetilde{\varphi} - \psi\|_{\mathbb{Q}^1}}{1-\mu},$$

where all the computations are performed in interval arithmetics.

Extension to non D-finite case

The extension to the non D-finite case is based on approximating the coefficients of the differential operators by Chebyshev models, and computing an integral kernel operator by overloading operations with Chebyshev models. The resulting polynomial part defines a polynomial kernel $k_P(t,s)$ and respectively the polynomial integral operator $\mathbf{K}_P$, such that $\|\mathbf{K}-\mathbf{K}_P\|_{\mathbb{Q}^1}$ is explicitly bounded by a constant depending only on r and the initial error bounds in the LODE coefficients approximation. This justifies the fact that $\mathbf{K}$ is well-approximated by $\mathbf{K}_P$ when the coefficients of Equation (2.32) are well approximated by polynomials. Similar computations follow, by noting that the linear part of $\mathbf{T}$ can be decomposed into three parts:

$$\|\mathbf{I}-\mathbf{A}\cdot(\mathbf{I}+\mathbf{K})\|_{\mathbb{Q}^1} \leqslant \|\mathbf{I}-\mathbf{A}\cdot(\mathbf{I}+\mathbf{K}_P^{[n]})\|_{\mathbb{Q}^1} + \|\mathbf{A}\cdot(\mathbf{K}_P-\mathbf{K}_P^{[n]})\|_{\mathbb{Q}^1} + \|\mathbf{A}\cdot(\mathbf{K}-\mathbf{K}_P)\|_{\mathbb{Q}^1}.$$

The first two parts are exactly the ones of (2.38) (where the polynomial integral operator $\mathbf{K}$ is now called $\mathbf{K}_P$) and can be rigorously upper bounded using the same techniques. The last part can be upper bounded thanks to the initial computations with Chebyshev models. It is interesting to notice that the order of magnitude of n is largely determined by the second part (as in the polynomial case), whereas the third part mainly depends on the degree of the approximating polynomials for $\alpha_j(t)$.

In conclusion, we observe that our validation method is easily adapted to the general case where the coefficients α_j are non-polynomial functions rigorously approximated by polynomials. However, contrary to the polynomial case where the involved degrees are usually low, the degrees of the approximants can be rather large, resulting in a dense linear problem and poorer time efficiency. Yet, in practice, the method remains efficient on problems with reasonable coefficient magnitude and time interval under consideration. The case of other boundary conditions is treated by reducing a general BVP validation problem to $r+1$ IVP validation problems (see [J2] for more details). Our new approach is to be illustrated also for validating solutions of LODEs appearing in robust guidance algorithms for space trajectories in Chapter 3.

We conclude by providing a brief description of the complexity result obtained, whose exact formulation is rather technical and is omitted in this report.

Complexity estimates

Firstly, notice that a numerical approximation of degree d can be obtained in $O(d)$ arithmetic operations either with Algorithm 18 or by the one proposed [165]. Concerning the validation, the relevant parameters involved are the degree of the numerical approximation d and the truncation order n (of the operator $\mathbf{K}$) needed to obtain a certified Lipschitz constant $\mu < 1$. In function of these two parameters, the complexity is $O(d+n)$, hence when n is small with respect to d, the validation is linear in d also. However, we were not able to prove that this is always the case and the theoretical bound we have is $n = O(dB^2\exp(2B))$, where B quantifies the norm of the coefficients of the integral operator (2.34). However this exponential bound is very pessimistic in practice, and hence our validation algorithm proceeds by setting a truncation order function of a numerical estimate of the norm $\|(\mathbf{I}+\mathbf{K}^{[n]})^{-1}\cdot(\mathbf{K}-\mathbf{K}^{[n]})\|_{\mathbb{Q}^1}$. This is done by numerically applying this operator on T_{n+1}, which is a heuristic similar to estimating the truncation

error of a Chebyshev series by its first neglected term [49, Chap.4.4, Thm. 6]. If the estimated value is not correct, one proceeds by doubling n, until $\mu < 1$.

2.5 Conclusion

In this chapter, we proposed several symbolic-numeric algorithms, which combine computer algebra, polynomial optimization and validated computations. We based our developments on the algebraic structure and algorithmic properties of D-finite (holonomic) functions. We also exploit other (more functional analysis-like) properties of differential operators like compactness, convergence, invertibility in suitable Banach spaces. Moreover, we focus on the "sparsity" generated by these operators and how this can be efficiently put to use in our algorithms. These core properties are applied in several (at first sight unconnected) areas. Finally, our approach is based both on providing good theoretical complexity estimates, but also practical code which actually works in applications as it will be further discussed in the next chapter. Possible extensions of these works are discussed in Chapter 4.

Chapter 3

Validated Computations for Aerospace

Since my CNRS recruitment in January 2013, one of my general research goals was to bring more reliable computations in the field of optimal control theory and aerospace. This was possible thanks to my collaboration with Denis Arzelier (CNRS Researcher, LAAS). Based on his previous works and research and transfer projects, with CNES or ADS for instance, I became aware of and then contributed to two main space-related themes. One concerns the collision risk assessment for orbital objects [J5, C12, C13, R2], the other is related to orbital rendezvous and proximity operations [C8, C3, C4, C7, J1]. These works also benefited from the collaboration with other researchers from LAAS, namely J.-B. Lasserre (CNRS Researcher, LAAS), C. Louembet (Maître de conférences, Université Toulouse 3) and A. Rondepierre (Maître de conférences, INSA Toulouse), as well as F. Camps (research engineer at LAAS) and also, the following students. Firstly, R. Serra defended his PhD in December 2015, under the supervision of D. Arzelier and A. Rondepierre; I contributed to his supervision in the second part of this PhD (2013-2015). Secondly, P.R. Arantes-Gilz, co-supervised with C. Louembet, defended his PhD thesis in October 2018. Thirdly, F. Bréhard is to defend in July 2019, co-supervised together with N. Brisebarre and D. Pous (CNRS Researchers at LIP, Lyon). Last but not least, N. Deak did an undergraduate internship in 2015 under my supervision. Most of the above results were already presented in detail in the Phd thesis of R. Serra [197], or that of P.R. Arantes-Gilz [7]. The goal of this chapter is to focus two contributions, which particularly illustrate the close blend and interaction between the techniques presented in Chapter 2 with optimal control and space-related applications, as follows.

1. (Collision probability [J5, C12, C13]). We proposed a new accurate, reliable and efficient method to compute the orbital collision probability between two spherical objects involved in a short-term encounter, under Gaussian uncertainty. In this model of conjunction, the probability of collision is reduced to the integral of a 2D Gaussian probability density function over a disk.

   This computation needs to be particularly reliable and fast. In brief, this stems from the fact that the number of space objects in Low Earth Orbits, with diameter above 10 cm, a large majority being space debris, has dramatically increased during the last fifteen years. These orbiting debris constitute a serious hazard for operational satellites. Accordingly, for the overall safety of flight, it is critical to provide adequate mitigation and avoidance strategies when a conjunction includes at least one active satellite. Usually, the relative debris–satellite positions and velocities are only approximately known, hence the risk of an on-orbit collision is modeled as a collision probability. If it is evaluated to be sufficiently high, a collision avoidance maneuver is decided, but each such maneuver reduces the remaining satellite fuel and thus its active in-orbit life. Yet, a wrong computation which underestimates the risk, could result in the satellite loss. This implies that both efficient and accurate algorithms are needed. In Section 3.1, a more detailed description and modeling of this problem is presented, which boils down to the direct use of Algorithm 13 given in Chapter 2.

   This work had a particular practical impact, since after having independently implemented and run it on test databases (millions of cases), CNES uses it now as their reference method. This also

resulted in a long term collaboration with our CNRS partners S. Laurens and J.C. Dolado, as part of, for instance, the research and transfer project *Global Collision Probability and Satellites Station Keeping (2016-2018)*, which I coordinated.

2. (Validated impulsive spacecraft rendezvous [C8, J2]). The rendezvous (RdV) problem is a process which meets two spacecraft, originally moving on different orbits, in order to match their positions and velocities. A rather general case of RdV can be a spacecraft (referred to as chaser) targeting an object (referred to as target, e.g. International Space Station) on its orbit. Since the '60s, many ideas have been developed, and today, we are interested in successful RdV which minimizes fuel consumption, with increased autonomy (no human operator). This implies that validation of computations and solutions is at stake. Impulsive RdV problems concern in practice a large number of satellites, which are equipped with ergol thrusters. The impulsive approximation for the thrust means that an instantaneous velocity increment is applied to the chaser for each impulse.

   In what follows, we focus on the fixed-time minimum-fuel rendezvous between close elliptic orbits of an active spacecraft with a passive target spacecraft, assuming a linear impulsive setting and a Keplerian relative motion. Firstly, closely following the developments in [C8], we present the original optimal control problem, as well as its solution, given by the locations and magnitude of the impulses, and obtained by a numerical iterative algorithm. An interesting contribution is that this algorithm comes from the formulation of a semi-infinite convex optimization problem, using a relaxation scheme and duality theory in normed linear spaces. Secondly, we propose an *a posteriori* validation for the obtained solution, via Rigorous Polynomial Approximations, which were already discussed in Section 2.4.

   The algorithm proposed to solve this optimal control problem was also extended in different settings. In [C3] different linearized dynamics corresponding to the circular restricted 3-body-problem are considered. Furthermore, in [C1] an *exchange algorithm* [222, 45, 41, 40] which solves an approximation problem related to efficient machine implementation of mathematical functions is revisited, explained and extended based on the duality theory. Finally, note that *a posteriori* validation techniques were also extended by my students P.R. Arantes-Gilz and F. Bréhard, together with C. Gazzino (PhD student at LAAS) for other linearized dynamics by providing semi-analytical transition matrices via Chebyshev polynomials [8].

3.1 Collision probability

Since the collision between the Russian satellite COSMOS 1934 and one debris of COSMOS 926 in December 1991, no less than eight orbital collisions have been reported between operational satellites (e.g. IRIDIUM 33 and COSMOS 2251 collision in February 2009, the 10th), or between satellites and debris (e.g. the French satellite CERISE hit by a debris in July 1996, the 24th or the collision between THOR BURNER and a debris of Long March in January 2005, the 17th). Collision risk is particularly high in low orbits and different space agencies (CNES, ESA, NASA) and operators (Airbus Defense and Space (ADS), GMV) have established alert procedures to assess the risks of collision for controlled satellites, and to authorize avoidance maneuvers if the predicted risk exceeds some tolerance threshold. These procedures have undergone many changes in recent years and the field of collision avoidance techniques is currently in full development. For the different evolutions of these procedures, according to different agencies and operators, we refer to [70], [69] for ESA, [121, 120] for CNES and [21] for ADS.

At the origin of any procedure of collision avoidance between two orbital objects, whether controlled or not, lies the information of conjunction between the two objects. Since 2009, a Conjunction Message is sent by the Joint Space Operations Center (JSpOC) to all spacecraft owners and operators, concerning approximately 15000 objects listed in the Two-Line Elements (TLE) catalog provided by USSTRATCOM (US Strategic Command). The information provided by the JSpOC consists of a Conjunction Assessment Report (CAR) containing few information: the Time of Closest Approach (TCA), the miss distance between the two objects, statistical and geometrical information on the position and the velocities of each object. These messages are sent only three days before the date of the encounter. To obtain more accurate information on the possible encounter, it is necessary to subscribe to a service which will in

return provide a Conjunction Summary Report (CSM) from which is extracted the information needed to calculate the risk of collision between both objects. This collision risk assessment evaluates the risk for individual encounters.

The most general methods to accurately compute the global collision probability, without any additional assumption, are based on Monte-Carlo simulations, see e.g. [6, 64] in the context of a simple encounter or [82] in the context of a multiple encounter. These methods use a random sampling of N vectors in the space of initial conditions. For each of them, the corresponding trajectories are propagated according to the dynamical model adopted on the discretized time interval $[0, T]$. We count 1 if there is a collision, 0 otherwise. At the end, the collision probability is given by the formula: $\mathcal{P}_c([0,T]) = \frac{1}{N}\sum_{i=1}^{N} \delta_i$. The number of trials to be made depends on the requested precision as well as the value of the probability: a low value requires a lot of samples to be correctly estimated, and simulations can be dramatically time-consuming. This is one of the major disadvantages of Monte Carlo methods which makes them unsuitable for detecting low probability events in high dimension such as multiple events [82]. Therefore alternative approaches had to be explored to assess the risk of collision between two or more objects.

In the particular context of encounters between two objects, encounters are usually classified into two families: the short-term encounters [2, 46, 58, 72] and the long-term encounters [46, 64]. In the context of short-term encounters, conjunctions are assumed to be short and rare and several simplifying assumptions, enabling a more efficient computation of the collision probability can be made. The relative velocity between the two objects is assumed to be very high (several km/s) and the relative motion is assumed rectilinear on the time interval of the encounter. Finally, it is also assumed that the cross-correlations between the estimated states of the two objects are very small and therefore negligible. Such encounters typically occur in low orbits where the orbital velocities are high. Long-term encounters are characterized by relative velocities of the order of m/s, and correspond to situations where both objects spend significant time in proximity to each other. The motion equations of both objects are linearized around the reference orbit. This type of encounter is more common in the context of formation flying or proximity operations. Investigations about the range of validity of the short-term encounter model can be found in [48, 57]. We focus in what follows on the case of a short-term encounter between spherical objects under Gaussian-distributed uncertainty.

3.1.1 Short-term encounter

Let us recall the five assumptions needed to define the short-term encounter model under Gaussian-distributed uncertainty for spherical objects:

1. The relative trajectories are approximated as rectilinear.
2. The velocities are considered as deterministic variables.
3. Initial position vectors of both objects are Gaussian independent random vectors.
4. Each object is approximated by a spherical geometrical shape.
5. The time boundaries of the conjunction are extended to infinity.

The fact that the relative motion is rectilinear motivates the choice of a frame of study with one axis along the relative velocity. One possibility is to introduce the so-called encounter frame [71, 2, 171, 48] defined at reference time.

Encounter frame

This frame is centered on the mean position of one of the two objects and is built from the so-called encounter plane. This plane contains the origin and is orthogonal to the direction of the relative velocity. The configuration considered in this study is represented in Figure 3.1. The origin of the frame is located at the center of the primary object p. The basis vector $e_{\tilde{z}}$ is oriented along the relative velocity $v = v_s - v_p$. The basis vector $e_{\tilde{x}}$ belongs to the encounter plane: it points towards the orthogonal projection of

the mean relative position $\mu(r^0)$ onto the encounter plane. Finally, the basis vector $e_{\tilde{y}}$ completes the right-handed system and thus belongs to the encounter plane as well. In summary, one has

$$e_{\tilde{z}} = \frac{v}{\|v\|}, \quad e_{\tilde{y}} = \frac{v \times \mu(r^0)}{\|v \times \mu(r^0)\|}, \quad e_{\tilde{x}} = e_{\tilde{y}} \times e_{\tilde{z}}. \tag{3.1}$$

Let $(\tilde{x}_m, 0, \tilde{z}_m)$ be the coordinates of the mean relative position in the encounter frame where the $\tilde{y}$ coordinate is zero by construction. It is worth noticing that, since the relative trajectory is rectilinear, $\tilde{x}_m$ is in fact equal to the miss distance.

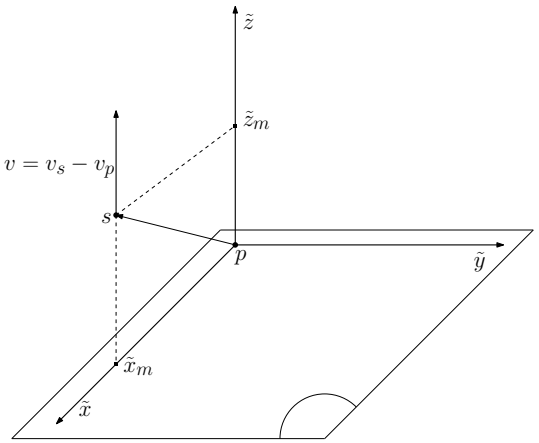


Figure 3.1 – Encounter plane and frame $(e_{\tilde{x}}, e_{\tilde{y}}, e_{\tilde{z}})$.

Integral representation

Under the above assumptions, the probability of collision can be formulated as a 2-D integral in the encounter plane. For spherical objects, the domain of integration is a closed disk $\bar{\mathcal{B}}((0,0), R)$ centered at the origin of radius R. The quantity R is the combined radius and is defined as the sum of the respective radii of the two objects i.e. $R = R_p + R_s$.

The two-dimensional probability distribution involved in the probability of collision describes the distribution of the relative position in the encounter plane. From the hypothesis on the nature of uncertainty, it is a multivariate normal law. Therefore, it is completely defined by its mean vector and its covariance matrix [167]. Let $\Sigma_{\tilde{x}\tilde{y}}$ be the covariance matrix of the relative coordinates in the encounter plane. The probability of collision can then be written as:

$$\mathcal{P}_c = \frac{1}{2\pi\sqrt{|\Sigma_{\tilde{x}\tilde{y}}|}} \int\limits_{\bar{\mathcal{B}}((0,0),R)} \exp\left(-\frac{1}{2}\begin{bmatrix}\tilde{x} - \tilde{x}_m & \tilde{y}\end{bmatrix} \Sigma_{\tilde{x}\tilde{y}}^{-1} \begin{bmatrix}\tilde{x} - \tilde{x}_m & \tilde{y}\end{bmatrix}^T\right) \mathrm{d}\tilde{x}\mathrm{d}\tilde{y}. \tag{3.2}$$

Equation (3.2) shows that the probability of collision only depends on the combined radius R, the miss distance $\tilde{x}_m$ and the covariance matrix $\Sigma_{\tilde{x}\tilde{y}}$ of the relative coordinates in the encounter plane.

Frame rotation In order to eliminate the cross-terms of the Gaussian function, a rotation of angle $-\theta$ to the principal axis of the covariance matrix is performed in the encounter plane (see Figure 3.2). This transformation does not change the nature of the domain of integration which remains a disk of radius R centered at the origin. The new coordinates, denoted (x, y), are respectively along the major and the minor axis. This transformation allows to write a formula with a simpler integrand:

$$\mathcal{P}_c = \frac{1}{2\pi\sigma_x\sigma_y} \int\limits_{\bar{\mathcal{B}}((0,0),R)} \exp\left(-\frac{1}{2}\left(\frac{(x - x_m)^2}{\sigma_x^2} + \frac{(y - y_m)^2}{\sigma_y^2}\right)\right) \mathrm{d}x\mathrm{d}y, \tag{3.3}$$

where the quantities σ_x and σ_y are standard deviations of the new coordinates. As a matter of fact, σ_x^2 and σ_y^2 are respectively the largest and the smallest eigenvalues of $\Sigma_{\tilde{x}\tilde{y}}$.

Similarly, the rotation of angle $-\theta$ can be explicitly computed function of $\Sigma_{\tilde{x}\tilde{y}}$, and one has:

$$x_m = \tilde{x}_m \cos\theta, \; y_m = -\tilde{x}_m \sin\theta. \tag{3.4}$$

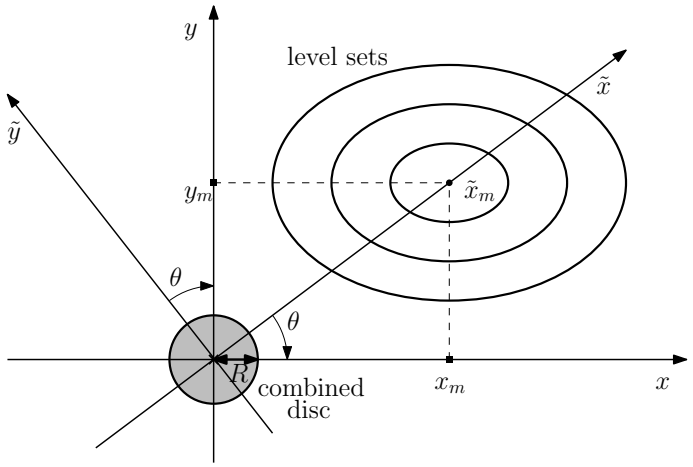


Figure 3.2 – Rotation to the principal axis of the covariance matrix in the encounter plane.

Equation (3.3) defines the so-called short-encounter formula for the probability of collision between two spherical objects under Gaussian-distributed uncertainty. It has the form of a 2-D integral over a disk centered at the origin of a Gaussian function with no cross-terms. This corresponds exactly to Equation (2.4), which can be efficiently and reliably evaluated by Algorithm 13 presented in Section 2.2.

Previous works and discussion

Several techniques for evaluating (2.4) have been developed: Foster's [71], Patera's [171] and Alfano's [3], based on numerical integration schemes. These methods are strongly dependent upon the chosen integration method and need to manage a sensitive trade-off between precision and computation time. Another approach more similar to ours is Chan's [47, 48] who derives a truncated series-based formula, but with an approximation with respect to the initial model.

Furthermore, these methods [5, 48, 171] were unable to guarantee any accuracy requirements. This is because usually either numerical integration schemes or truncated power series were used, but no rigorous proof regarding the method's convergence rate was given. The truncation orders or discretization steps were fixed *by trial and error* or by comparing against other numerical tools which might offer higher accuracy.

In contrast, our solution [J5] benefits both from computer algebra and numerical evaluation tools, which results in a method that is not only reliable (the number of guaranteed correct digits is user-input) but also faster than quadrature schemes.

Finally, we mention another recent work of Garcia-Pelayo [81], where the authors also propose a series-based implementation, but which turns out to be exactly our series without the preconditioning. Hence, this method suffers from important cancellation issues, as already explained in Section 2.2.

Numerical tests

The performance of our method is assessed from two perspectives. First, since it is based on a series expansion, the numerical accuracy varies in function of the number of terms computed. Algorithm 13 offers an automatic way of computing the number of terms needed for a user-required accuracy. We exemplify it in what follows on practical cases. Second, our method is compared with other methods from the literature concerning the quality of the results obtained.

Case #	Input parameters (m)				
	σ_x	σ_y	R	x_m	y_m
Chan 1	50	25	5	10	0
Chan 2	50	25	5	0	10
Chan 3	75	25	5	10	0
Chan 4	75	25	5	0	10
Chan 5	3,000	1,000	10	1,000	0
Chan 6	3,000	1,000	10	0	1,000
Chan 7	3,000	1,000	10	10,000	0
Chan 8	3,000	1,000	10	0	10,000
Chan 9	10,000	1,000	10	10,000	0
Chan 10	10,000	1,000	10	0	10,000
Chan 11	3,000	1,000	50	5,000	0
Chan 12	3,000	1,000	50	0	5,000
CSM 1	152.8814468961533	57.918666623295984	10.3	60.583685340533115	84.875546447209487
CSM 1	5,756.840725983703	15.988242371297744	1.3	115.0558998093139	-81.618369910317043
CSM 3	643.4092722122279	94.230921098486149	5.3	693.4058939950484	102.1772470067133
Alfano 3	114.2585190378857	1.410183033040157	15	0.159164620813659	-3.887207383647396
Alfano 5	177.8109003935867	0.037327944173609	10	2.123006718041866	-1.221789517557463

Table 3.1 – Inputs for test cases from Chan (1–12), CSMs (1–3) and Alfano (3,5)

Figure 3.3 shows the number n of series terms needed in Algorithm 13 for a requested accuracy δ ranging from 10^{-1} to 10^{-13}. We observe that the number of terms for the absolute error to reach the machine precision (10^{-13}) is less 40 in all of Chan's Cases and CSM cases. Note that this number is computed *a priori* and it is a sufficient number. The actual number of terms for the accuracy to be met may be smaller in some cases, but for these practical examples it is not conservative. Chan's cases 8 and 10 are not drawn since the values obtained directly from the minorant/majorant series i.e., lines 2 & 3 of Algorithm 13 are sufficient for the whole range of absolute error considered.

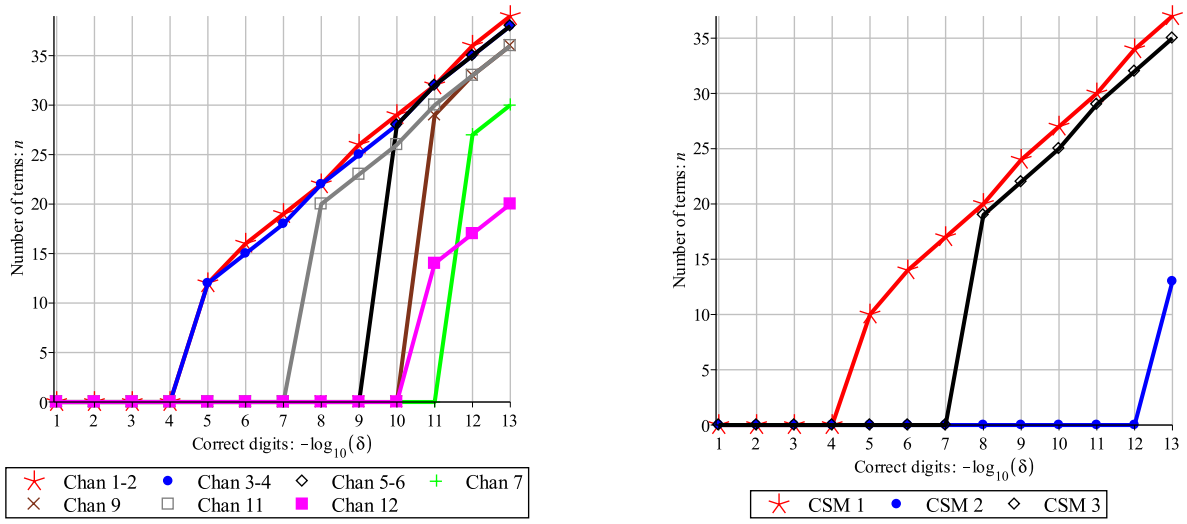


Figure 3.3 – Number of terms in the series needed in Algorithm 13 for a requested accuracy δ ranging from 10^{-1} to 10^{-13}. Cases shown: Chan's Cases (left) and CSM cases (right) from Table 3.1.

Comparison with other methods

Three algorithms from the literature, namely Alfano's [4], Patera's [171] and respectively Chan's [48] have been implemented. The chosen test cases are described in Table 3.1: the first 12 cases can be found in [48, Chapter 5] and are supposed to be representative of real short-term encounters; the next 3 cases are real-case scenarios: the data were retrieved from CSMs (Conjunction Summary Messages) sent by the Joint Space Operations Center to the industrial partner of our study; the last two cases were obtained using the physical parameters of test cases number 3 and 5 provided by Alfano [6].

On the other hand, for Alfano's cases, our Algorithm needs much more terms. In order to reach a good accuracy, Algorithm 13 computes $n = 689$ for Alfano's Case 3 and $n = 10^{13}$ for Alfano's Case 5. This is conservative, since 90 terms for Case 3 and respectively 37000 terms for Case 5 are sufficient to obtain the value given in Table 3.4. This shows however that in some degenerate cases the number of terms needed may increase drastically.

The corresponding results for the probability of collision obtained with different methods are summarized in Tables 3.2, 3.3 and 3.4. The reference values in Table 3.2 were provided by NASA [48] using Foster's method. For Table 3.3, they were given by the industrial partner and for Table 3.4 they were obtained from Monte Carlo trials. All tests were performed with Matlab© R2014a on an Intel® Xeon® at 3.60GHz.

Since in our method the required accuracy can be set *a priori* in Algorithm 13, the obtained values are identical (in most cases) or very close to the reference. For Chan's test cases, number 1 to 12, Patera's method gives also 0% of relative error. For the same examples, Alfano's method also performs well, but it fails for very low probabilities like in test cases 8 and 10. On the other hand, Chan's method gives non negligible relative errors for some cases - namely 1, 2, 3, 4, 8, 10, 12 and CSM1–3. As far as precision is concerned, it is definitely the least effective. It is not surprising since it is based on an additional approximation with respect to the original short-term encounter model. For that reason, it gives meaningless results for the two test cases provided by Alfano [6], see Table 3.4. These examples were originally designed to compare the efficiency of several methods of the literature and are somehow more tedious as far as computation is concerned. They are challenging also for our method, since the number of terms to be considered in the series expansion is important. Nevertheless, the new method gives satisfactory results.

Concerning timings, our method is very fast: for each case tested, the results are obtained in less than one second; in frequent cases, when the bounds l_0 and u_0 are sufficient (lines $2-3$, Algorithm 13), the response is almost instantaneous (10^{-5} seconds).

Case	Collision Probability (-)				
	Alfano	Patera	Chan	Algorithm 13	Reference
Chan 1	9.742×10^{-3}	$9.74\mathbf{1} \times 10^{-3}$	$9.7\mathbf{54} \times 10^{-3}$	9.742×10^{-3}	9.742×10^{-3}
Chan 2	9.181×10^{-3}	9.181×10^{-3}	$9.18\mathbf{9} \times 10^{-3}$	9.181×10^{-3}	9.181×10^{-3}
Chan 3	6.571×10^{-3}	6.571×10^{-3}	$6.5\mathbf{86} \times 10^{-3}$	6.571×10^{-3}	6.571×10^{-3}
Chan 4	6.125×10^{-3}	6.125×10^{-3}	6.135×10^{-3}	6.125×10^{-3}	6.125×10^{-3}
Chan 5	1.577×10^{-5}	1.577×10^{-5}	1.577×10^{-5}	1.577×10^{-5}	1.577×10^{-5}
Chan 6	1.011×10^{-5}	1.011×10^{-5}	1.011×10^{-5}	1.011×10^{-5}	1.011×10^{-5}
Chan 7	6.443×10^{-8}	6.443×10^{-8}	6.443×10^{-8}	6.443×10^{-8}	6.443×10^{-8}
Chan 8	**0**	3.219×10^{-27}	$3.21\mathbf{6} \times 10^{-27}$	3.219×10^{-27}	3.219×10^{-27}
Chan 9	3.033×10^{-6}	3.033×10^{-6}	3.033×10^{-6}	3.033×10^{-6}	3.033×10^{-6}
Chan 10	**0**	9.656×10^{-28}	$9.6\mathbf{45} \times 10^{-28}$	9.656×10^{-28}	9.656×10^{-28}
Chan 11	1.039×10^{-4}	1.039×10^{-4}	1.039×10^{-4}	1.039×10^{-4}	1.039×10^{-4}
Chan 12	1.564×10^{-9}	1.564×10^{-9}	$1.5\mathbf{56} \times 10^{-9}$	1.564×10^{-9}	1.564×10^{-9}

Table 3.2 – Comparison of collision probability value –with 4 significant digits– for Chan's test cases number 1 to 12. The digits different from the reference value are represented in bold.

Case	Collision Probability (-)				
	Alfano	Patera	Chan	Algorithm 13	Reference
CSM 1	1.9002×10^{-3}	1.9001×10^{-3}	$1.\mathbf{8934} \times 10^{-3}$	1.9002×10^{-3}	1.9002×10^{-3}
CSM 2	2.0553×10^{-11}	2.0552×10^{-11}	$2.0\mathbf{135} \times 10^{-11}$	2.0553×10^{-11}	2.0553×10^{-11}
CSM 3	7.2004×10^{-5}	7.2000×10^{-5}	7.2000×10^{-5}	7.2003×10^{-5}	7.2003×10^{-5}

Table 3.3 – Comparison of collision probability value –with 5 significant digits– for tests cases from CSMs. The digits different from the reference value are represented in bold.

Case	Collision Probability (-)				
	Alfano	Patera	Chan	Algorithm 13	Reference
Alfano's No. 3	1.0038×10^{-1}	1.0087×10^{-1}	3.1264×10^{-2}	1.0038×10^{-1}	1.0085×10^{-1}
Alfano's No. 5	4.4712×10^{-2}	4.4520×10^{-2}	1.6618×10^{-77}	4.4509×10^{-2}	4.4499×10^{-2}

Table 3.4 – Comparison of collision probability value - with 5 significant digits - for test cases from [6].

Finally, we mention that we recently compared our preconditioned series with the non-preconditioned one, which was also claimed to be some kind of "new series" by [81]. This comparison was done in collaboration with CNES on their whole database of millions of close-conjunction cases. It turns out that for various *real-life* cases, the cancellation phenomenon is present, as expected from our theoretical results. This renders the non-preconditioned series completely inaccurate, that is, no significant digit or not even the sign can be obtained when evaluating in finite (double) precision. This invalidates the *observations* of [81], which stated that for practical cases, 2 terms of the non-preconditioned series are always enough for accurate evaluation, i.e. roughly 4 or 5 correct digits after the decimal dot. It also provides a practical argument in favor of solid symbolic-numeric evaluation tools.

3.1.2 Brief discussion on long-term/multiple encounters

In the general context of formation flying satellites or proximity operations, the hypothesis of short-term encounters can no longer be considered valid for the calculation of the overall risk of collision [82, 42]. This is mainly due to lower relative velocities (of the order of 1 m/s), so objects spend significant time in proximity to each other and the cross-correlations between the estimated states are not negligible anymore. Several works [170, 144, 46, 64, 173, 39, 58, 116] extended the probability calculation in the case of long-term encounters, also called nonlinear framework, under different assumptions and limitations. For the more general case of assessing the risk of multiple encounters that is, one/many debris with a constellation of satellites, like OneWeb (600 satellites expected, 6 already launched by Feb. 2019) for instance, even fewer studies exist [63, 173, 77].

So far, handling full generality with respect to the dynamics of the objects, the encounter duration, the potentially high number of objects involved, and the distribution of their initial state, was completely out of reach. This concerns both a clear mathematical modeling and a computationally efficient solution.

From a theoretical perspective, a different general convex-optimization-based framework for analysis and optimal control of dynamical systems was proposed in [92, 112, 204, 125, 142]. This is based on the formulation of an infinite-dimensional linear programming problem in the cone of nonnegative measures and so-called Lasserre hierarchy of relaxations [123].

Based on these works, we proposed a fully general mathematical modeling of *the probability of collision of multiple encounters*, in the measure theory framework [R2]. The main ingredients of this modeling are: (1) lifting of the nonlinear dynamics into a linear equation of measures via Liouville's equation; (2) stating a linear optimization problem on measures, whose objective function is exactly the sought probability of collision; (3) practically solving moment problems via a hierarchy of semi-definite optimization.

More precisely, we were able to propose two complementary linear-programming problems on the space of non-negative measures, based on either *computing the probability that no collision occurs*, or the opposite. Obviously, the sum of these two probabilities is 1, but the numerical solving of these

complementary LP problems on measures is done via a sequence of finite dimensional optimization problems, whose *optimal values provide both upper and lower bounds on the sought collision probability.*

While this practical numerical way of solving LP problems on measures is well-known and applicable, in our case, important numerical issues have been identified. Firstly, the dimension of the general problem is currently prohibitive for existing semi-definite solvers. Secondly, even simple examples show that numerical results in low dimension do not achieve a good accuracy. This is thus an important on-going research area for us, whose ramifications led us to consider related moment problems like those presented in Section 2.3. These issues as well as the sometimes partial solutions we provided so far, show that there still is an important need of cross-fertilization between symbolic-numeric and optimization methods, which will be further discussed in Chapter 4. In the meantime, we describe another contribution, where this goal can be achieved.

3.2 Validated impulsive spacecraft rendezvous

Since the first space missions (Gemini, Apollo, Vostok) involving more than one vehicle, space rendezvous between two spacecraft has become a key technology raising relevant open control issues. Formation flight (PRISMA), on-orbit satellite servicing or supply missions to the International Space Station (ISS) are all examples of projects that require adequate rendezvous planning tools. A main challenge is to achieve autonomous far range rendezvous on elliptical orbits while preserving optimality in terms of fuel consumption. As explained in the introduction, the rendezvous (RdV) problem consists in meeting two spacecraft, originally moving on different orbits, in order to match their positions and velocities. The impulsive approximation for the thrust means that an instantaneous velocity increment is applied to the chaser for each impulse. In this setting, one is interested to find the guidance law that achieves the maneuver with the lowest possible fuel consumption. This leads to define a minimum-fuel optimal control problem.

Specifically, in [C8] we focused on the fixed-time minimum-fuel rendezvous between close elliptic orbits of an active spacecraft with a passive target spacecraft, assuming a linear impulsive setting and a linearized Keplerian relative motion [43]. The original optimal control problem is transformed into a semi-infinite convex optimization problem using a relaxation scheme and duality theory in normed linear spaces. A new numerical convergent algorithm based on discretization methods is designed to solve this problem. Its solution is then used in a general simple procedure dedicated to the computation of the optimal velocity increments and optimal impulses locations. As a by-product, one also obtains an analytical solution for the out-of-plane rendezvous problem.

Furthermore, since this algorithm is numeric, one could be interested in *a posteriori* validating the obtained solution with the techniques proposed in Chapter 2. Let us first focus on the optimal control problem.

3.2.1 Optimal control formulation of the rendezvous problem

We consider the relative dynamics in a moving Local-Vertical-Local-Horizontal (LVLH) frame located at the center of gravity of a passive target and which rotates with its angular velocity. In this frame, the state vector $X^T = \begin{bmatrix} x & y & z & v_x & v_y & v_z \end{bmatrix}$ is composed of the positions and velocities of a chaser satellite in the in-track, cross-track and radial axes, respectively. Using the true anomaly of the target-vehicle orbit as the independent variable, and assuming a linearization of the relative equations of motion[1], a system of linear differential equations with periodic coefficients is easily obtained. The considered minimum-fuel linearized rendezvous problem may be reformulated as the following optimal control problem.

Problem 4. (Optimal control problem) Let $\mathcal{L}_{1,p}([\nu_0, \nu_f], \mathbb{R}^r)$ be the normed linear space of Lebesgue integrable functions $u : [\nu_0, \nu_f] \to \mathbb{R}^r$, with the norm given by $\|u\|_{1,p} = \int_{\nu_0}^{\nu_f} \|u(\nu)\|_p \mathrm{d}\nu$, and where $\|\cdot\|_p$

[1]Their validity is guaranteed when the distance between the target and the chaser is assumed to be small compared to the radius of the target vehicle orbit.

is the usual ℓ_p-norm of an r-dimensional vector.

Find $\bar{u} \in \mathcal{L}_{1,p}([\nu_0, \nu_f], \mathbb{R}^r)$ solution of the optimal control problem:

$$\begin{array}{ll} \inf\limits_{u} & \|u\|_{1,p} = \inf\limits_{u} \displaystyle\int_{\nu_0}^{\nu_f} \|u(\nu)\|_p \mathrm{d}\nu \\ \text{s.t.} & X'(\nu) = A(\nu)X(\nu) + Bu(\nu),\ \forall\, \nu \in\ [\nu_0, \nu_f] \\ & X(\nu_0) = X_0,\ X(\nu_f) = X_f \in \mathbb{R}^n,\ \nu_0,\ \nu_f \text{ fixed}, \end{array} \tag{3.5}$$

where matrices $A(\nu)$ and B define the state-space model of relative dynamics given by [212]:

$$A(\nu) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3/(1+e\cos(\nu)) & -2 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \mathbb{O}_{3\times 3} \\ \mathbb{1}_3 \end{bmatrix}. \tag{3.6}$$

The form of these matrices shows that the equations describing motion in the plane of the target-vehicle orbit and those describing motion normal to the orbit plane can be decoupled and handled separately. Therefore, the out-of-plane and in-plane rendezvous will be dealt with independently in sequel developments. Indeed, the state vector dimension and the number of inputs in (3.5) are denoted n and r, respectively with $n = 2$, $r = 1$ for the out-of-plane case and $n = 4$, $r = 2$ for the in-plane case.

Remark 3.2.1. In Problem 4, the 1-norm cost captures indirectly the consumption of fuel used. In fact, the performance index used in Problem 4 is an upper-bound expressed as an angular velocity, on the usual characteristic velocity expressed in m/s.

Previous works Indirect approaches, based on the optimality conditions derived from the Pontryagin's maximum principle and leading to the so-called primer vector theory ([130]), have been extensively studied [43, 136, 179, 11]. However, due to the nonconvex and polynomial nature of these conditions, numerical approaches remained either too complicated, or made use of heuristics which exhibit only suboptimal solutions on some instances. An important theoretical contribution for the optimal control problem was proposed in [161] and revisited in [54]. The idea is to recast it as a simpler optimization problem using a relaxation scheme and the duality theory in minimum-norm problems. In [54], a linear programming problem on measures is formulated, which is solved by a hierarchy of linear-matrix inequalities. However, the numerical solving of these hierarchies remains cumbersome.

Following [161], our contribution was to propose a new simpler convergent iterative numerical algorithm to solve the fixed-time impulsive linear rendezvous (without fixing *a priori* the number of impulses). This was based on the observation that by topological duality results [139, 161], the problem to be solved becomes a Semi-Infinite Convex Programming (SICP) Problem. In turn, SICP can be solved by iterative discretization methods [184]. For instance, one of the most classical algorithms, which can be interpreted in the SICP context, with the further simplification of the constraints being linear, is the Remez algorithm [49].

In order to obtain the SICP formulation, the theoretical framework employed [161, C8] is summarized below:

- **Formulation of a minimum norm moment problem.** It is known that the solutions of the uncontrolled linearized dynamics in equation (3.5) form an n-dimensional affine space and a closed-form fundamental matrix, say $\varphi(\nu)$, was provided by Yamanaka and Ankersen [230]. Defining the matrix $Y(\nu) = \varphi^{-1}(\nu)B = \begin{bmatrix} y_1(\nu) \cdots y_n(\nu) \end{bmatrix}^T \in \mathbb{R}^{n\times r}$, one has:

$$\begin{array}{rcl} c & = & \varphi^{-1}(\nu_f)X(\nu_f) - \varphi^{-1}(\nu_0)X_0, \\ & = & \displaystyle\int_{\nu_0}^{\nu_f} \varphi^{-1}(\sigma)Bu(\sigma)\mathrm{d}\sigma = \int_{\nu_0}^{\nu_f} Y(\nu)u(\sigma)\mathrm{d}\sigma. \end{array} \tag{3.7}$$

Hence, Problem 4 can be equivalently written as:

Problem 5. (Minimum norm moment problem) Find $\bar{u}(t) \in \mathcal{L}_{1,p}([\nu_0, \nu_f], \mathbb{R}^r)$ solution of the minimum norm moment problem:

$$\begin{aligned} \inf_u \quad & \|u\|_{1,p} = \inf_u \int_{\nu_0}^{\nu_f} \|u(\nu)\|_p \mathrm{d}\nu \\ \text{s.t.} \quad & \int_{\nu_0}^{\nu_f} Y(\sigma)u(\sigma)\mathrm{d}\sigma = c, \ \nu_0, \ \nu_f \text{ fixed.} \end{aligned} \tag{3.8}$$

Several remarks are important at this point:

- the specific matrices $Y(\nu)$ encountered in the rendezvous problem, $y_1(\nu) \cdots y_n(\nu)$ are linearly independent elements of $\mathcal{C}([\nu_0, \nu_f], \mathbb{R}^r)$;
- when the fundamental matrix $\varphi(\nu)$ cannot be obtained exactly in closed-form, techniques from Section 2.4 can be employed to obtain efficient polynomial approximations for its entries;
- Problem 5 may not reach its optimal solution due to concentration effects [188]. This is mainly due to the fact that the functional space $\mathcal{L}_{1,p}([\nu_0, \nu_f], \mathbb{R}^r)$ in which the optimal solution is sought, is not the topological dual of any other functional space [139]. It is then necessary to consider a so-called *relaxed problem*, whose solutions are thought of as generalized solutions of Problem 5. Specifically, they are searched for in the space of functions of bounded variation $\mathrm{BV}([\nu_0, \nu_f], \mathbb{R}^r)$ [161, 139].

- **Relaxation of the original problem.** It is important to note that a unique association can be made between a function $g \in \mathrm{BV}([\nu_0, \nu_f], \mathbb{R}^r)$ and a linear functional belonging to the topological dual space $l \in \mathcal{C}^*([\nu_0, \nu_f], \mathbb{R}^r)$ of the space $\mathcal{C}([\nu_0, \nu_f], \mathbb{R}^r)$, by the Riesz Representation Theorem [139], via the duality bracket:

$$l(y_i) = \langle y_i(\cdot), l \rangle = \int_{\nu_0}^{\nu_f} y_i(\nu)^T \mathrm{d}g(\nu). \tag{3.9}$$

Hence, Problem 5 may be relaxed as:

Problem 6. (Linear minimum norm problem)
Find a linear functional $\bar{l} \in \mathcal{C}^*([\nu_0, \nu_f], \mathbb{R}^r)$ solution of the linear minimum norm problem:

$$\begin{aligned} \bar{\eta} = \inf_l \quad & \|l\| \\ \text{s.t.} \quad & l(y_i) = \langle y_i(\cdot), l \rangle = c_i, \ \forall\, i = 1, \cdots, n. \end{aligned} \tag{3.10}$$

Note that the dual norm of the functional l is defined as $\|l\| = \sup_{\|y(\cdot)\|_q \leqslant 1} |l(y)|$, for continuous functions $y : [\nu_0, \nu_f] \to \mathbb{R}^r$, equipped with the norm $\|y(\cdot)\|_q = \sup_{\nu_0 \leqslant \nu \leqslant \nu_f} \|y(\nu)\|_q$ (and with $\frac{1}{p} + \frac{1}{q} = 1$).
It is shown in [161] that the infimum of Problem 6, denoted by $\bar{\eta}$, is reached and that it is equal to the infimum of Problem 5.

Although Problem 6 is still an infinite-dimensional optimization problem, it is particularly appealing due to the important following result [161] [139, Chapter 5].

Theorem 3.2.2. Let $y_i(\cdot) \in \mathcal{C}([\nu_0, \nu_f], \mathbb{R}^r)$, $\forall\, i = 1, \cdots, n$ and suppose that

$$D = \{l \in \mathcal{C}^* : \langle y_i(\cdot), l \rangle = c_i, \ i = 1, \cdots, n\} \neq \emptyset, \tag{3.11}$$

then

$$\bar{\eta} = \min_{l \in D} \|l\| = \max_{\|Y^T(\nu)\lambda\|_q \leqslant 1} c^T \lambda. \tag{3.12}$$

In addition, let $\bar{l}$ and $\bar{\lambda}$ be optimal solutions of (3.12), $\bar{\lambda} = \mathrm{Arg}[\max_{\|Y^T(\nu)\lambda\|_q \leqslant 1} c^T \lambda]$ and let $\bar{y}(\nu) = \sum_{i=1}^{n} \bar{\lambda}_i y_i(\nu) = Y^T(\nu)\bar{\lambda} \in \mathbb{R}^r$. Then the optimal $\bar{l}$ is aligned with the optimal $\bar{y}$:

$$\langle \bar{y}(\cdot), \bar{l} \rangle \quad = \quad \int_{\nu_0}^{\nu_f} \bar{\lambda}^T Y(\nu) \mathrm{d}\bar{g}(\nu) = \|\bar{y}(\cdot)\|_q \|\bar{l}\|. \tag{3.13}$$

The two problems defined in eq. (3.12) may be considered as dual through the equality of the optimal values of their respective objectives and the relation between their solutions thanks to the alignment condition in eq. (3.13). This results in a significant simplification: the infinite-dimensional optimization Problem 6 has been converted to a search of an optimal vector $\bar{\lambda}$ in a finite-dimensional vector space submitted to a continuum of constraints, yielding a Semi-Infinite Convex Problem (SICP):

Problem 7. (SICP problem) Find $\bar{\lambda} \in \mathbb{R}^n$ solution of

$$\begin{aligned} \bar{\mu} = \min_{\lambda \in \mathbb{R}^n} \quad & -c^T \lambda \\ & \|Y^T(\nu)\lambda\|_q \leqslant 1. \end{aligned} \tag{3.14}$$

Note that $\bar{\mu} = -\bar{\eta}$. An efficient numerical method for solving Problem 7 is given in Sec. 3.2.2. Once its solution is obtained, the alignment relation between the function $\bar{y}(\cdot)$ element of the Banach space $\mathcal{C}([\nu_0, \nu_f], \mathbb{R}^r)$ and the functional $\bar{l}$ belonging to its dual space $\mathcal{C}^*([\nu_0, \nu_f], \mathbb{R}^r)$ is particularly important to get back to the optimal control searched for as a bounded variation function.

Theorem 3.2.3. (Recover solution as step function [161])
Let $y_i(\cdot) \in \mathcal{C}([\nu_0, \nu_f], \mathbb{R}^r)$, $i = 1, \ldots, n$ and $\bar{\lambda} \in \mathbb{R}^n$ be an optimal solution of Problem (3.14). Define the sets $\Gamma_s = \{\hat{\nu} \in [\nu_0, \nu_f] : |\bar{y}_s(\hat{\nu})| = 1\}$ and $\Gamma = \left\{ \hat{\nu} \in [\nu_0, \nu_f],\ \|\bar{y}(\hat{\nu})\|_q = \max_{\nu_0 \leqslant \nu \leqslant \nu_f} \|\bar{y}(\nu)\|_q = 1 \right\}$. Note that $\Gamma = \cup_s \Gamma_s$ for $p = 1$. There is an optimal solution $\bar{g}(\cdot) \in \mathrm{BV}([\nu_0, \nu_f], \mathbb{R}^r)$ corresponding to the optimal solution $\bar{l}$ in equation (3.9), which is a step function with at most n points of discontinuity $\hat{\nu}_j \in \Gamma$, $j = 1, \cdots, N \leqslant n$. Its jumps are given by:

$$\begin{aligned} & \bar{g}_s(\hat{\nu}_j) - \bar{g}_s(\hat{\nu}_j^-) = \alpha_{\hat{\nu}_j} \mathrm{sgn}(\bar{y}_s(\hat{\nu}_j)) \chi_{\Gamma_j},\ \alpha_{\hat{\nu}_j} > 0, \\ & \text{when } p = 1, \\ & \text{or} \\ & \bar{g}_s(\hat{\nu}_j) - \bar{g}_s(\hat{\nu}_j^-) = \alpha_{\hat{\nu}_j} |\bar{y}_s(\hat{\nu}_j)|^{q-1} \mathrm{sgn}(\bar{y}_s(\hat{\nu}_j)), \\ & \text{when } 1 < p < \infty, \end{aligned} \tag{3.15}$$

for $s = 1, \cdots, r$ and $\alpha_{\hat{\nu}_j}$ solutions of the linear system:

$$\sum_{j=1}^{N} \beta_i(\hat{\nu}_j) \alpha_{\hat{\nu}_j} = c_i,\ i = 1, \cdots, n, \tag{3.16}$$

where $\beta_i(\hat{\nu}_j)$ are given by:

$$\begin{aligned} & \beta_i(\hat{\nu}_j) = \sum_{s=1}^{r} y_{i,s}(\hat{\nu}_j) \mathrm{sgn}(\bar{y}_s(\hat{\nu}_j)),\ \text{when } p = 1, \\ & \text{or} \\ & \beta_i(\hat{\nu}_j) = \sum_{s=1}^{r} y_{i,s}(\hat{\nu}_j) |\bar{y}_s(\hat{\nu}_j)|^{q-1} \mathrm{sgn}(\bar{y}_s(\hat{\nu}_j)), \\ & \text{when } 1 < p < \infty, \end{aligned} \tag{3.17}$$

for all $j = 1, \cdots, N$.

This theorem states important results that have been known for a while in the aerospace community but whose value has not been completely exploited to derive efficient numerical algorithms for impulsive maneuvers design. First, it says that the optimal controlled trajectory for the minimum-fuel Keplerian linearized elliptic rendezvous problem is purely impulsive and that the number of impulses is upper-limited by n which is the dimension of the fixed final conditions of the optimal control problem.

Remark 3.2.4. It is also shown in [161] that a sequence of functions $u_\varepsilon(\cdot) \in \mathcal{L}_{1,p}([\nu_0, \nu_f], \mathbb{R}^r)$ converges to a linear combination of $\delta(\cdot)$ functions corresponding to the function $\bar{g}(\cdot)$ with equal norms. Let

$\Delta V(\hat{\nu}_j) = \bar{g}(\hat{\nu}_j) - \bar{g}(\hat{\nu}_j^-)$, then roughly speaking, this may be described by:

$$\bar{u}_\varepsilon(\nu) \quad \to \quad \sum_{j=1}^{N} \Delta V(\hat{\nu}_j)\delta(\hat{\nu}_j - \nu),\ \varepsilon \to 0. \tag{3.18}$$

Indeed, the initial optimal control problem amounts to finding the sequences of optimal impulse locations $\{\hat{\nu}_i\}_{i=1,\cdots,N}$ and optimal impulse vectors $\{\Delta V(\hat{\nu}_i)\}_{i=1,\cdots,N}$ verifying the boundary equation:

$$c = \sum_{i=1}^{N} Y(\hat{\nu}_i)\Delta V(\hat{\nu}_i). \tag{3.19}$$

3.2.2 A convergent discretization approach

Based on Problem 7 and Theorem 3.2.3, a convergent iterative numerical method is presented. Firstly, the SICP Problem 7 is solved using Algorithm 19, whose convergence proof was given in [C8]. Namely, the idea is to consider an efficient discretization procedure [184, Chap.7], by constructing a sequence of finite subsets $\Theta_i \subseteq \Theta := [\nu_0, \nu_f]$ and solving Problem 7 on this discretization Θ_i respectively. The discretization procedure implemented was to add at each iteration the *point which violates most the constraints*, see line 6 of Algorithm 19. Under certain assumptions discussed in [C8] (among which Θ is compact, Slater condition holds for Problem 7, the initial set Θ_0 can be chosen in a convenient way), the sequence of solutions of finite dimensional problems, given in line 7 of the algorithm, is convergent to that of the SICP Problem 7.

Algorithm 19 Numerical procedure for solving Problem 7.

Input: interval $\Theta = [\nu_0, \nu_f]$, matrix $Y(\nu)$, initial condition c, accuracy ε
Output: $\mu^{(i)}$ and $\lambda^{(i)}$ numerical solution of Problem 7
Init:

1 $i \leftarrow 0$;

2 $\Theta_0 \leftarrow \{\theta_0; \theta_1\} \subset \Theta$ s.t. $\theta_0 - \theta_1 \neq k\pi$;

3 Solve eq. (3.19) for ΔV_0 and ΔV_1;

4 Solve for $\lambda^{(0)}$ the system $Y^T(\theta_k)\lambda^{(0)} = \Delta V_k / \|\Delta V_k\|_q$, $k = 0, 1$.

5 **While** $\max_{\theta\in\Theta} \|Y(\theta)^T\lambda^{(i)}\|_q - 1 > \varepsilon$ **do**

6 $i \leftarrow i + 1$; $\Theta_i \leftarrow \Theta_{i-1} \cup \left\{ \arg \left[\max_{\theta\in\Theta} \|Y^T(\theta)\lambda^{(i)}\|_q \right] \right\}$;

7 **Find** $\lambda^{(i)}$ solution of discretized problem:

$$\begin{array}{rcll} \mu^{(i)} & = & \inf\limits_{\lambda\in\mathbb{R}^n} & -c^T\lambda \\ & & \text{s.t.} & \|Y^T(\theta_k)\lambda\|_q \leqslant 1 \quad \text{for all } \theta_k \in \Theta_i \end{array}$$

8 **return** $\mu^{(i)}, \lambda^{(i)}$.

In practice, the norms and corresponding problems to be solved are:
– for a gimbaled single thruster one has $p = q = 2$, which gives a semi-infinite positive semi-definite (SDP) problem:

$$\begin{array}{ll} \inf\limits_{\lambda\in\mathbb{R}^n} & -c^T\lambda \\ \text{s.t.} & \begin{bmatrix} -1 & \lambda^T Y(\nu) \\ Y^T(\nu)\lambda & -\mathbf{1} \end{bmatrix} \preceq \mathbf{0},\ \forall\, \nu \in [\nu_0, \nu_f]; \end{array} \tag{3.20}$$

– for 6 ungimbaled identical thrusters, one has $p = 1,\ q = \infty$ which gives a semi-infinite linear programing (LP) problem:

$$\begin{aligned} &\inf_{\lambda\in\mathbb{R}^n} \; -c^T\lambda \\ &\text{s.t.} \; \left|\sum_{i=1}^{n} \lambda_i y_{i,s}(\nu)\right| \leqslant 1, \forall\, \nu\in[\nu_0,\nu_f],\; s=1,\dots,r. \end{aligned} \tag{3.21}$$

We also obtained in [C8] an estimation of the accuracy of the obtained numerical value $\mu^{(i)}$ with respect to the optimal cost η in Problem 6. The discretization method produces outer approximations of a solution of the SIP problem, thus providing increasing lower bounds for its solution:

$$\bar{\eta} = \max_{\|Y^T(\nu)\lambda\|_q\leqslant 1} c^T\lambda = -\min_{\|Y^T(\nu)\lambda\|_q\leqslant 1} -c^T\lambda \leqslant -\mu^{(i)}. \tag{3.22}$$

A lower bound can also be obtained. If after i iterations, $\max_{\theta\in\Theta}\|Y(\theta)^T\lambda^{(i)}\|_q \leqslant 1+\varepsilon$, where ε is a user defined input parameter, then

$$\frac{-\mu^{(i)}}{1+\varepsilon} \leqslant \bar{\eta}. \tag{3.23}$$

Thus, given ε, the output $\mu^{(i)}, \lambda^{(i)}$ of Algorithm 19 provides a good numerical approximation for the optimal cost of the original problem, $\bar{\eta}$. Secondly, one identifies the impulse locations and velocity increments in Algorithm 20 which is based on Theorem 3.2.3. Specifically, the impulse locations can be identified by finding

$$\Gamma = \{\hat{\nu}_k \in [\nu_0,\nu_f] :\; \|Y(\hat{\nu}_k)^T\lambda^{(i)}\|_q = 1\}.$$

This is done numerically on a grid of $[\nu_0,\nu_f]$ (lines 1-2 of Algorithm 20). Then one solves the system given in eq. (3.19). This is always possible, since, according to Neustadt, the following holds: if at most n locations are found in Γ, the system is underdetermined/determined and it has at least one solution (lines 5-6); if more than n locations are found in Γ, one can select n among them such that the system has a solution (lines 7-8).

Algorithm 20 Numerical Reconstruction of impulse locations and vectors

Input: interval $\Theta=[\nu_0,\nu_f]$, matrix $Y(\nu)$, initial condition c, accuracy ε, solution $\lambda^{(i)}\in\mathbb{R}^n$ of Pb. 7
Output: impulse locations and impulse vectors $\Gamma_{imp}, \{\Delta V_i\}$

1 $\Gamma_d \leftarrow$ discretized grid of $[\nu_0,\nu_f]$
2 $\Gamma \leftarrow \{\hat{\nu}_k\in\Gamma_d : \|Y(\hat{\nu}_k)^T\lambda^{(i)}\|_q - 1 \in [-\varepsilon,\varepsilon]\}$
3 $N \leftarrow$ size(Γ)
4 **if** $(N\leqslant n)$ **then**
5 $\Gamma_{imp}\leftarrow\Gamma$
6 **Solve** for $\Delta V_i, i=1,\dots,N$, the linear system $c=\sum_{\hat{\nu}_i\in\Gamma_{imp}} Y(\hat{\nu}_i)\Delta V_i$.
7 **else**
8 $\Gamma_{imp}\leftarrow$ **Choose** n points in Γ s.t. the linear system $c=\sum_{\hat{\nu}_i\in\Gamma_{imp}} Y(\hat{\nu}_i)\Delta V_i$ has a solution.
9 **return** $\Gamma_{imp}, \{\Delta V_i\}$.

3.2.3 Numerical example

One of the numerical examples of [C8] is presented in what follows.

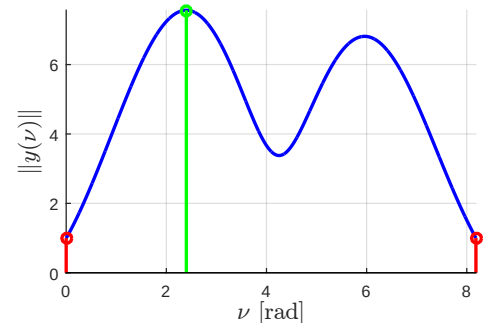
Example 3.2.5 (ATV Example - Numerical solution)**.** It concerns the in-plane motion case and related to some example of the Automated Transfer Vehicle (ATV) setup [117]. The parameters of the reference orbit and of the rendezvous are given in Table 3.5.

Semi-major axis	a = 6763 km.
Inclination	i=52 deg.
Argument of perigee	ω=0 deg.
Longitude of the ascending node	Ω= 0 deg.
Eccentricity	e = 0.0052
Initial time	ν_0 = 0 rad.
Initial state vector X_0^T	[-30 0.5 8.514 0] km. - m/s.
Initial state vector $\tilde{X}_0^T$	[-51.9222 0.0865 0.95734 0].10^4
Final anomaly	ν_f = 8.1832 rad.
Duration	$t_f - t_0$ = 7200 s.
Final state vector X_f^T	[-100 0 0 0] m. - m/s.
Final state vector $\tilde{X}_f^T$	[-76.3818 0 69.1519 0]

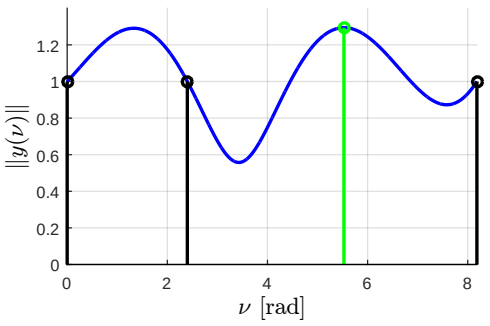
Table 3.5 – Parameters of the ATV example.

For the in-plane rendezvous, two different cases are studied: I- a single gimbaled thruster using $\mathcal{L}_{1,2}$ norm and II- 6 ungimbaled thrusters with $\mathcal{L}_{1,1}$ norm. Note that the numerical solver employed is SDPA [231].

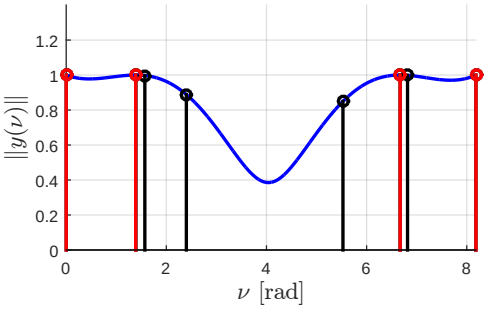
I: $\mathcal{L}_{1,2}$ norm For $\varepsilon = 10^{-4}$, Algorithm 19 needs 6 iterations and returns an approximation of the optimal solution $\bar{\lambda} = [-1.177,\ 1.132,\ -1.571,\ 14.36]^T .10^{-4}$. Then, Algorithm 20 builds a 4-impulse minimum-fuel solution with a cost of 10.7989 m/s. The approximations of optimal impulse locations are given by $\Gamma_{imp} = \{0,\ 1.3872,\ 6.6639,\ 8.1832\}$ [rad]. This process is illustrated in Figure 3.4.



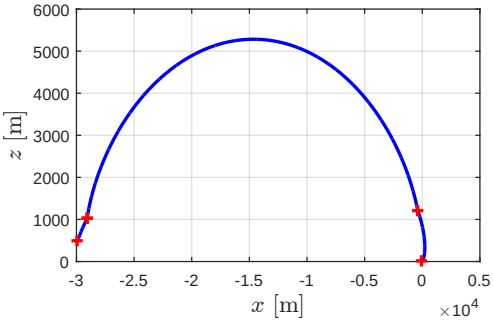
(a) Primer vector norm: Initialization.



(b) Primer vector norm: First iteration.



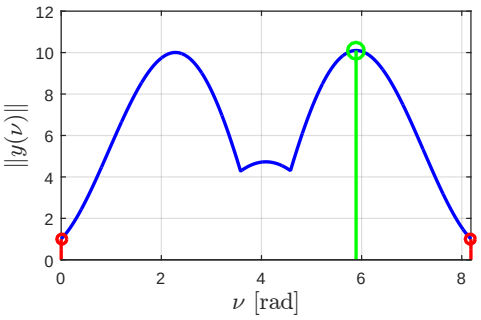
(c) Primer vector norm: Final iteration.



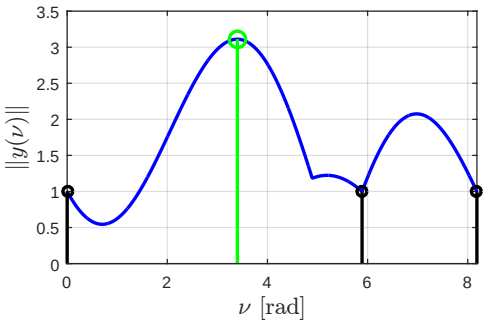
(d) Optimal trajectory in (x, z) plane.

Figure 3.4 – $\mathcal{L}_{1,2}$ norm, ATV example. The next point added to the discretization scheme is shown in green; the final impulse locations are shown in red.

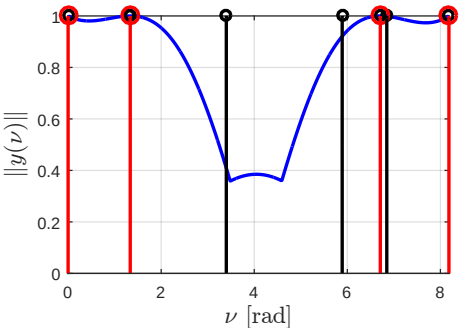
II: $\mathcal{L}_{1,1}$ norm Similarly, the $\mathcal{L}_{1,1}$ case is run considering a tolerance parameter $\varepsilon = 10^{-4}$. After 5 iterations of Algorithm 19, $\bar{\lambda} = [0.1041 \ -0.1083 \ 0.1373 \ 1.2679]^T$ (see Figure 3.5). Then, the impulse locations are given by $\Gamma_{imp} = \{0,\ 1.3352,\ 6.7087,\ 8.1832\}$ [rad], with a fuel-consumption for this in-plane maneuver of 10.8415 m/s. The comparisons of $\mathcal{L}_{1,2}$ and $\mathcal{L}_{1,1}$ fuel-minimum solutions show a minor difference with respect to the optimal locations and overall consumption.



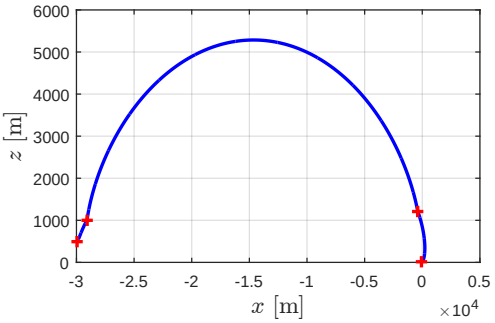
(a) Primer vector norm: Initialization.



(b) Primer vector norm: First iteration.



(c) Primer vector norm: Final iteration.



(d) Optimal trajectory in (x, z) plane.

Figure 3.5 – $\mathcal{L}_{1,1}$ norm, ATV example. The next point added to the discretization scheme is shown in green; the final impulse locations are shown in red.

3.2.4 A posteriori validation

As depicted in Figures 3.5d and 3.4d the trajectory in the (x, z) plane seems to be correct. It is however the result of a numerical algorithm and a numerical integration of the controlled ODE system (3.6), so an *a posteriori* certification can be useful. Furthermore, in general, a closed-form fundamental matrix for such systems is unavailable, which would hinder the use Algorithm 19. A solution to these two problems is to use the RPAs techniques proposed in Section 2.4. This would offer in general both an approximation for a fundamental/transition matrix of linear ODE systems and a certification step, which consists in validating the obtained controlled trajectories.

Concretely, we want to provide rigorous polynomial approximations of the chaser's in-plane trajectories over each segment $[\nu_i, \nu_{i+1}]$, where ν_i and ν_{i+1} are two consecutive impulse times. These are precisely IVP problems over compact segments, where at each step i, we propagate the position and velocity of the chaser from ν_i to ν_{i+1} and compute the new conditions at ν_{i+1} by incrementing the obtained velocity $(x'(\nu_{i+1}), z'(\nu_{i+1}))$ by the impulse velocity vector $(\Delta x'(\nu_{i+1}), \Delta z'(\nu_{i+1}))$. We focus on the in-plane problem, because the out-of-plane is a simple oscillator, which has a classical solution. For the notations to be clear, at step i over $[\nu_i, \nu_{i+1}]$, let's call $\nu_o = \nu_i$ and $\nu_f = \nu_{i+1}$.

The in-plane part of the ODE system (3.6), is recast under a form similar to what was discussed in

Section 2.4, as follows:

$$\begin{gathered} z''(\nu) + \left(4 - \frac{3}{1+e\cos\nu}\right) z(\nu) = \zeta, \\ x(\nu) = x(\nu_0) + (x'(\nu_0) - 2z(\nu_0))(\nu - \nu_0) + 2\int_{\nu_0}^{\nu} z(s)\mathrm{d}s, \\ \zeta = 4z(\nu_0) - 2x'(\nu_0) \quad \text{and} \quad \nu \in [\nu_0, \nu_f]. \end{gathered} \tag{3.24}$$

We use the affine rescaling $\nu = \nu(t) = \frac{\nu_o+\nu_f}{2} + \frac{\nu_f-\nu_o}{2}t$ for $t \in [-1,1]$, and setting $f(t) = z \circ \nu = z\left(\frac{\nu_o+\nu_f}{2} + \frac{\nu_f-\nu_o}{2}t\right)$, we get the following LODE over $[-1,1]$:

$$f''(t) + \alpha_0(t) f(t) = \omega^2 \zeta, \tag{3.25}$$

with

$$\alpha_0(t) = \omega^2\left(4 - \frac{3}{1+e\cos(\xi+\omega t)}\right), \quad \xi = \frac{\nu_o+\nu_f}{2}, \quad \omega = \frac{\nu_f-\nu_o}{2}.$$

This equation is interesting for several reasons. First, the coefficient $\alpha_0(t)$ is not polynomial so that all the steps of the method presented in Section 2.4 are involved to get a rigorous error bound. Then, even if the denominator in $\alpha_0(t)$ never vanishes over $\mathbb{R}$, the coefficient has singularities in the complex plane. Hence, if the total time of the mission is long enough (that is, if ω is large enough), a Taylor method needs to use interval subdivision, while Chebyshev methods remain valid over the whole interval. Finally, choosing various values for e and ω will allow us to test the method in difficult cases (that is, when e gets close to 1 or when $\nu_f - \nu_o$ gets large). This is shown in what follows.

Approximating the non-polynomial coefficient. The first step consists in computing a rigorous $\mathcal{U}^1$-polynomial approximation of $\alpha_0(t)$. The cosine term $\cos(\xi+\omega t)$ is a solution of $f'' + \omega^2 f = 0$, and can be rigorously approximated using our method in this easy case. We consider that $\xi = 0$. When ω gets large, the Chebyshev series will converge more slowly so that we will need a larger degree d. Then, to approximate the inverse series, it is clear that when e gets close to 1, the minimal value of the denominator gets close to 0 and we will again need a large degree to get an accurate approximation. In Table 3.6, we sum up approximation results for 3 different values of e and 3 different values of ω.

e	ω	$d=10$	$d=20$	$d=50$	$d=100$	$d=200$
.0052	$\pi/4$	$3\cdot10^{-15}$	$2\cdot10^{-28}$	$1\cdot10^{-64}$	≈ 0	≈ 0
	π	$5\cdot10^{-7}$	$1\cdot10^{-13}$	$7\cdot10^{-34}$	$1\cdot10^{-65}$	≈ 0
	4π	1.8	$7\cdot10^{-3}$	$1\cdot10^{-8}$	$3\cdot10^{-19}$	$2\cdot10^{-39}$
.2	$\pi/4$	$4\cdot10^{-12}$	$4\cdot10^{-22}$	$4\cdot10^{-52}$	≈ 0	≈ 0
	π	$1\cdot10^{-3}$	$9\cdot10^{-8}$	$4\cdot10^{-19}$	$7\cdot10^{-39}$	$1\cdot10^{-77}$
	4π	$7\cdot10$	$1\cdot10^{1}$	$4\cdot10^{-2}$	$4\cdot10^{-6}$	$3\cdot10^{-14}$
.80621	$\pi/4$	$3\cdot10^{-14}$	$8\cdot10^{-20}$	$4\cdot10^{-47}$	$1\cdot10^{-92}$	≈ 0
	π	$4\cdot10^{-1}$	$8\cdot10^{-3}$	$3\cdot10^{-9}$	$2\cdot10^{-19}$	$7\cdot10^{-40}$
	4π	$2\cdot10^{3}$	$1\cdot10^{3}$	$2\cdot10^{2}$	$1\cdot10^{1}$	$4\cdot10^{-2}$

Table 3.6 – Chebyshev approximations errors of the coefficient $\alpha_0(t)$, function of eccentricity, interval and degree.

Obtaining a contracting Newton-like operator. Having a rigorous approximation $a_0(t)$ of degree d for $\alpha_0(t)$, we can now apply the validation method to equation (3.25). The integral transform produces coefficients:

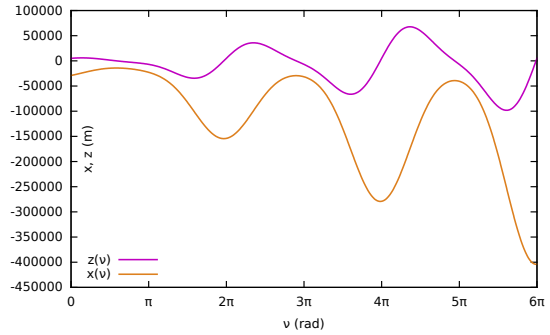
$$\beta_0(t) = t a_0(t), \qquad \beta_1(t) = -a_0(t),$$

which induces an almost-banded structure for the operator $\mathbf{K}$ (the width of the band is $d+1$). In Table 3.7, we present the parameter n which is the truncation order for $\mathbf{K}$, to obtain a contracting Newton operator $\mathbf{T}$, in three different situations depending on e, ω and the appropriately chosen d. The final column gives the certified upper bound given by the algorithm for the Lipschitz constant of $\mathbf{T}$. A numerical solution of equation (3.24) can be now validated.

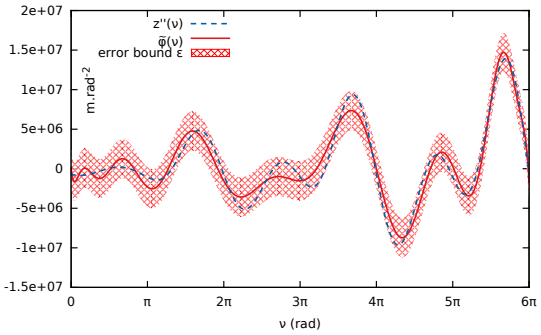
e	ω	d	n	bound
.0052	$\pi/4$	10	30	$1.5 \cdot 10^{-3}$
.0052	4π	30	90	$4.1 \cdot 10^{-2}$
.2	π	20	60	$6.4 \cdot 10^{-3}$
.80621	$\pi/4$	10	30	$3.2 \cdot 10^{-3}$
.80621	π	30	90	$4.9 \cdot 10^{-2}$
.80621	4π	500	1500	$2.1 \cdot 10^{-3}$

Table 3.7 – Validation of the Lipschitz constant of the quasi-Newton operator $\mathbf{T}$.

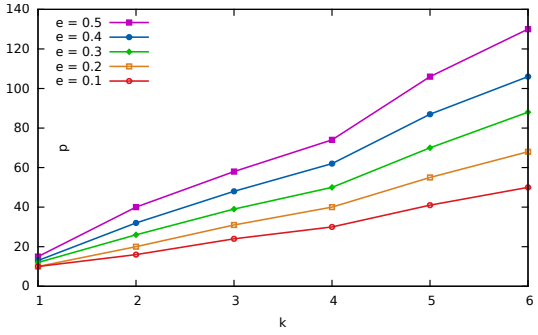
This is firstly shown on an academic example. Let us fix the eccentricity $e = 0.5$ (in order to observe the approximation errors at drawing scale), the interval $[\nu_0, \nu_f] = [0, 6\pi]$ corresponding to 3 periods, and the initial conditions $(x(\nu_0), z(\nu_0), x'(\nu_0), z'(\nu_0)) = (-3{\cdot}10^4 \text{ m}, 5{\cdot}10^3 \text{ m}, 9{\cdot}10^3 \text{ m} \cdot \text{rad}^{-1}, 4{\cdot}10^3 \text{ m} \cdot \text{rad}^{-1})$. The corresponding functions $x(\nu)$ and $z(\nu)$ are plotted in Figure 3.6a. Figure 3.6b represents an approximation



(a) Exact representation of $x(\nu)$ and $z(\nu)$ for $\nu \in [0, 6\pi]$ and $e = 0.5$.



(b) Rigorous approximation $\widetilde{\varphi}$ of degree 18 of z'', over $[0, 6\pi]$ and for $e = 0.5$.



(c) Approximation degree d needed to rigorously approximate z on $[0, k\pi]$ within an error of 1m, in function of k and the eccentricity e.

Figure 3.6 – Different validation results related to the spacecraft rendezvous problem.

of degree $n = 18$ of $z''(\nu)$ (radial acceleration), together with the rigorous error bound obtained by our method. The dashed curve corresponds to the exact solution, which as expected lies inside the tube

defined by our rigorous approximation. One notices that we obtain a tight error bound, even for the $\|\cdot\|_\infty$ norm. Figure 3.6c gives the minimal degree d corresponding to an approximation of z for which our algorithm is able to certify an error below 1m, in function of the period length and the eccentricity of the target reference orbit. Finally, we provide a validation of the obtained numerical solution of the previous ATV example, in the case **II ($\mathcal{L}_{1,1}$ norm)**.

Example 3.2.6 (ATV Example - *A posteriori* validation)**.** Between two impulses a polynomial with a validated error bound is computed for $x(\nu)$, $x'(\nu)$, $z(\nu)$ and $z'(\nu)$ as shown before. After each impulse, the new initial conditions are: for the positions, it suffices to rigorously evaluate the polynomials plus the error bound at the final time ν_f; the velocities are obtained as the interval sum between the impulse values and the rigorous evaluation of $x'(\nu_f)$ and $z'(\nu_f)$. This provides a validated enclosure of the position and velocity at each point and in particular at the end of the considered interval. In Table 3.8, we provide an enclosure of the final states obtained with this validated trajectory propagation procedure in function of different polynomial degrees d.

d	$x(\nu_f)$	$z(\nu_f)$	$\dot{x}(\nu_f)$	$\dot{z}(\nu_f)$
25	-100 + [-2.68**e1**, 2.68**e1**]	[-7.40**e0**, 7.40**e0**]	[-1.82**e-2**, 1.82**e-2**]	[-5.37**e-3**, 5.37**e-3**]
30	-100 + [-1.01**e-1**, 1.01**e-1**]	[-2.76**e-2**, 2.76**e-2**]	[-6.77**e-5**, 6.77**e-5**]	[-2.01**e-5**, 2.01**e-5**]
40	-100 + [-2.32**e-5**, 2.32**e-5**]	[-6.39**e-6**, 6.39**e-6**]	[-1.56**e-8**, 1.56**e-8**]	[-4.63**e-9**, 4.63**e-9**]
50	-100 + [-2.04**e-8**, 1.64**e-8**]	[-5.04**e-9**, 5.06**e-9**]	[-1.23**e-11**, 1.23**e-11**]	[-3.66**e-12**, 3.65**e-12**]

Table 3.8 – Final states obtained with rigorous trajectory propagation for the ATV example in function of the approximation degree d employed.

3.2.5 Other spin-off results

The presented work combines optimal control, computer algebra and approximation theory in the framework of aerospace domain. It features simplicity, speed and reliability. On the one hand, it makes use of state of the art linear/SDP solvers; on classical rendezvous mission examples, for accuracies of $\varepsilon = 10^{-4}$, no more than 10 iterations are necessary, which accounts for few milliseconds on a modern computer. On the other hand, error bounds provide guarantees that the accuracy requirements are met, both in terms of consumption and trajectory validation.

The numerical algorithm solution to the optimal problem presented, also provided us with valuable insight for solving other problems related or not to aerospace. At first, this allowed us to provide a closed-form solution for the elliptic out-of-plane rendezvous problem [C8] (which was not detailed here for the sake of brevity) and to get a better grasp on a more intricate geometric interpretation for the in-plane case, which is the subject of future works. Then, we applied mainly the same algorithm to solve the problem of fixed-time fuel-optimal trajectories with high-thrust propulsion in the vicinity of Lagrange points in the circular restricted three-body problem [C3].

Incidentally, we also observed that the SICP problem formulation presented above can be applied to a problem coming from the mathematical function implementation in machine, which was briefly discussed in Section 1.2. We believe that it is interesting to note the *cyclic* aspect of these results: we make use of validated computations to provide more reliable optimal control solutions, which in turn provide us with new solutions of well-known problems in computer arithmetic. Thus, we take a moment to briefly summarize this very recent result [C1] at the end of this chapter.

A spin-off result in mathematical functions implementation As shown in Example 1.2.1 for instance, the problem of evaluating a function f in FP arithmetic, usually boils down to two main steps:

- an **approximation** polynomial p is searched for, on some specific interval I, such that two main requirements are met: its coefficients are representable with a specified fixed precision format (usually, binary32, binary64, or an unevaluated sum of such formats) and the approximation error

is less than a target $\varepsilon_{\text{approx}}$, whether absolute $\|f-p\|_\infty \leqslant \varepsilon_{\text{approx}}$ or relative $\|(f-p)/f\|_\infty \leqslant \varepsilon_{\text{approx}}$. For that, efficient algorithms were developed in [36, 35]. In the simpler case of polynomials $p = \sum_{i=0}^{n} a_i t^i$ of given degree n, with real coefficients a_i, this is equivalent to the so-called *minimax* problem:

$$\min_{\substack{a_i\in\mathbb{R},\\ i\in[0..n]}} \max_{t\in I} |f(t)-p(t)|, \tag{P_{minimax}}$$

which can be solved by the Remez algorithm (see [35, 50] and references therein). This iterative algorithm has quadratic convergence and rather low complexity, since it involves solving a linear system of size $n+2$ at each step, together with numerically computing the extrema of $f-p$ over I.

- an efficient *evaluation* scheme for p is searched for; since after each addition or multiplication, rounding errors occur, one must ensure that the computed value $\tilde{p}$ satisfies $\|p-\tilde{p}\|_{\infty I} \leqslant \varepsilon_{\text{eval}}$ (or $\|(p-\tilde{p})/p\|_{\infty I} \leqslant \varepsilon_{\text{eval}}$) for a given threshold $\varepsilon_{\text{eval}}$. Heuristics presented in [129] extend the precision of the *important* coefficients, such that the evaluation error remains below $\varepsilon_{\text{eval}}$.

These two steps are usually independently considered, except for very small precisions or polynomial degrees, where an exhaustive search on the rounded coefficients is possible [208]. However, as explicitly mentioned in [36], *one would like to take into account the roundoff error that occurs during polynomial evaluation: getting the polynomial, with constraints on the size of the coefficients, that minimizes the total (approximation plus roundoff) error would be extremely useful*. We made some progress on this open question, by formulating a semi-infinite linear optimization problem (SIP) whose solution is the best polynomial with respect to the uniform norm of the sum of both errors.

The original optimization problem

$$\min_{\substack{a_i\in\mathbb{R},\\ i\in[0..n]}} \max_{t\in I} \left(|f(t)-p(t)| + |\tilde{p}(t)-p(t)|\right), \tag{P_{general}}$$

is firstly formulated as a convex SIP problem, using a linearized bound say, $\theta(\boldsymbol{a},t)$, for the evaluation error $|\tilde{p}(t)-p(t)|$:

$$\begin{array}{ll} \min\limits_{(\overline{a},\boldsymbol{a})\in\mathbb{R}^{n+2}} & \overline{a} \\ \text{s.t.} & |f(t)-\boldsymbol{\pi_0}(t)^T\boldsymbol{a}| + \theta(\boldsymbol{a},t) - \overline{a} \leqslant 0, \quad t\in I, \end{array} \tag{P'_{general}}$$

where we denote the monomial basis by $\boldsymbol{\pi_0}(t) = (1,\dots,t^n)^T$.

Note that in [C1], we provide an algorithm based on [164, 119, 201], which computes closed-form expressions for $\theta(\boldsymbol{a},t)$, of the form $\theta(\boldsymbol{a},t) = \sum_{i=1}^{n+1} |\boldsymbol{\pi}_i(t)^T\boldsymbol{a}|$. For instance, for Horner evaluation scheme one has

$$\begin{aligned} \boldsymbol{\pi}_1(t)^T &= (u, ut, \dots, ut^{n-1}, ut^n), \\ \boldsymbol{\pi}_2(t)^T &= (0, 2ut, \dots, 2ut^{n-1}, 2ut^n), \dots, \\ \boldsymbol{\pi}_n(t)^T &= (0, 0, \dots, 2ut^{n-1}, 2ut^n), \\ \boldsymbol{\pi}_{n+1}(t)^T &= (0, 0, \dots, 0, ut^n), \end{aligned}$$

where u is the unit roundoff (cf. Definition 1.1.3 i.e. $u = 2^{-53}$, for round-to-nearest in *binary64* precision).

An important observation is that, based on the above formula, (P'_{general}) can be formulated as a **linear** SIP, at the expense of a different index set Ω replacing the previous index set I. Here, the set of constraints of (P'_{general}) involving absolute values is replaced by as many linear constraints as required to represent all possible sign combinations. With the following definitions:

$$\begin{aligned} \boldsymbol{x} = (\overline{a},\boldsymbol{a}) \in \mathbb{R}^{n+2},\ \boldsymbol{z} = (1,0,\dots,0) \in \mathbb{R}^{n+2}, \\ \boldsymbol{\alpha}(t,\sigma_0,\dots,\sigma_n+1) = (1, \sigma_0\boldsymbol{\pi_0}^T(t) + \sum_{i=1}^{n+1} \sigma_i \boldsymbol{\pi_i}^T(t))^T \in \mathbb{R}^{n+2}, \\ \mathfrak{S} = \{-1,0,1\}^{n+2},\ \omega = (t,\sigma_0,\dots,\sigma_{n+1}) \in \Omega := I\times\mathfrak{S}, \end{aligned}$$

Problem (P'_{general}) is exactly the following linear SIP:

$$\begin{array}{ll}\min\limits_{\boldsymbol{x}\in\mathbb{R}^{n+2}} & \boldsymbol{z}^T\boldsymbol{x}\\ \text{s.t.} & \boldsymbol{\alpha}(\omega)^T\boldsymbol{x}\geqslant c(\omega),\quad \omega\in\Omega,\end{array}\tag{P}$$

where $c(\omega)=\sigma_0 f(t)$, Ω is a compact metric space and the function $g(\boldsymbol{x},\omega)=c(\omega)-\boldsymbol{\alpha}(\omega)^T\boldsymbol{x}\leqslant 0$ defining the feasible set is a continuous function from $\mathbb{R}^{n+2}\times\Omega$ into $\mathbb{R}$. Note that for $\mathfrak{S}'=\{-1,1\}\times\{0\}^{n+1}$ and $\Omega'=I\times\mathfrak{S}'\subseteq\Omega$, ($P_{\text{minimax}}$) is exactly retrieved as shown in the next example.

Example 3.2.7. For $n=5$, Problem (P_{minimax}) is:

$$\begin{array}{ll}\min\limits_{(\overline{a},\boldsymbol{a})\in\mathbb{R}^7} & \overline{a}\\ \text{s.t.} & (1,\sigma_0 1,\sigma_0 t,\ldots,\sigma_0 t^5)(\overline{a},a_0,a_1,\ldots,a_5)^T\geqslant\sigma_0 f(t),\\ & \sigma_0=\mp 1,\quad t\in I.\end{array}\tag{3.26}$$

while Problem (P'_{general}), assuming Horner evaluation is:

$$\begin{array}{ll}\min\limits_{(\overline{a},\boldsymbol{a})\in\mathbb{R}^7} & \overline{a}\\ \text{s.t.} & (1,\sigma_0+\sigma_1 u,(\sigma_0+\sigma_1 u+\sigma_2 2u)t,\ldots,(\sigma_0+\sigma_1 u+\ldots+\sigma_5 u)t^5)(\overline{a},a_0,a_1,\ldots,a_5)^T\geqslant\sigma_0 f(t),\\ & \sigma_0=\mp 1,\sigma_1=\mp 1,\ldots,\sigma_5=\mp 1,\quad t\in I.\end{array}\tag{3.27}$$

For Problem (P'_{general}), a convergent iterative algorithm similar to Algorithm 19 could be employed, which considers solving at each step a discretized version of Problem (P). In this case, a discretization (P_m) of (P) for a set $\boldsymbol{\omega}=\{\omega_1,\ldots,\omega_m\}\subseteq\Omega$ is the following linear program (LP):

$$\begin{array}{ll}\min\limits_{\boldsymbol{x}\in\mathbb{R}^{n+2}} & \boldsymbol{z}^T\boldsymbol{x}\\ \text{s.t.} & \boldsymbol{\alpha}(\omega_j)^T\boldsymbol{x}\geqslant c(\omega_j),\quad j=1,\cdots m.\end{array}\tag{P_m}$$

Let us recall that Algorithm 19 is based on the iterative principle that after solving at a certain step a discretized version (P_m), it finds the "most violating point ω not in the set $\boldsymbol{\omega}$, adds it to the discretized set, then repeats the solving process. However, for this case of linear SIP and under certain mild assumptions, we presented in [C1], an improvement of this algorithm, which performs an *exchange of points* at each step. Moreover, we proved that instead of solving an LP at each step, one can further simplify this approach to solving a system of linear equations, which is even more efficient. This can be seen as a generalization, in the above framework, of the Remez algorithm, which solves Problem (P_{minimax}). To prove its correctness, important discretization properties of linear SIP problems are employed in [C1], based for instance on the more recent survey [198] and revisiting older works [222, 45, 41, 40]. We state only the main result in what follows. We need however, to define similarly to Section 3.2.1, the topological *dual* of Problem (P), in the space $\mathcal{C}(\Omega)^*$ of signed Borel measures μ over $(\Omega,\mathcal{B}(\mathbb{R}^{n+2}))$ [89, Section 21.5]:

$$\begin{array}{ll}\max\limits_{\mu\succeq 0} & \int_\Omega c(\omega)\mathrm{d}\mu(\omega)\\ \text{s.t.} & \int_\Omega \boldsymbol{\alpha}(\omega)\mathrm{d}\mu(\omega)=\boldsymbol{z}.\end{array}\tag{D}$$

Its discretization (D_m) of (D) is:

$$\begin{array}{ll}\max\limits_{\substack{y_j\geqslant 0\\ j\in[1..m]}} & \sum\limits_{j=1}^m c(\omega_j)y_j\\ \text{s.t.} & \sum\limits_{j=1}^m y_j\boldsymbol{\alpha}(\omega_j)=\boldsymbol{z}.\end{array}\tag{D_m}$$

Denoting for a Problem (P), respectively $\mathrm{val}(P)$ and $\mathrm{Sol}(P)$, its optimal value and the set of its optimal solutions, one has the following inequalities:

- the *weak duality*, that is $\mathrm{val}(D) \leqslant \mathrm{val}(P)$ always holds;
- $\mathrm{val}(D_m) \leqslant \mathrm{val}(D)$ and $\mathrm{val}(P_m) \leqslant \mathrm{val}(P)$;
- $\mathrm{val}(D_m) = \mathrm{val}(P_m)$ (strong duality holds) provided that none of (P_m) or (D_m) is infeasible.

Conditions for having only equalities i.e., $\mathrm{val}(D_m) = \mathrm{val}(P_m) = \mathrm{val}(D) = \mathrm{val}(P)$ may be obtained by using conjugate duality theory as developed in [198, Theorems 2.2, 2.3 and 3.2].

Theorem 3.2.8. [198, Thm. 2.2, 2.3, 3.2] Under the assumptions:

A1 Ω is a compact metric space, $\boldsymbol{\alpha} : \Omega \to \mathbb{R}^d$ and $c : \Omega \to \mathbb{R}$ are continuous functions;

A2 $\mathrm{val}(P)$ is finite;

A3 *(Slater's condition)*: there exists $\boldsymbol{x}^\circ$ such that:

$$\boldsymbol{\alpha}(\omega)^T \boldsymbol{x}^\circ > c(\omega), \quad \text{for all } \omega \in \Omega; \tag{3.28}$$

A4 There exist $\omega_1, \ldots, \omega_d \in \Omega$ with $(\boldsymbol{\alpha}(\omega_1), \cdots, \boldsymbol{\alpha}(\omega_d))$ linearly independent such that:

$$\exists\, y_1, \ldots, y_d > 0, \quad \boldsymbol{z} = \sum_{j=1}^{d} y_j \boldsymbol{\alpha}(\omega_j), \tag{3.29}$$

the following statements are true:

(i) $\mathrm{Sol}(P) \neq \emptyset$ and bounded;

(ii) $\mathrm{Sol}(D) \neq \emptyset$ and bounded;

(iii) Problem (P) is reducible (that is $\mathrm{val}(P) = \mathrm{val}(P_m)$) to a Problem (P_m) with $m \leqslant d$;

(iv) $\mathrm{val}(P) = \mathrm{val}(D) = \mathrm{val}(P_m) = \mathrm{val}(D_m)$.

We proved in [C1] that Assumptions A1-A4 are satisfied for our Problem (P'_{general}), with $d = n + 2$ and therefore results (i)-(iv) of Theorem 3.2.8 apply. This allows for recasting the problem (P'_{general}) as the problem of finding the right discretization $\{\omega_1, \ldots, \omega_{n+2}\}$ such that item (iv) of Theorem 3.2.8 applies and to solve the associated (P_m) and/or (D_m).

Based on these results, an adaptation of an algorithm of [40], which can be seen as a generalization of the dual simplex algorithm for Problem (D) is as follows.

One finds at each iteration ℓ, the solution $\boldsymbol{y}^{(\ell)}$ of $(D^{(\ell)}_{n+2})$, with $\boldsymbol{\omega}^{(\ell)} = \{{\omega_j}^{(\ell)}\}_{j=1}^{n+2}$. Such a solution is a feasible (but not necessarily optimal) point of the dual Problem (D). Moreover, the objective value $\mathbf{z}^T \mathbf{x}^{(\ell)}$ of $(P^{(\ell)}_m)$ and (P) for the instance $\boldsymbol{x}^{(\ell)} := (\overline{a}^{(\ell)}, \boldsymbol{a}^{(\ell)})$ is equal to the objective value of (D) for the instance $\boldsymbol{y}^{(\ell)}$[2]. Hence, either $\boldsymbol{x}^{(\ell)}$ is a feasible solution of Problem (P) by Theorem 3.2.8, or it is an infeasible point of Problem (P). In the latter case, one of these constraints is replaced by a new one, indexed by ${\omega_*}^{(\ell)}$, in an exchange step in order to increase the objective value of the dual and works towards primal feasibility. In order to prove the convergence of this process, one needs an assumption on the dual solution, which always holds in the Remez algorithm. It is not proven in our current setting, but it never failed in practice.

Assumption 3.2.9. At each iteration ℓ, the solution $\boldsymbol{y}^{(\ell)}$ of the dual discretized Problem $(D^{(\ell)}_{n+2})$ is an interior point, that is $y_j^{(\ell)} > 0$ for all $j \in [1\,..\,n+2]$.

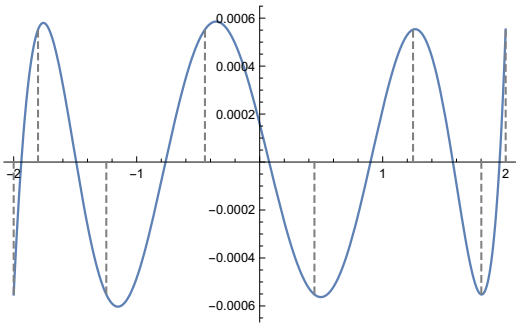
We conclude by illustrating this method on an academic example.

[2]The feasible set of (P) is included in the feasible set of $(P^{(\ell)}_m)$, for all ℓ.

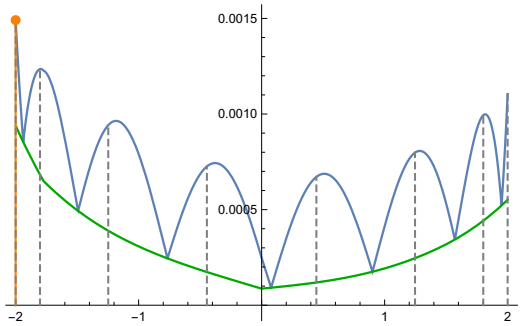
Example 3.2.10 (Airy function)**.** Let Ai over $I = [-2, 2]$, approximated by a polynomial of degree $n = 6$, evaluated using the Horner scheme with $u = 2^{-12}$. We fix a tolerance $\tau = 0.01$.

At iteration 0 (Figure 3.7), the points $t_j^{(0)}$ are initialized with the Chebyshev nodes and the signatures $\boldsymbol{\sigma_j}^{(0)}$ define a Remez-like system of linear equations on the coefficients of the polynomial (Figure 3.7d). Its solution $\boldsymbol{x}^{(0)} = (\overline{a}^{(0)}, \boldsymbol{a}^{(0)})$ defines a polynomial $p^{(0)}(t) = \boldsymbol{a}^{(0)T}\boldsymbol{\pi_0}(t)$, whose approximation error is depicted in Figure 3.7a. It exhibits quasi-equioscillations indicating that $\boldsymbol{p}^{(0)}$ is rather close to the degree-6 minimax approximation of Ai over I. However, the total error is more important near -2 and 2 (Figure 3.7b), due to the evaluation depicted in green. In particular, the algorithm detects the maximum error at $t_*^{(0)} = -2$ (in orange). Note that $t_1^{(0)}$ was already equal to -2, but $\omega_1^{(0)} \neq \omega_*^{(0)}$ since the signatures are different. To perform the exchange, the dual solution is needed (Figure 3.7c). It is a positive combination of Dirac measures supported on the finite set $\boldsymbol{\omega}^{(0)}$. Moving forward to iteration 6 (Figure 3.8), the total error is more balanced, though still not optimal. Both the signatures and the approximation error are now completely different from the Remez solution.

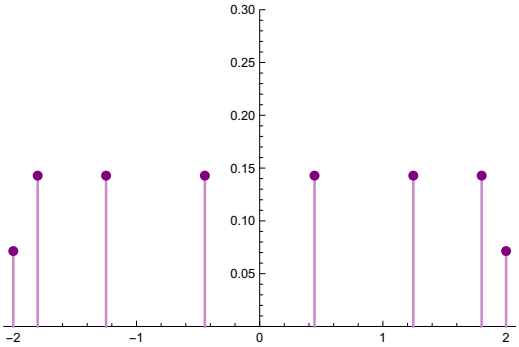
Eventually, the algorithm stops at iteration 9 (Figure 3.9). Indeed, the maximum total error $\overline{a}_*^{(9)}$ (in orange) is less than 1% higher than the error $\overline{a}^{(9)}$ over the discrete set $\boldsymbol{\omega}^{(9)}$. Note that the total error reaches its maximum at $n + 2 = 8$ points. This became possible by unbalancing the approximation error, namely reducing the amplitude of the oscillations near -2 and 2, at the cost of higher oscillations in the middle of I.



(a) Approximation error



(b) Total error



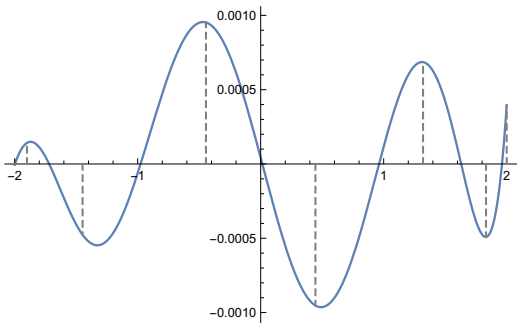
(c) Dual solution

t_j	σ_{j0}	σ_{j1}	σ_{j2}	σ_{j3}	σ_{j4}	σ_{j5}	σ_{j6}	σ_{j7}
-2.	+	o	o	o	o	o	o	o
-1.803	-	o	o	o	o	o	o	o
-1.247	+	o	o	o	o	o	o	o
-0.445	-	o	o	o	o	o	o	o
0.445	+	o	o	o	o	o	o	o
1.247	-	o	o	o	o	o	o	o
1.802	+	o	o	o	o	o	o	o
2.	-	o	o	o	o	o	o	o

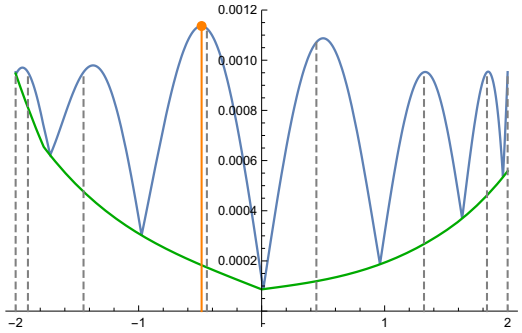
(d) Points and signatures

Figure 3.7 – Approximation of Ai over $[-2, 2]$: iteration 0

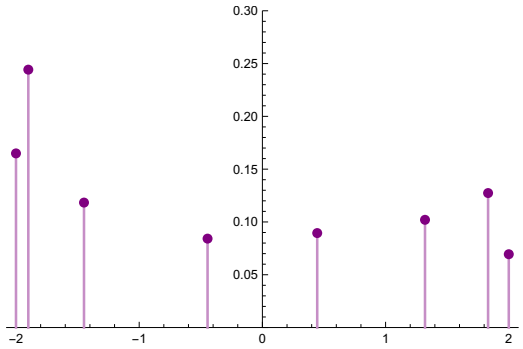
Other examples show that in practice, the total error obtained with our approach is better than the previous methods which work in two separate steps (approximation and evaluation). We also show that in some other cases, the minimax polynomial solution of problem (P_{minimax}) is very close to the solution of (P_{general}). Further experiments are necessary to assess the whole practical importance of this method.



(a) Approximation error



(b) Total error

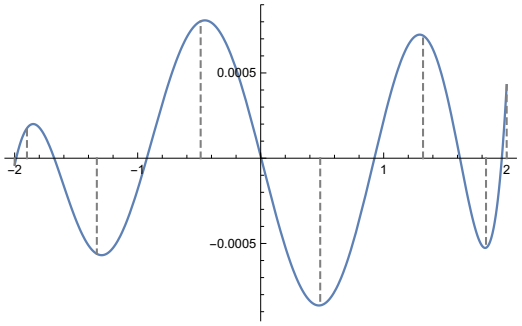


(c) Dual solution

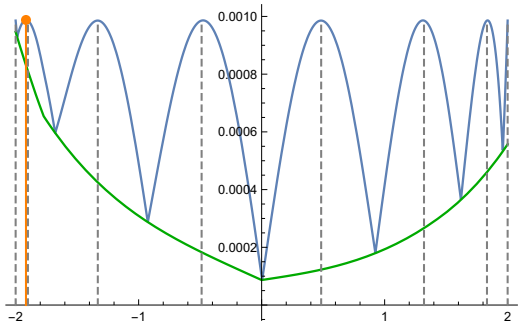
t_j	σ_{j0}	σ_{j1}	σ_{j2}	σ_{j3}	σ_{j4}	σ_{j5}	σ_{j6}	σ_{j7}
-2.	+	-	-	+	+	+	+	-
-1.9	-	-	-	+	+	+	+	-
-1.448	+	-	-	+	+	+	-	-
-0.445	-	o	o	o	o	o	o	o
0.445	+	o	o	o	o	o	o	o
1.32	-	-	+	+	-	-	+	-
1.832	+	-	-	+	-	-	+	-
2.	-	-	-	+	-	-	+	-

(d) Points and signatures

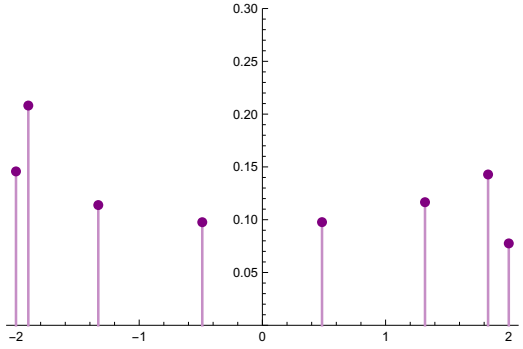
Figure 3.8 – Approximation of Ai over $[-2, 2]$: iteration 6



(a) Approximation error



(b) Total error



(c) Dual solution

t_j	σ_{j0}	σ_{j1}	σ_{j2}	σ_{j3}	σ_{j4}	σ_{j5}	σ_{j6}	σ_{j7}
-2.	+	-	-	+	+	+	+	-
-1.9	-	-	-	+	+	+	+	-
-1.332	+	-	-	+	+	+	-	-
-0.488	-	-	-	+	+	+	-	-
0.484	+	-	+	+	-	-	+	-
1.32	-	-	+	+	-	-	+	-
1.832	+	-	-	+	-	-	+	-
2.	-	-	-	+	-	-	+	-

(d) Points and signatures

Figure 3.9 – Approximation of Ai over $[-2, 2]$: iteration 9

All in all, the examples presented in this Chapter show that numerically reliable and efficient solutions can be obtained when generic optimization frameworks are rethought and tuned from an approximation and/or computer algebra perspective. This encouraging trend will be further discussed in Chapter 4.

Chapter 4

Research perspectives

My long term research goal is to bring more reliable computations in the field of optimal control theory and aerospace applications (e.g. spacecraft rendezvous, in-orbit servicing, collision risk mitigation), where several difficult, but very promising challenges are left to be addressed. Several works [182, 107, 108, 103, 137] in this sense considered the application of interval analysis in the field of control. However, they make extensive use of interval branch and bound methods and little or no use at all of powerful symbolic algorithms existing nowadays.

For that, I consider four main challenges:

- C1: Efficient Computation/Evaluation/Manipulation of Rigorous Polynomial Approximations (RPAs);
- C2: Extension and Integration of RPAs in the framework of Optimal Control Theory;
- C3: Applications to Guidance, Navigation and Control of Spacecraft;
- C4: Computer Arithmetic aspects pertaining to C1– C3.

The first objective is to consider the symbolic-numeric aspect: modern computer algebra algorithms (rooted in commutative and differential algebra) are employed via approximation theory (in suitable functional spaces) to obtain efficient approximations and analytic error bounds. Secondly, at subsequent numerical levels, roundoff errors are handled, especially in the context of extended precision or embedded computations.

4.1 RPAs: Rigorous Polynomial Approximations

As discussed in Chapter 2, at the symbolic-numeric level, we aim not only to compute *approximations*, but also *enclosures* of errors. The width of such an enclosure gives a direct quality measurement of the computation, and can be used to adaptively improve the calculations at run-time. Polynomials are one of the most efficient and exploited ways for computing in this setting. To improve their reliability, we worked on *Rigorous Polynomial Approximations* (RPA): a polynomial approximation together with rigorous error bounds (Section 2.4). Broadly speaking, the idea of working with polynomial approximations instead of functions is analogous to using floating-point arithmetic instead of real numbers [66, 210]: various *generalized Fourier series*, including Chebyshev series, play the role of floating-point numbers. However, with RPAs, one comprises a function space counterpart of interval arithmetic by providing rigorous truncation error bounds. The main appeal of this approach is the ability to solve functional equations rigorously using enclosure methods [110, 160]. We discussed, in Chapter 2, several recent results based on Chebyshev series and *D-finite functions*. My goal is to further exploit these properties in the context of RPAs for ordinary differential equations (ODEs). Namely, the objective is the efficient computation, manipulation and evaluation of rigorous polynomial approximations as solutions of ODEs.

Probably the most used self-validating method in the study of nonlinear algebraic, integral or differential equations is based on the *contraction mapping principle*. This is a fixed point theorem which guarantees that a contraction mapping of a complete metric space to itself has a unique fixed point. We already made use of this theorem (see Thm. 2.4.8) in the general setting of quasi-Newton validation, in Section 2.4.2. This corresponds to *a posteriori* validation: usually, one finds a good numerical solution p and then verifies that the image under the contracting mapping of a ball centered at p and of *suitable* radius R is invariant. This implies that the actual solution is enclosed in the *ball* $\mathcal{B}(p, R)$. This theorem can also be used in a constructive way: we can obtain the fixed point as the limit of an iteration scheme defined by repeated images under the mapping of an arbitrary starting point in the space. When implemented numerically using *ball* based computations, one has to pay special attention to the well-known issues of overestimation or wrapping effect. Hence, we are interested in efficiently manipulating and implementing operations with such *closed balls* having roughly speaking, as center a truncated series and with radius a certain norm for the error. A key element for this extension is that, as shown in [15, Chap.8], remarkable properties of recurrences satisfied by series coefficients of D-finite functions hold for such generalized Fourier series. This can be made constructive through the use of Ore-algebras as shown in Chapter 2.

Application to certified quadratures: infinitesimal Hilbert's 16th problem An ongoing work, in collaboration with F. Bréhard, N. Brisebarre and W. Tucker, concerns the certified evaluation of certain Abelian integrals involved in a computer-assisted proof of a new lower bound on the Hilbert number for quartic systems. This is related to the second part of Hilbert's 16th problem [97], which asks about the maximum number and location of limit cycles of a planar polynomial vector field of degree d. Solving this problem even for the case $d = 2$ seems to be out of reach at the present state of knowledge. A restricted version of Hilbert's 16th problem, known as the infinitesimal Hilbert's 16th problem, asks for the number of limit cycles that can bifurcate from a perturbation of a Hamiltonian system [10]. These limit cycles are related to the zeros of certain Abelian integrals on a union of compact intervals, via the so-called Poincaré-Pontryagin theorem. Based on RPAs representations, we are building an integration routine that is both fast (in order to explore a huge field of possible systems, together with the attached compact intervals) and certified (the computations are highly unstable). Moreover, for obtaining continuous approximations of these integrals in function of certain parameters, we make use of algebraic properties of D-finite functions and Laplace transform, in the same manner as presented in Chapter 2.

RPAs in generalized Fourier series Spectral and collocation methods for numerical approximations of ODEs are current in literature [32, 209, 131] and were recently applied in the aerospace domain [219]. However, obtaining effective rigorous accuracy bounds on the solutions is less developed, especially from a generic algorithmic point of view. We aim to generalize our validated Newton-like method (see Section 2.4) to an efficient computer algebra-based algorithm for validated solutions of nonlinear ODEs. An important objective in this context is to extend our approach to generalized Fourier series. Truncated Chebyshev Series or Chebyshev interpolation polynomials are often used in practice because they are near-best approximations with respect to the uniform norm. Also, in the space of square integrable functions, truncated Chebyshev series are best approximations with respect to the attached $\mathcal{L}^2(1/\sqrt{1-x^2})$ norm. However, for different convergence domains or due to some intrinsic properties of the problems considered, it is more suitable to use other truncated series, with different norms: Legendre polynomials are preferred for pseudo-spectral optimal control [187], Hermite for polynomial chaos propagation [220] with applications in spacecraft trajectories design [168], Fourier for the rigorous verification of several invariant objects arising in dynamical systems, like time periodic orbits for Kuramoto-Sivashinsky equations, Bessel series when dealing with specific boundary conditions [9]. In general, let us consider the Banach space of $\mathcal{L}^2$ functions defined on some some set X, and the attached $||\cdot||_2$ norm and denote $\mathcal{B}(p, R) = \{f : ||f - p||_2 \leqslant R\}$ the closed ball of center p and radius R. This is a complete metric subspace. These kinds of *balls* or RPAs are intensively used objects in most rigorous computing methods. A short term goal is to provide efficient algorithms, similar to those presented in Section 2.4.1, which provide recurrences satisfied by the coefficients of p, as well as accurate evaluation schemes for it. Then, *a posteriori* validated error bounds could be obtained, by setting the problem in the suitable Banach space and applying Picard's iteration or quasi-Newton contracting map.

Multivariate RPAs Finally, let us consider the extension to multivariate orthogonal series. Classical

univariate Chebyshev polynomials are among the most important building blocks in approximation theory and are also frequently used for multidimensional approximations. Besides tensor products (like the one used for bivariate approximations in Chebfun), other less classical extensions for multivariate orthogonal polynomials were developed [83]. From an algorithmic point of view, the main important connection to be made is between multivariate orthogonal series and holonomic D-modules, which are multivariate equivalent of D-finite functions i.e., their partial derivatives generate a vector space of finite dimension on the field of rational functions (cf. Section 2.1.2).

From the approximation point of view, an important issue here is to maintain good approximation properties with respect to the considered norm. In the univariate case, both truncated Chebyshev series and Chebyshev interpolation polynomials are near-best and the connection between them is classical, see for example [R3, Chap. 4][211]. In the multivariate case, however, there is no explicit solution for finding *optimal* or *near-optimal* interpolation points with respect to the uniform norm of the interpolation error. One solution is to find points that come from zeros of quasi-orthogonal polynomials in several variables [65] which would help us to make a connection similar to the univariate case, between orthogonal series and polynomial interpolation through minimal cubatures formulas. Moreover, in [190], multivariate Chebyshev polynomials on triangular domains are proposed. Their numerical results on some quadrature problems seem to indicate that this choice is also suitable for RPAs.

Efficient numerical evaluation of RPAs As discussed in Section 2.2, when evaluating RPAs in finite precision, catastrophic cancellation i.e., subtraction of terms of same order of magnitude which results in important precision loss can appear. The problematic phenomenon, for truncated convergent series evaluation, is that the absolute values of the terms involved, start by increasing before decreasing towards 0, while their signs are not identical. In the classical case of the error function $\mathrm{erf}(x)$, this problem can be circumvented by factoring out an exponential. On these lines, we obtained in Section 2.2, a new result for some cases of truncated convergent power series to be evaluated on the real line. Although very useful, our approach remains ad-hoc for the moment and more work is necessary for this method to be efficient in general. Efficient generic algorithms for reducing such cancellation effects are to be found. Moreover, the problem of extending this method for generalized Fourier series evaluation is open.

We already have several promising hints for generalizing our algorithm, which combine both symbolic and numeric aspects in complex analysis: the asymptotic behavior of series coefficients is connected to their integral representation [228], which in turn, can be efficiently numerically evaluated with saddle-point methods [68, Chapter VIII]. A different angle for seeing our pre-conditioning method is that it represents a first result in extending Polya's positivstellensatz [175] (an algebraic description of positive polynomials) to the case of analytic functions.

4.2 Extension and Integration of RPAs in the framework of Optimal Control

In Optimal Control, the objective is to find an optimal control input function, subject to some performance measure and the system dynamics, which are complex in general. We deal with nonlinear ODEs or with a differential-algebraic equations. Moreover, uncertainties are often present in dynamics or state measurements. Nowadays, the great majority of *practical* i.e. computer-based methods in optimal control are heavily based on standard linear algebra routines and floating-point arithmetic [19], while the development of computer algebra and symbolic-numeric techniques in this field is marginal.

Thus, optimal control is a natural outlet for reliable computation: symbolic-numeric approximations are to be declined in the context of both deterministic (direct and indirect) and stochastic optimal control. This implies either the integration of RPAs in the existing algorithms or more often the adaptation of these algorithms towards RPAs in order to improve reliability and speed.

We distinguish in (deterministic) optimal control two distinct numerical approaches: the indirect and direct methods. The first class consists in writing (usually necessary) optimality conditions based on the Pontryagin maximum principle [37] and thus posing a Two Point Boundary Value Problem (TPBVP) which can be nonlinear, and solving it numerically with shooting methods for instance. The second

approach is to discretize and thus directly translate the initial problem into a nonlinear programming problem with an appropriate parametrization. Only an approximation of the initial problem is solved in the latter case. In both cases, we are interested in validated computation methods to guarantee the obtained numerical solutions. Beside the development of new algorithms for validated solutions, an important issue is to certify the computation of optimal trajectories (especially for space objects), particularly for embedded control systems (trajectories are computed on on-board embedded computers which allows for increasing the autonomy for space missions).

Direct optimal control This basically consists in transforming the problem to a nonlinear finite-dimensional optimization problem: depending on the transcription approach, both the control and the state variables may be discretized. Many variants exist [19] and among them, we are interested in improving the speed and reliability of pseudo-spectral and collocation methods. For example, pseudo-spectral Legendre method has already been used in space applications like the zero-propellant maneuver of International Space Station in 2006, or the minimum-time rotational maneuver performed in orbit by TRACE telescope in 2010 [187], while variants based on Chebyshev or other Galerkin discretizations [30] also exist. However, as mentioned in [187], "at the theoretical level, stronger convergence theorems are required for these methods". In addition, the efficiency of computations has to be improved, driven by a demand to perform mission planning at a faster and faster pace [187]. My claim is that RPAs-based methods are efficient and in the same time provide convergence proofs: a validated error bound is a mathematical certificate that the numerical solution is sufficiently close to the real one.

Indirect optimal control The main idea is to reduce the problem to a Two Point Boundary Value Problem (TPBVP) with the help of Pontryagin maximum principle [37]. Using shooting methods consists mainly in obtaining a solution to a Cauchy problem where the initial value is unknown (formal) and then use Newton method (or its variations) in order to obtain this value as the zero of the function which relies the boundary constraints. When a multiple shooting method is used, a subdivision of the time interval is considered and the problem is reduced to finding the zeros of a function which is defined on a vectorial space of higher dimension (proportional to the number of subdivisions). My intuition is that computer algebra methods for RPAs could provide solutions to Cauchy problems with formal initial conditions. Then, interval Newton method (and higher order methods for zero finding) [229] will be used to obtain an enclosure of the solution. Validated Newton-like methods are very versatile and are based on powerful fix-point theorems which can be readily adapted to provide enclosures of the solution. RPAs and validated Newton-like methods can be readily adapted to provide solutions enclosures for TPBVP.

Stochastic optimal control It is often employed when uncertainty and stochastic effects play a significant role in the dynamics of the system. A very recent approach [31] is based on Polynomial Chaos [225]. First introduced by Wiener, Polynomial Chaos decomposes stochastic processes into a convergent series of Hermite polynomials in a Gaussian random variable. So-called Wiener-Askey Polynomial or generalized Polynomial Chaos (gPC), extends this idea to various distributions modeled with orthogonal polynomials in the corresponding Hilbert spaces and was successfully used for analyzing stability or linear quadratic control of stochastic systems or for spacecraft orbit propagation under Gaussian type uncertainties [220, 105]. Symbolic-numeric algorithms employed for the computation of polynomial coefficients in RPAs are easily adapted to this framework and could drastically improve the computation time and numerical efficiency of these methods.

4.3 Reliable Computations for Guidance, Navigation and Control of Spacecraft

Previously mentioned techniques are to be applied for the guidance, navigation and control of spacecraft when uncertain models are used. Firstly, new symbolic-numeric spectral methods are used in the context of stochastic optimal control for the computation of optimal collision avoidance maneuvers in the case of mega-constellations. Secondly, we consider the reliable computation of optimal trajectories: either ground

mission planning tools or embedded control systems (on-board computation for increased autonomy). Specific applications include spacecraft rendezvous, in-orbit servicing, collision risk mitigation.

My third major objective is to provide effective solutions for concrete space projects: both speed and reliability are at stake.

The risk assessment and collision avoidance strategies for the case of mega-constellations is one of our practical short-term objectives.

As briefly mentioned in Section 3.1.2, the general framework of the problem of collision avoidance was already studied in our report [R2], where the objective is to give an exact and rigorous mathematical modeling of the problem of the computation of the probability of collision for multiple encounters. Our approach follows the general convex optimization-based framework for analysis and optimal control of dynamical systems proposed in [92, 112, 204, 125, 142]. This is based on (1) the formulation of an infinite-dimensional linear programming problem in the cone of nonnegative measures (also known as *lifting of nonlinear dynamics via Liouville's equation* and (2) the theory of non-negative polynomials and so-called Lasserre hierarchy of relaxations. In this general framework, finite dimensional relaxations outer- or inner- approximate by polynomial sublevel sets various quantities of interest in controlled dynamical systems, like the region of attraction [113, 92, 112] or the maximal positively invariant set [166].

While the theoretical convergence of these approximants is proved, in practice the underlying methods are numeric, the corresponding solvers work in the ill-conditioned monomial basis and no certified counterpart was developed except for very small instances [93].

In our practical case, our numerical experiments show that firstly, the dimension of the general problem is currently prohibitive for existing semi-definite solvers. Secondly, several examples show that numerical results in low dimension do not achieve a good accuracy. Recent works suggest that formulating the Moment-Sum-of-Squares problem with respect to orthogonal bases results in better numerical stability [73]. In this sense, for stochastic solutions of an ordinary differential equation which describes the propagated orbit, orthogonal expansions are frequently used in the so-called *polynomial chaos expansions* [105, 220].

We intend to use symbolic-numeric methods for obtaining and manipulating RPAs in the practical solving of the linear optimization problem of measures. We aim for a three-fold improvement: (1) algebraic properties (like recurrences satisfied by moments of D-finite functions) reduce the number of variables in the optimization problem to solve; (2) the better conditioning of orthogonal bases improves numerical solving, thus allowing for higher order relaxations (in finite precision solvers); (3) a preliminary study can be made on how to efficiently propose solutions which are rigorous and certified.

We believe that with the help of RPAs algorithms, *a posteriori* uncertainty orbit propagation can be rendered more efficient and reliable (an important challenge is that the relative dynamics between debris and a cluster of satellites is nonlinear).

Spacecraft rendezvous has become a key technology raising relevant open control issues. Formation flight (PRISMA), on-orbit satellite servicing or supply missions require adequate rendezvous planning tools. A main challenge is to achieve autonomous far range rendezvous on elliptical orbits while preserving optimality in terms of fuel consumption, which defines a minimum-fuel optimal control problem. In Section 3.2, we developed a new numerical convergent algorithm for solving this problem in a linearized fixed-time framework. Validation of trajectories is done *a posteriori* with RPAs for LODEs, which was presented in Section 2.4. Our goal is to extend this problem to a framework that is more general than linearized Keplerian dynamics. Solutions of more accurate ODE models, which take into account orbital perturbations for example, are to be obtained with RPAs. This will allow for developing new tractable and guaranteed algorithms. Especially in the context of increased satellite autonomy, this could improve on our work [J1] of embedding model predictive control algorithms on low-consumption processors specifically certified for spaceflight.

4.4 Computer Arithmetic related aspects

Once Computer Algebra provides the canvas for RPAs (e.g., a recurrence relation satisfied by the polynomial coefficients), evaluation takes place in the realm of numerics. Usually, the computations for

evaluating the polynomial (e.g., by unrolling the recurrence) are done in standard floating-point double precision (53 bits) for efficiency. Thus, roundoff errors appear at the computer arithmetic level and need to be handled. Some techniques for avoiding catastrophic loss of precision were presented in Section 2.2. When needed, extended precision as presented in Chapter 1 or an interval arithmetic built on it, can be used for a small overhead. Our goal is to put CAMPARY at the core of reliable numerical code on RPAs and more generally, make it a natural tool for symbolic-numeric computations. Note that highly parallel subsystems (like GPUs) have already been used for embedded pseudo-spectral optimal control for spacecraft [187].

This would provide a complete top-down approach for optimal control algorithms implementations by ensuring that the Computer Arithmetic tools are adequate.

In this context, we will focus on several specific tasks which extend our previous developments. So far, we focused on the tight and rigorous error bound analysis of basic arithmetic operations in extended precision obtained via FP expansions. We intend to provide similar improvements in larger calculations.

Error analysis for fast transforms. The Fast Fourier Transform (FFT) and related algorithms (e.g., fast cosine transforms) play a central role in large precision arithmetic, since they are used in fast polynomial and integer multiplication algorithms [194]. Performing an FFT-based multiplication requires a careful error control: all intermediate calculations are done in FP arithmetic, and since the final, *exact*, results (e.g., the coefficients of the polynomial product) are integers, we get them by rounding to the nearest integer the *computed* results. Hence the error on the computed results must be less than $1/2$. Many error analysis results of the FFT [181, 96, 172, 174] bound the *relative mean-square error*. For our applications to high-precision arithmetic, we also need bounds in terms of *infinity norm* [91]. Recently, in an already submitted work [R1], we have improved the bound [91] and built *bad input cases* for which the attained error is around one eighth of the bound. A short term objective is to extend these results to other similar transforms.

Fast and accurate algorithms for complex and quaternion floating-point arithmetic. Similarly to our developments in Chapter 1, we will consider complex operations in FP arithmetic. Currently, complex arithmetic primitives are in general not as accurate or efficient as basic arithmetic operations and are frequently subject to *spurious under/overflow* (an intermediate calculation underflows or overflows, making the *computed* final result irrelevant, although the *exact* result is in the domain of the representable numbers). For these reasons, complex arithmetic provided by programming languages is seldom used. Arithmetic operations on quaternions as well as conversion algorithms to/from rotation matrices are prone to similar numerical issues as the complex numbers, as recently illustrated in [193]. Quaternions represent rotations in $\mathbb{R}^3$ in a non-singular way, which makes them useful in computer graphics, drone and aerospace vehicle control. We aim at providing accurate, reliable, and fast algorithms for complex and quaternion arithmetic. We will frequently use error-free transforms (which were recalled in Section 1.2) to enable faithful evaluation. The efficiency of this library will be assessed on concrete aerospace applications mentioned above.

Exhaustive List of Publications

Books

[B1] J.-M. Muller, N. Brunie, F. de Dinechin, C.-P. Jeannerod, M. Joldes, V. Lefèvre, G. Melquiond, N. Revol, and S. Torres. *Handbook of Floating-Point Arithmetic*. Birkhäuser, 2018.

International peer-reviewed journals

[J1] P. R. Arantes Gilz, M. Joldes, C. Louembet, and F. Camps, "Stable Model Predictive Strategy for Rendezvous Hovering Phases Allowing for Control Saturation," *Journal of Guidance, Control, and Dynamics*, p. 42, 2019. Accepted for publication, https://hal.archives-ouvertes.fr/hal-01678768.

[J2] F. Bréhard, N. Brisebarre, and M. Joldeş, "Validated and numerically efficient Chebyshev spectral methods for linear ordinary differential equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 44, no. 4, p. 44, 2018. https://hal.archives-ouvertes.fr/hal-01526272/.

[J3] M. Joldes, J.-M. Muller, and V. Popescu, "Tight and rigorous error bounds for basic building blocks of double-word arithmetic," *ACM Transactions on Mathematical Software (TOMS)*, vol. 44, no. 2, p. 15, 2017. https://hal.archives-ouvertes.fr/hal-01351529v3.

[J4] A. Benoit, M. Joldes, and M. Mezzarobba, "Rigorous uniform approximation of D-finite functions using Chebyshev expansions," *Mathematics of Computation*, vol. 86, no. 305, pp. 1303–1341, 2016. https://hal.archives-ouvertes.fr/hal-01022420.

[J5] R. Serra, D. Arzelier, M. Joldes, J.-B. Lasserre, A. Rondepierre, and B. Salvy, "Fast and accurate computation of orbital collision probability for short-term encounters," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 5, pp. 1009–1021, 2016. https://hal.archives-ouvertes.fr/hal-01132149/.

[J6] M. Joldes, O. Marty, J.-M. Muller, and V. Popescu, "Arithmetic algorithms for extended precision using floating-point expansions," *IEEE Transactions on Computers*, vol. 65, pp. 1197–1210, April 2016. https://hal.archives-ouvertes.fr/hal-01111551v2.

[J7] M. Joldes, V. Popescu, and W. Tucker, "Searching for sinks for the Hénon map using a multiple-precision GPU arithmetic library," *SIGARCH Comput. Archit. News*, vol. 42, no. 4, pp. 63–68, 2014. https://hal.archives-ouvertes.fr/hal-00957438.

[J8] S. Chevillard, J. Harrison, M. Joldes, and C. Lauter, "Efficient and accurate computation of upper bounds of approximation errors," *Theoretical Computer Science*, vol. 16, no. 412, pp. 1523–1543, 2011.

Publications in the peer-reviewed proceedings of international conferences

[C1] D. Arzelier, F. Bréhard, and M. Joldes, "Exchange algorithm for evaluation and approximation error-optimized polynomials," in *Computer Arithmetic (ARITH), 2019 IEEE 26th Symposium on*, 2019. To appear, https://hal.archives-ouvertes.fr/hal-02006606/.

[C2] F. Bréhard, M. Joldes, and J.-B. Lasserre, "On moment problems with holonomic functions," in *Symbolic and Algebraic Computation, International Symposium, ISSAC 2019, Beijing, China, July 15-18, 2019, Proceedings*, July 2019. To appear, https://hal.archives-ouvertes.fr/hal-02006645.

[C3] R. Serra, D. Arzelier, F. Bréhard, and M. Joldes, "Fuel-optimal impulsive fixed-time trajectories in the linearized circular restricted 3-body-problem," in *IAF Astrodynamics Symposium in 69TH international astronautical congress*, pp. 1–9, 2018. https://hal.archives-ouvertes.fr/hal-01830253.

[C4] F. Camps, P. R. A. Gilz, M. Joldes, and C. Louembet, "Embedding a SDP-based control algorithm for the orbital rendezvous hovering phases," in *Proceedings of tof IEEE International Conference on Integrated Navigation Systems, Jun 2018, Saint Petersburg, Russia. 10.23919/ICINS.2018.8405931*, pp. 1–7, IEEE, 2018. https://hal.laas.fr/hal-01729956.

[C5] M. Joldes, J.-M. Muller, and V. Popescu, "Implementation and performance evaluation of an extended precision floating-point arithmetic library for high-accuracy semidefinite programming," in *Computer Arithmetic (ARITH), 2017 IEEE 24th Symposium on*, pp. 27–34, IEEE, 2017. https://hal.archives-ouvertes.fr/hal-01491255/.

[C6] S. Boldo, M. Joldes, J.-M. Muller, and V. Popescu, "Formal verification of a floating-point expansion renormalization algorithm," in *International Conference on Interactive Theorem Proving*, pp. 98–113, Springer, 2017. https://hal.archives-ouvertes.fr/hal-01512417/document.

[C7] P. R. A. Gilz, M. Joldes, C. Louembet, and F. Camps, "Model predictive control for rendezvous hovering phases based on a novel description of constrained trajectories," in *IFAC-PapersOnLine*, vol. 50, pp. 7229–7234, Elsevier, 2017. https://hal.laas.fr/hal-01484764.

[C8] D. Arzelier, F. Bréhard, N. Deak, M. Joldes, C. Louembet, A. Rondepierre, and R. Serra, "Linearized impulsive fixed-time fuel-optimal space rendezvous: A new numerical approach," in *Proceedings of the 20th IFAC Symposium on Automatic Control in Aerospace, 21-25 August, 2016, Sherbrooke, Quebec, Canada*, vol. 49, pp. 373–378, 2016. https://hal.archives-ouvertes.fr/hal-01275427.

[C9] S. Collange, M. Joldes, J.-M. Muller, and V. Popescu, "Parallel floating-point expansions for extended-precision GPU computations," in *Proceedings of ASAP 2016: The 27th Annual IEEE International Conference on Application-specific Systems, Architectures and Processors, 6-8 July 2016, London, England*, pp. 139–146, IEEE, 2016. https://hal.archives-ouvertes.fr/hal-01298206.

[C10] F. Bréhard, N. Brisebarre, and M. Joldes, "A new efficient algorithm for computing validated Chebyshev approximations solutions of linear differential equations," in *SCAN 2016: 17th International Symposium on Scientific Computing,Computer Arithmetic and Verified Numerics, Uppsala, Sweden, Sept. 2016*, pp. 41–43, 2016.

[C11] M. Joldes, J.-M. Muller, V. Popescu, and W. Tucker, "Campary: Cuda multiple precision arithmetic library and applications," in *Mathematical Software – ICMS 2016: 5th International Conference, Berlin, Germany, July 11-14, 2016, Proceedings* (G.-M. Greuel, T. Koch, P. Paule, and A. Sommese, eds.), (Cham), pp. 232–240, Springer International Publishing, 2016. https://hal.archives-ouvertes.fr/hal-01312858.

[C12] R. Serra, D. Arzelier, M. Joldes, J.-B. Lasserre, A. Rondepierre, and B. Salvy, "A new method to compute the probability of collision for short-term space encounters," in *Astrodynamics Specialist Conference*, pp. 1–7, Aug 2014.

[C13] R. Serra, D. Arzelier, M. Joldes, and A. Rondepierre, "Probabilistic collision avoidance for long-term space encounters via risk selection," in *Advances in Aerospace Guidance, Navigation and Control*, pp. 679–698, Springer, Dec 2015. https://hal.archives-ouvertes.fr/hal-01995936/document.

[C14] M. Joldes, J. Muller, and V. Popescu, "On the computation of the reciprocal of floating point expansions using an adapted Newton-Raphson iteration," in *IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors, ASAP 2014, Zurich, Switzerland, June 18-20, 2014*, pp. 63–67, IEEE, 2014. https://hal.archives-ouvertes.fr/docs/00/95/73/79/PDF/invNewton.pdf.

[C15] N. Brisebarre, M. Joldes, É. Martin-Dorel, M. Mayero, J. Muller, I. Pasca, L. Rideau, and L. Théry, "Rigorous polynomial approximation using Taylor Models in Coq," in *NASA Formal Methods - 4th International Symposium, NFM 2012, Norfolk, VA, USA, April 3-5, 2012. Proceedings* (A. Goodloe and S. Person, eds.), vol. 7226 of *Lecture Notes in Computer Science*, pp. 85–99, Springer, 2012.

[C16] N. Brisebarre, M. Joldes, P. Kornerup, É. Martin-Dorel, and J. Muller, "Augmented precision square roots and 2-d norms, and discussion on correctly rounding sqrt(xˆ2+yˆ2)," in *20th IEEE Symposium on Computer Arithmetic, ARITH 2011, Tübingen, Germany, 25-27 July 2011* (E. Antelo, D. Hough, and P. Ienne, eds.), pp. 23–30, IEEE Computer Society, 2011.

[C17] N. Brisebarre and M. Joldes, "Chebyshev interpolation polynomial-based tools for rigorous computing," in *Symbolic and Algebraic Computation, International Symposium, ISSAC 2010, Munich, Germany, July 25-28, 2010, Proceedings* (W. Koepf, ed.), pp. 147–154, ACM, 2010.

[C18] F. de Dinechin, M. Joldes, and B. Pasca, "Automatic generation of polynomial-based hardware architectures for function evaluation," in *21st IEEE International Conference on Application-specific Systems Architectures and Processors, ASAP 2010, Rennes, France, 7-9 July 2010* (F. Charot, F. Hannig, J. Teich, and C. Wolinski, eds.), pp. 216–222, IEEE, 2010.

[C19] F. de Dinechin, M. Joldes, B. Pasca, and G. Revy, "Multiplicative square root algorithms for FPGAs," in *International Conference on Field Programmable Logic and Applications, FPL 2010, August 31 2010 - September 2, 2010, Milano, Italy*, pp. 574–577, IEEE, 2010.

[C20] S. Chevillard, M. Joldes, and C. Q. Lauter, "Sollya: An environment for the development of numerical codes," in *Mathematical Software - ICMS 2010, Third International Congress on Mathematical Software, Kobe, Japan, September 13-17, 2010. Proceedings* (K. Fukuda, J. van der Hoeven, M. Joswig, and N. Takayama, eds.), vol. 6327 of *Lecture Notes in Computer Science*, pp. 28–31, Springer, 2010.

[C21] S. Chevillard, M. Joldes, and C. Q. Lauter, "Certified and fast computation of supremum norms of approximation errors," in *19th IEEE Symposium on Computer Arithmetic, ARITH 2009, Portland, Oregon, USA, 9-10 June 2009* (J. D. Bruguera, M. Cornea, D. D. Sarma, and J. Harrison, eds.), pp. 169–176, IEEE Computer Society, 2009.

Publications in the peer-reviewed proceedings of national conferences

[NC1] M. Joldes, V. Popescu, and W. Tucker, "Searching for sinks of Hénon map using a multiple-precision GPU arithmetic library," in *Forum des Jeunes Mathématicien-ne-s*, p. 6, Nov 2013.

[NC2] M. Joldes, "When a logarithm is a misspelled algorithm," in *Proceedings of the Association Femmes et mathématiques*, Sept. 2010.

[NC3] F. de Dinechin, M. Joldes, B. Pasca, and G. Revy, "Racines carrées multiplicatives sur FPGA," in *SYMPosium en Architectures nouvelles de machines (SYMPA)*, (Toulouse), Sept. 2009.

Articles under submission, research reports

[R1] N. Brisebarre, M. Joldes, J.-M. Muller, A.-M. Naneş, and J. Picot, "Error analysis of some operations involved in the Fast Fourier Transform." Submitted, https://hal.archives-ouvertes.fr/hal-01949458/, Dec. 2018.

[R2] D. Arzelier, F. Bréhard, M. Joldes, J.-B. Lasserre, M. Léo, and A. Rondepierre, "Global probability of collision: Problem modeling via occupation measures," Tech. Rep. Version 2.0, LAAS-CNRS, June 2018. CNES Research and Transfer Contract, https://hal.laas.fr/hal-02077552.

[R3] M. Joldes, *Rigorous Polynomial Approximations and Applications*. Thèse, École Normale Supérieure de Lyon - ENS LYON, Sept. 2011. https://tel.archives-ouvertes.fr/tel-00657843.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series*. For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 1964.

[2] M. R. Akella and K. T. Alfriend. Probability of collision between space objects. *Journal of Guidance, Control and Dynamics*, 23(5), September-October 2000.

[3] S. Alfano. Aerospace support to space situation awareness. In *MIT Lincoln Laboratory Satellite Operations and Safety Workshop*, Haystack Observatory, Chelmsford, MA, USA, October 2002.

[4] S. Alfano. A numerical implementation of spherical objet collision probability. *Journal of Astronautical Sciences*, 53(1), January-March 2005.

[5] S. Alfano. Review of conjunction probability methods for short-term encounters. In *AAS/AIAA Space Flight Mechanics Meeting*, Sedona, Arizona, USA, February 2007.

[6] S. Alfano. Satellite conjunction Monte Carlo analysis. In *Proceedings of AAS/AIAA Spaceflight Mechanics Meeting*, number AAS 09-233, Savannah, Georgia, USA, February 2009.

[7] P. R. Arantes Gilz. *Embedded and validated control algorithms for the spacecraft rendezvous*. Thèse, Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), Oct. 2018.

[8] P. R. Arantes Gilz, F. Bréhard, and C. Gazzino. Validated semi-analytical transition matrix for linearized relative spacecraft dynamics via Chebyshev polynomials. In *2018 Space Flight Mechanics Meeting*, 2018.

[9] G. Arioli and H. Koch. Non-symmetric low-index solutions for a symmetric boundary value problem. *Journal of Differential Equations*, 252(1):448 – 458, 2012.

[10] V. I. Arnol'd. *Ten problems*, volume 1 of *Adv. Soviet Math.* Amer. Math. Soc., Providence, RI, 1990.

[11] D. Arzelier, C. Louembet, A. Rondepierre, and M. Kara-Zaitri. Analytical study of the impulsive approximation. *Journal of Optimization Theory and Applications*, 159(1):210–230, 2013.

[12] R. Barrio, A. Dena, and W. Tucker. A database of rigorous and high-precision periodic orbits of the Lorenz model. *Computer Physics Communications*, 194:76–83, 2015.

[13] D. Batenkov. Moment inversion problem for piecewise D-finite functions. *Inverse Problems*, 25(10):105001, 24, 2009.

[14] M. Benedicks and L. Carleson. The dynamics of the Hénon map. *Annals of Mathematics*, 133(1):pp. 73–169, 1991.

[15] A. Benoit. *Fast Semi-numerical Algorithms for Chebyshev Series*. These, Ecole Polytechnique, 2012.

[16] A. Benoit, A. Bostan, and J. van der Hoeven. Quasi-optimal multiplication of linear differential operators. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'12)*, 08 2012.

[17] A. Benoit and B. Salvy. Chebyshev expansions for solutions of linear differential equations. In J. May, editor, *ISSAC '09: Proceedings of the twenty-second international symposium on Symbolic and algebraic computation*, pages 23–30, 2009.

[18] V. Berinde. *Iterative approximation of fixed points*, volume 1912 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007.

[19] J. Betts. *Practical methods for optimal control and estimation using nonlinear programming*. Advances in Design and Control. SIAM, 2ed edition, 2010.

[20] W. G. Bickley, L. J. Comrie, J. C. Miller, D. H. Sadler, and A. J. Thompson. *Bessel functions. Part II. Functions of positive integer order*, volume X of *Mathematical Tables*. British Association for the Advancement of Science, 1952.

[21] F. Bonaventure and A. Gicquel. Collision risk management in astrium satellites. In *Proceedings of the European Space Surveillance Conference*, Madrid, Espagne, Juin 2011.

[22] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization methods and Software*, 11(1-4):613–623, 1999.

[23] B. Borchers. SDPLIB 1.2, a library of semidefinite programming test problems. *Optimization Methods and Software*, 11(1-4):683–690, 1999.

[24] A. Bostan, F. Chyzak, T. Cluzeau, and B. Salvy. Low complexity algorithms for linear recurrences. In J.-G. Dumas, editor, *ISSAC'06*, pages 31–38. ACM Press, 2006.

[25] A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy, and É. Schost. *Algorithmes efficaces en calcul formel*. Published by the Authors, https://hal.archives-ouvertes.fr/AECF/, 2017.

[26] A. Bostan, F. Chyzak, P. Lairez, and B. Salvy. Generalized Hermite reduction, creative telescoping and definite integration of d-finite functions. In *Proceedings of International Symposium on Symbolic and Algebraic Computation, New York, USA,2018*, pages 95–102, 2018.

[27] A. Bostan, F. Chyzak, and N. Le Roux. Products of ordinary differential operators by evaluation and interpolation. In *ISSAC '08: Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 23–30, New York, NY, USA, 2008. ACM.

[28] A. Bostan, F. Chyzak, Z. Li, and B. Salvy. Fast computation of common left multiples of linear ordinary differential operators. In M. van Hoeij and J. van der Hoeven, editors, *ISSAC '12: Proceedings of the twenty-fifth International Symposium on Symbolic and Algebraic Computation*, pages 99–106, 2012.

[29] A. Bostan and M. Kauers. The complete generating function for Gessel walks is algebraic. *Proceedings of the American Mathematical Society*, 138(9):3063–3078, 2010.

[30] R. Boucher, W. Kang, and Q. Gong. Galerkin optimal control. *Journal of Optimization Theory and Applications*, 169(3):825–847, 2016.

[31] G. I. Boutselis, G. De La Torre, and E. A. Theodorou. Stochastic optimal control using polynomial chaos variational integrators. In *American Control Conference (ACC), 2016*, pages 6586–6591, 2016.

[32] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications Inc., Mineola, NY, second edition, 2001.

[33] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications, 2001.

[34] R. P. Brent and P. Zimmermann. *Modern Computer Arithmetic*. 2011.

[35] N. Brisebarre and S. Chevillard. Efficient polynomial L^∞-approximations. In P. Kornerup and J.-M. Muller, editors, *18th IEEE SYMPOSIUM on Computer Arithmetic*, pages 169–176, Los Alamitos, CA, 2007. IEEE Computer Society.

[36] N. Brisebarre, J.-M. Muller, and A. Tisserand. Computing machine-efficient polynomial approximations. *ACM Trans. Math. Software*, 32(2):236–256, June 2006.

[37] A. Bryson and Y. Ho. *Applied optimal control*. Blaisdell Publishing Company, USA, 1969.

[38] B. Bánhelyi, T. Csendes, T. Krisztin, and A. Neumaier. Global attractivity of the zero solution for Wright's equation. *SIAM Journal on Applied Dynamical Systems*, 13(1):537–563, 2014.

[39] M. Campbell. Collision monitoring within satellite clusters. *IEEE Transactions on Control Systems Technology*, 13(1):42–55, January 2005. http://dx.doi.org/10.1109/tcst.2004.838550.

[40] C. Carasso. *L'algorithme d'échange en optimisation convexe*. PhD thesis, Université Joseph-Fourier-Grenoble I, 1973.

[41] C. Carasso and P. J. Laurent. Un algorithme général pour l'approximation au sens de Tchebycheff de fonctions bornées sur un ensemble quelconque. In *Approximation Theory*, number 556 in Lecture Notes in Mathematics, pages 99–121. Springer Berlin Heidelberg, 1976.

[42] J. Carpenter. Conservative analytical collision probability for design of orbital formations. In *2nd International Symposium on Formation Flying*, 2004.

[43] T. Carter and J. Brient. Linearized impulsive rendezvous problem. *Journal of Optimization Theory and Applications*, 86(3), September 1995. doi: 10.1007/BF02192159.

[44] CelesTrak. Iridium 33/Cosmos 2251 Collision, https://web.archive.org/web/20090317043727/http://celestrak.com/events/collision.asp, 2009.

[45] B. L. Chalmers. The Remez exchange algorithm for approximation with linear restrictions. *Trans. Amer. Math. Soc.*, 223:103–131, 1976.

[46] F. Chan. *Spacecraft Collision Probability*. American Institute of Aeronautics and Astronautics, 2008.

[47] F. K. Chan. Collision probability analyses for earth-orbiting satellites. In *Proceedings of the 7th International Space Conference of Pacific Basin Societies*, Nagasaki, Japan, July 2002.

[48] F. K. Chan. *Spacecraft Collision Probability*. AIAA. The Aerospace Press, 2008.

[49] E. W. Cheney. *Introduction to approximation theory*. AMS Chelsea Publishing, Providence, RI, 1998. Reprint of the second (1982) edition.

[50] S. Chevillard, J. Harrison, M. Joldeş, and C. Lauter. Efficient and accurate computation of upper bounds of approximation errors. *Theoret. Comput. Sci.*, 412(16):1523–1543, 2011.

[51] S. Chevillard and M. Mezzarobba. Multiple precision evaluation of the Airy Ai function with reduced cancellation. In A. Nannarelli, P.-M. Seidel, and P. T. P. Tang, editors, *Proceedings of the 21st IEEE Symposium on Computer Arithmetic*. IEEE Computer Society, 2013.

[52] D. V. Chudnovsky and G. V. Chudnovsky. Approximations and complex multiplication according to ramanujan. In *Pi: A Source Book*, pages 596–622. Springer, 2004.

[53] F. Chyzak. *The ABC of Creative Telescoping: Algorithms, Bounds, Complexity*. Memoir of accreditation to supervise research (HDR), Université d'Orsay, Apr. 2014.

[54] M. Claeys, D. Arzelier, D. Henrion, and J. Lasserre. Moment LMI approach to LTV impulse control. In *Conference on Decision and Control*, Florence, Italy, 2013.

[55] C. W. Clenshaw. A note on the summation of Chebyshev series. *Mathematics of Computation*, 9:118–120, 1955.

[56] C. W. Clenshaw. The numerical solution of linear differential equations in Chebyshev series. *Proceedings of the Cambridge Philosophical Society*, 53(1):134–149, 1957.

[57] V. Coppola. Evaluating the Short Encounter Assumption of the Probability of Collision Formula. In *AAS/AIAA Spaceflight Mechanics Meeting*, number AAS 12-248, February 2012.

[58] V. Coppola. Including Velocity Uncertainty in the Probability of Collision between Space Objects. *Advances in the Astronautical Sciences*, 143, 2012.

[59] C. Daramy-Loirat, D. Defour, F. de Dinechin, M. Gallet, N. Gast, C. Q. Lauter, and J.-M. Muller. CR-LIBM, a library of correctly-rounded elementary functions in double-precision. Technical report, LIP Laboratory, Arenaire team, Dec. 2006.

[60] J. Davenport, B. Poonen, J. Maynard, H. Helfgott, P. H. Tiep, and L. Cruz-Filipe. Machine-Assisted Proofs (ICM 2018 Panel). *arXiv preprint arXiv:1809.08062*, 2018.

[61] E. de Klerk and R. Sotirov. A new library of structured semidefinite programming instances. *Optimization Methods and Software*, 24(6):959–971, 2009.

[62] T. J. Dekker. A floating-point technique for extending the available precision. *Numerische Mathematik*, 18(3):224–242, 1971.

[63] K. DeMars and M. Gualdoni. Information-Theoretic Approaches to Space Object Collision. Paper presented at the 7th European Conference on Space Debris, apr 2017.

[64] J. Dolado-Perez, P. Legendre, R. Garmier, B. Revelin, and X. Pena. Satellite Collision Probability Computation for Long Term Encounters. *Advanced Astronomical Society / American Institute of Aeronautics and Astronautics Astrodynamics Specialist Conference*, 142, 2011.

[65] C. F. Dunkl and Y. Xu. *Orthogonal Polynomials of Several Variables*. Cambridge University Press, 2001.

[66] C. Epstein, W. Miranker, and T. Rivlin. Ultra-arithmetic I: function data types. *Mathematics and Computers in Simulation*, 24(1):1–18, 1982.

[67] J.-L. Figueras, M. Gameiro, J. P. Lessard, and R. de la Llave. A Framework for the Numerical Computation and a Posteriori Verification of Invariant Objects of Evolution Equations. *ArXiv e-prints*, May 2016.

[68] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, New York, NY, USA, 1 edition, 2009.

[69] E. Fletcher, V. Navarro, L. Martìn, H. Krag, T. Flohrer, S. Hernàndez, J. De Castro, and J. Arranz. Conjunction evolutions: The process of adapting and evolving operational collision warning software from server to service oriented architecture. In *Proceedings of AIAA SpaceOps Conference*, Stockholm, Sweden, Juin 2012.

[70] T. Flohrer, B. Baptisda Virgili, H. Krag, H. Klinkrad, K. Merz, T. Schildknecht, and S. Lemmens. Tailoring the observation scenarios and data processing techniques for supporting conjunction event assessments. In *Proceedings of the 8th US/Russian Space Surveillance Workshop*, Listvyanka, Russie, Janvier 2012.

[71] J. L. Foster and H. S. Estes. A parametric analysis of orbital debris collision probability and maneuver rate for space debris. *NASA/JSC-25898*, August 1992.

[72] J. L. Foster and S. E. Herbert. A Parametric Analysis of orbital Debris Collision Probability and Maneuver Rate for Space Vehicles. Technical report, NASA Johnson Space Center, August 1992.

[73] S. Foucart and J.-B. Lasserre. Determining Projection Constants of Univariate Polynomial Spaces. *J. Approx. Theory*, 235:74–91, 2018.

[74] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. MPFR: A Multiple-Precision Binary Floating-Point Library with Correct Rounding. *ACM Transactions on Mathematical Software*, 33(2), 2007. Available at http://www.mpfr.org/.

[75] L. Fox. Chebyshev methods for ordinary differential equations. *The Computer Journal*, 4(4):318, 1962.

[76] L. Fox and I. Parker. *Chebyshev polynomials in numerical analysis*. Oxford University Press, 1968.

[77] R. Frigm, M. Hejduk, L. Johnson, and D. Plakalovic. Total probability of collision as a metric for finite conjunction assessment and collision risk management. Technical report, NASA Johnson Space Center, September 2015.

[78] Z. Galias and W. Tucker. Combination of exhaustive search and continuation method for the study of sinks in the Hénon map. In *Proc. IEEE Int. Symposium on Circuits and Systems, ISCAS'13*, pages 2571–2574, Beijing, May 2013.

[79] Z. Galias and W. Tucker. Is the Hénon attractor chaotic? *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(3):033102, 2015.

[80] A. Galligo. Some algorithmic questions on ideals of differential operators. In *European Conference on Computer Algebra*, pages 413–421. Springer, 1985.

[81] R. García-Pelayo and J. Hernando-Ayuso. Series for collision probability in short-encounter model. *Journal of Guidance, Control, and Dynamics*, pages 1908–1916, 2016.

[82] R. Garmier, J. Dolado-Perez, X. Pena, B. Revelin, and P. Legendre. Collision Risk Assessment for Multiple Encounters, 2009. http://www.academia.edu/12629726/Collision_Risk_Assessment_for_Multiple_Encounters.

[83] M. Gasca and T. Sauer. Polynomial interpolation in several variables. *Advances in Computational Mathematics*, 12:377–410, 2000. 10.1023/A:1018981505752.

[84] W. Gawronski, J. Müller, and M. Reinhard. Reduced cancellation in the evaluation of entire functions and applications to the error function. *SIAM Journal on Numerical Analysis*, 45(6):2564–2576, 2007.

[85] K. Geddes. Symbolic computation of recurrence equations for the Chebyshev series solution of linear ODE's. In *Proceedings of the 1977 MACSYMA User's Conference*, pages 405–423, 1977.

[86] I. Gohberg, S. Goldberg, and M. A. Kaashoek. *Basic classes of linear operators*. Birkhäuser Verlag, Basel, 2003.

[87] L. Greengard. Spectral integration and two-point boundary value problems. *SIAM Journal on Numerical Analysis*, 28(4):1071–1080, 1991.

[88] T. C. Hales, J. Harrison, S. McLaughlin, T. Nipkow, S. Obua, and R. Zumkeller. A revision of the proof of the Kepler conjecture. *Discrete & Computational Geometry*, 44(1):1–34, 2010.

[89] R. Halsey and F. Patrick. *Real Analysis*. Prentice Hall, 2010.

[90] H. A. Helfgott. Major arcs for Goldbach's problem. *arXiv preprint arXiv:1305.2897*, 2013.

[91] P. Henrici. *Applied and Computational Complex Analysis, Vol. 3*. Wiley, New York, 1986.

[92] D. Henrion and M. Korda. Convex computation of the region of attraction of polynomial control systems. *IEEE Trans. Automat. Control*, 59(2):297–312, 2014.

[93] D. Henrion and F. Messine. Finding largest small polygons with GloptiPoly. *J. Global Optim.*, 56(3):1017–1028, 2013.

[94] D. Henrion, S. Naldi, and M. Safey El Din. SPECTRA - a Maple library for solving linear matrix inequalities in exact arithmetic. arXiv:1611.01947, 2016.

[95] Y. Hida, X. S. Li, and D. H. Bailey. Algorithms for quad-double precision floating-point arithmetic. In N. Burgess and L. Ciminiera, editors, *Proceedings of the 15th IEEE Symposium on Computer Arithmetic (ARITH-16)*, pages 155–162, Vail, CO, June 2001.

[96] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, 2nd edition, 2002.

[97] D. Hilbert. Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Congress zu Paris 1900. *Nachr. Ges. Wiss. Göttingen, Math.-Phys. Kl.*, 1900:253–297, 1900.

[98] A. Hungria, J.-P. Lessard, and J. D. Mireles James. Rigorous numerics for analytic solutions of differential equations: the radii polynomial approach. *Math. Comp.*, 85(299):1427–1459, 2016.

[99] M. Hénon. A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics*, 50:69–77, 1976. 10.1007/BF01608556.

[100] IEEE Computer Society. *IEEE Standard for Floating-Point Arithmetic*. IEEE Standard 754-2008, Aug. 2008. available at http://ieeexplore.ieee.org/servlet/opac?punumber=4610933.

[101] E. L. Ince. *Ordinary Differential Equations*. Dover Publications, New York, 1956.

[102] A. Iserles. *A first course in the numerical analysis of differential equations*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, second edition, 2009.

[103] L. Jaulin. *Automation for Robotics*. ISTE editions, 2015.

[104] C.-P. Jeannerod and S. M. Rump. Improved error bounds for inner products in floating-point arithmetic. *SIAM J. Matrix Anal. Appl.*, 34(2):338–344, 2013.

[105] B. A. Jones, A. Doostan, and G. H. Born. Nonlinear propagation of orbit uncertainty using non-intrusive polynomial chaos. *Journal of Guidance, Control, and Dynamics*, 36(2):430–444, Jan 2013.

[106] W. Kahan. Lecture notes on the status of IEEE-754. PDF file accessible at http://www.cs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF, 1996.

[107] M. Kanno and M. C. Smith. Validated numerical methods for systems and control engineering. *SIGSAM Bull.*, 37(3):72–73, Sept. 2003.

[108] M. Kanno and M. C. Smith. Validated numerical computation of the L-infinity-norm for linear dynamical systems. *J. Symb. Comput.*, 41(6):697–707, June 2006.

[109] Y. Katznelson. *An introduction to harmonic analysis*. Cambridge University Press, 2004.

[110] E. Kaucher and W. Miranker. *Self-validating numerics for function space problems*. Academic Press, 1984.

[111] D. E. Knuth. *The Art of Computer Programming*, volume 2. Addison-Wesley, Reading, MA, 3rd edition, 1998.

[112] M. Korda. *Moment-sum-of-squares hierarchies for set approximation and optimal control*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2016.

[113] M. Korda, D. Henrion, and C. N. Jones. Inner approximations of the region of attraction for polynomial dynamical systems. *IFAC Proceedings Volumes*, 46(23):534–539, 2013.

[114] C. Koutschan. Advanced applications of the holonomic systems approach. *ACM Comm. Computer Algebra*, 43(3/4):119, 2009.

[115] C. Koutschan. *Advanced applications of the holonomic systems approach*. PhD thesis, Research Institute for Symbolic Computation (RISC), Johannes Kepler University, Linz, Austria, 2009.

[116] G. Krier. Satellite Collision Probability for Long-term Encounters and Arbitrary Primary Satellite Shape. Paper presented at the 7th European Conference on Space Debris, apr 2017.

[117] P. Labourdette, E. Julien, F. Chemama, and D. Carbonne. ATV Jules Verne mission maneuver plan. In 21^{st} *International Symposium on space flight dynamics*, Toulouse, France, 2008.

[118] U. Langer and P. Paule. *Numerical and Symbolic Scientific Computing: Progress and Prospects*. Springer Science & Business Media, 2011.

[119] P. Langlois. Automatic linear correction of rounding errors. *BIT Numerical Mathematics*, 41(3):515–539, Jun 2001.

[120] F. Laporte, M. Moury, and G. Beaumet. CAESAR: An initiative of public service for collision risks mitigation. In *Proceedings of the 6th IAASS Conference*, Montréal, Canada, Mai 2013.

[121] F. Laporte and E. Sasot. Operational Management of Collision Risks for LEO Satellites at CNES. *Space Operations Communicator*, 5(4), 2008.

[122] J. Laskar and M. Gastineau. Existence of collisional trajectories of Mercury, Mars and Venus with the Earth. *Nature*, 459(7248):817–819, June 2009.

[123] J. B. Lasserre. *Moments, positive polynomials and their applications*, volume 1 of *Imperial College Press Optimization Series*. Imperial College Press, London, 2010.

[124] J. B. Lasserre. Computing Gaussian & exponential measures of semi-algebraic sets. *Advances in Applied Mathematics*, 91:137–163, 2017.

[125] J.-B. Lasserre, D. Henrion, C. Prieur, and E. Trélat. Nonlinear optimal control via occupation measures and LMI-relaxations. *SIAM J. Control Optim.*, 47(4):1643–1666, 2008.

[126] J. B. Lasserre and M. Putinar. Algebraic-exponential data recovery from moments. *Discrete Comput. Geom.*, 54(4):993–1012, 2015.

[127] J. B. Lasserre and E. S. Zeron. Solving a class of multivariate integration problems via Laplace techniques. *Applicationes Mathematicae*, 2001.

[128] M. Laurent. Strengthened semidefinite programming bounds for codes. *Mathematical programming*, 109(2-3):239–261, 2007.

[129] C. Q. Lauter. *Arrondi Correct de Fonctions Mathématiques*. PhD thesis, ÉNS de Lyon, Lyon, France, Oct. 2008.

[130] D. Lawden. *Optimal trajectories for space navigation*. Butterworth, London, England, 1963.

[131] O. Le Maître and O. M. Knio. *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer Science & Business Media, 2010.

[132] J.-P. Lessard and C. Reinhardt. Rigorous numerics for nonlinear differential equations using Chebyshev series. *SIAM Journal on Numerical Analysis*, 52(1):1–22, 2014.

[133] B. Levin. *Lectures on Entire Functions*. Translations of Mathematical Monographs. American Mathematical Soc., 1996.

[134] S. Lewanowicz. Construction of a recurrence relation of the lowest order for coefficients of the Gegenbauer series. *Zastosowania Matematyki*, XV(3):345–395, 1976.

[135] S. Lewanowicz. A new approach to the problem of constructing recurrence relations for the Jacobi coefficients. *Zastos. Mat*, 21:303–326, 1991.

[136] P. Lion and M. Handelsman. Primer vector on fixed-time impulsive trajectories. *AIAA Journal*, 6(1):127–, 1968.

[137] P. D. Lizia. *Robust Space Trajectory and Space System Design using Differential Algebra*. Ph.D. thesis, Politecnico di Milano, Milano, Italy, 2008.

[138] M. Lu, B. He, and Q. Luo. Supporting extended precision on graphics processors. In *Proceedings of the Sixth International Workshop on Data Management on New Hardware*, DaMoN '10, pages 19–26, New York, NY, USA, 2010. ACM.

[139] D. Luenberger. *Optimization by Vector Space Methods*. John Wiley and Sons, New York, USA, 1969.

[140] K. Makino. *Rigorous Analysis of Nonlinear Motion in Particle Accelerators*. PhD thesis, Michigan State University, East Lansing, Michigan, USA, 1998.

[141] K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 4(4):379–456, 2003.

[142] S. Marx, T. Weisser, D. Henrion, and J. Lasserre. A moment approach for entropy solutions to nonlinear hyperbolic pdes. *To appear, Math. Control & Related Fields*, 2019.

[143] J. C. Mason and D. C. Handscomb. *Chebyshev polynomials*. CRC Press, 2002.

[144] D. McKinley. Development of a nonlinear probability of collision tool for the Earth observing system. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, page 6295, 2006.

[145] M. Mezzarobba. NumGfun: a package for numerical and analytic computation with D-finite functions. In S. M. Watt, editor, *ISSAC 2010: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation, 25-28 July 2010, Munich, Germany*, pages 139–146. ACM, 2010.

[146] M. Mezzarobba. Rigorous multiple-precision evaluation of D-finite functions in SageMath. Technical Report 1607.01967, arXiv, 2016. Extended abstract of a talk at the 5th International Congress on Mathematical Software.

[147] M. Mezzarobba. Truncation bounds for differentially finite series. *Annales Henri Lebesgue*, 2018.

[148] M. Mezzarobba and B. Salvy. Effective bounds for P-recursive sequences. *Journal of Symbolic Computation*, 45(10):1075–1096, 2010.

[149] H. D. Mittelmann and F. Vallentin. High-accuracy semidefinite programming bounds for kissing numbers. *Experimental Mathematics*, 19(2):175–179, 2010.

[150] O. Møller. Quasi double-precision in floating-point addition. *BIT*, 5:37–50, 1965.

[151] R. D. Monteiro. Primal–dual path-following algorithms for semidefinite programming. *SIAM Journal on Optimization*, 7(3):663–678, 1997.

[152] R. E. Moore. *Interval Analysis*. Prentice-Hall, 1966.

[153] B. Mourrain. Polynomial-exponential decomposition from moments. *Found. Comput. Math.*, 18(6):1435–1492, 2018.

[154] J.-M. Muller, N. Brisebarre, F. de Dinechin, C.-P. Jeannerod, V. Lefèvre, G. Melquiond, N. Revol, D. Stehlé, and S. Torres. *Handbook of Floating-Point Arithmetic*. Birkhäuser Boston, 2010.

[155] D. A. Măceş and M. A. Stadtherr. Computing fuzzy trajectories for nonlinear dynamic systems. *Computers & Chemical Engineering*, 52:10 – 25, 2013.

[156] M. Nakata. A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP,-QD and -DD. In *2010 IEEE International Symposium on Computer-Aided Control System Design*, pages 29–34. IEEE, 2010.

[157] M. Nakata, Y. Takao, S. Noda, and R. Himeno. A fast implementation of matrix-matrix product in double-double precision on NVIDIA C2050 and application to semidefinite programming. In *2012 Third International Conference on Networking and Computing*, pages 68–75, Dec 2012.

[158] T. Nakayama and D. Takahashi. Implementation of multiple-precision floating-point arithmetic library for GPU computing. In *Proceedings of the 23rd IASTED International Conference on Parallel and Distributed Computing and Systems*, PDCS 2011, pages 343–349, December 2011.

[159] A. Neumaier. *Interval methods for systems of equations*. Cambridge University Press, 1990.

[160] A. Neumaier. Taylor forms – use and limits. *Reliable Computing*, 9(1):43–79, 2003.

[161] L. Neustadt. Optimization, a moment problem, and nonlinear programming. *SIAM Journal of Control*, 2(1):33–53, 1964.

[162] NVIDIA. *NVIDIA CUDA Programming Guide 8.0.61*. 2017.

[163] T. Oaku. Algorithms for integrals of holonomic functions over domains defined by polynomial inequalities. *J. Symbolic Comput.*, 50:1–27, 2013.

[164] J. Oliver. Rounding error propagation in polynomial evaluation schemes. *J. Comput. Appl. Math.*, 5(2):85–97, 1979.

[165] S. Olver and A. Townsend. A fast and well-conditioned spectral method. *SIAM Review*, 55(3):462–489, 2013.

[166] A. Oustry, M. Tacchi, and D. Henrion. Inner approximations of the maximal positively invariant set for polynomial dynamical systems, 2019. Submitted.

[167] A. Papoulis and S. Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[168] R. S. Park and D. J. Scheeres. Nonlinear mapping of gaussian statistics: Theory and applications to spacecraft trajectory design. *Journal of Guidance, Control, and Dynamics*, 29(6):1367–1375, Nov 2006.

[169] S. Paszkowski. Zastosowania numeryczne wielomianow i szeregow Czebyszewa. *Podstawowe Algorytmy Numeryczne*, 1975.

[170] R. Patera. Satellite collision probability for nonlinear relative motion. *Journal of Guidance Control and Dynamics*, 26(5):728–733, 2003.

[171] R. P. Patera. General method for calculating satellite conjunction probability. *Journal of Guidance, Control and Dynamics*, 24(4), July-August 2001.

[172] C. Percival. Rapid multiplication modulo the sum and difference of highly composite numbers. *Math. Comp.*, 72:387–395, 2002.

[173] M. Phillips and S. Hur-Diaz. On-board estimation of collision probability for cluster flight. In *Proceedings of the AAS/AIAA Space Flight Mechanics Meeting*, 2013.

[174] G. Plonka, D. Potts, G. Steidl, and M. Tasche. *Numerical Fourier Analysis*. Appl. Numer. Harm. Anal. Birkhäuser, 2018.

[175] G. Pólya. Über positive darstellung von polynomen vierteljschr. *Naturforsch. Ges.*, 73, 1928.

[176] V. Popescu. *Towards fast and certified multiple-precision libraries*. PhD thesis, Université de Lyon, 2017. Available at https://hal.archives-ouvertes.fr/tel-01534090.

[177] M. J. D. Powell. *Approximation theory and methods*. Cambridge University Press, 1981.

[178] D. M. Priest. Algorithms for arbitrary precision floating point arithmetic. In P. Kornerup and D. W. Matula, editors, *Proceedings of the 10th IEEE Symposium on Computer Arithmetic (Arith-10)*, pages 132–144. IEEE Computer Society Press, Los Alamitos, CA, June 1991.

[179] J. Prussing. Primer vector theory and applications. In B. Conway, editor, *Spacecraft trajectory optimization*, pages 16–36. Cambridge University Press, New York, NY, USA, 2010.

[180] L. B. Rall. *Computational solution of nonlinear operator equations*. John Wiley & Sons Inc., New York, 1969.

[181] G. Ramos. Roundoff error analysis of the fast Fourier transform. *Math. Comp.*, 25(116):757–768, 1971.

[182] A. Rauh and E. P. Hofer. *Interval Methods for Optimal Control*, pages 397–418. Springer New York, New York, NY, 2009.

[183] L. Rebillard. *Etude théorique et algorithmique des series de Chebychev, solutions d'équations différentielles holonomes*. PhD thesis, Institut national polytechnique de Grenoble, 1998.

[184] R. Reemtsen and J.-J. Rückman. Numerical techniques for semi-infinite programming: A survey. In R. Reemtsen and S. Görner, editors, *Semi-infinite programming*, volume 25 of *Nonconvex Optimization and Its Applications*, pages 195–262. Springer, New York, NY, USA, 1998.

[185] N. Revol and F. Rouillier. Motivations for an arbitrary precision interval arithmetic and the MPFI library. *Reliable Computing*, 11:1–16, 2005. Available at http://mpfi.gforge.inria.fr/.

[186] T. J. Rivlin. *The Chebyshev polynomials*. John Wiley & Sons, first edition, 1974.

[187] I. M. Ross and M. Karpenko. A review of pseudospectral optimal control: From theory to flight. *Annual Reviews in Control*, 36(2):182 – 197, 2012.

[188] T. Roubíček. Numerical techniques in relaxed optimization problems. In A. Kurdila, P. Pardalos, and M. Zabarankin, editors, *Robust Optimization-Directed Design*, volume 81 of *Nonconvex Optimization and Its Applications*, pages 157–178. Springer, USA, 2006.

[189] S. M. Rump. Error estimation of floating-point summation and dot product. *BIT*, 52(1):201–220, 2012.

[190] B. N. Ryland and H. Z. Munthe-Kaas. On multivariate Chebyshev polynomials and spectral approximations on triangles. Hesthaven, Jan S. (ed.) et al., Spectral and high order methods for partial differential equations. Selected papers from the 8th ICOSAHOM '09 conference, Trondheim, Norway, June 22–26, 2009. Berlin: Springer. Lecture Notes in Computational Science and Engineering 76, 19-41 (2011)., 2011.

[191] B. Salvy. Linear differential equations as a data-structure. *Foundations of Computational Mathematics*, abs/1811.08616, 2018.

[192] B. Salvy and P. Zimmermann. Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software*, 20(2):163–177, 1994.

[193] S. Sarabandi and F. Thomas. Accurate computation of quaternions from rotation matrices. In J. Lenarcic and V. Parenti-Castelli, editors, *Advances in Robot Kinematics 2018*, pages 39–46, Cham, 2019. Springer International Publishing.

[194] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing*, 7:281–292, 1971. In German.

[195] A. Schrijver. New code upper bounds from the terwilliger algebra and semidefinite programming. *IEEE Transactions on Information Theory*, 51(8):2859–2866, 2005.

[196] L. Schwartz. *Théorie des distributions*. Publications de l'Institut de Mathématique de l'Université de Strasbourg, No. IX-X. Nouvelle édition, entiérement corrigée, refondue et augmentée. Hermann, Paris, 1966.

[197] R. Serra. *Opérations de proximité en orbite : évaluation du risque de collision et calcul de manoeuvres optimales pour l'évitement et le rendez-vous*. PhD thesis, 2015. http://www.theses.fr/2015ISAT0035.

[198] A. Shapiro. Semi-infinite programming, duality, discretization and optimality conditions. *Optimization*, 58(2):133–161, 2009.

[199] J. R. Shewchuk. Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete Computational Geometry*, 18:305–363, 1997.

[200] D. Simmons-Duffin. A Semidefinite Program Solver for the Conformal Bootstrap. *JHEP*, 06:174, 2015.

[201] A. Solovyev, M. S. Baranowski, I. Briggs, C. Jacobsen, Z. Rakamarić, and G. Gopalakrishnan. Rigorous estimation of floating-point round-off errors with symbolic taylor expansions. *ACM Trans. Program. Lang. Syst.*, 41(1):2:1–2:39, Dec. 2018.

[202] R. P. Stanley. Differentiably finite power series. *European Journal of Combinatorics*, 1(2):175–188, 1980.

[203] W. A. Stein et al. Sage Mathematics Software (Version 5.12), 2013. http://www.sagemath.org.

[204] S. Streif, D. Henrion, and R. Findeisen. Probabilistic and set-based model invalidation and estimation using lmis. *IFAC Proceedings Volumes*, 47(3):4110–4115, 2014.

[205] J. F. Sturm. Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999.

[206] N. Takayama. An algorithm of constructing the integral of a module - an infinite dimensional analog of gröbner basis. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation, Tokyo, Japan, 1990*, pages 206–211, 1990.

[207] G. Tan, L. Li, S. Triechle, E. Phillips, Y. Bao, and N. Sun. Fast implementation of dgemm on fermi gpu. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 35. ACM, 2011.

[208] A. Tisserand. High-performance hardware operators for polynomial evaluation. *International Journal of High Performance Systems Architecture (IJHPSA)*, 1(1):14–23, 2007.

[209] L. N. Trefethen. *Spectral methods in MATLAB*, volume 10. SIAM, 2000.

[210] L. N. Trefethen. Computing numerically with functions instead of numbers. *Mathematics in Computer Science*, 1(1):9–19, 2007.

[211] L. N. Trefethen. *Approximation theory and approximation practice*. SIAM, 2013.

[212] J. Tschauner. Elliptic orbit rendezvous. *AIAA Journal*, 5(6):1110–1113, 1967.

[213] W. Tucker. The Lorenz attractor exists. *C. R. Acad. Sci. Paris Sér. I Math.*, 328(12):1197–1202, 1999.

[214] W. Tucker. Auto-validating numerical methods, 2009.

[215] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95:189–217, 2003.

[216] J. B. van den Berg, A. Deschênes, J.-P. Lessard, and J. D. M. James. Stationary coexistence of hexagons and rolls via rigorous computations. *SIAM Journal on Applied Dynamical Systems*, 14(2):942–979, 2015.

[217] J. B. van den Berg and J.-P. Lessard. Rigorous numerics in dynamics. *Notices of the AMS*, 62(9), 2015.

[218] J. Van der Hoeven. Fast evaluation of holonomic functions. *Theoretical Computer Science*, 210(1):199–215, 1999.

[219] M. Vasile, C. O. Absil, and A. Riccardi. Set propagation in dynamical systems with generalised polynomial algebra and its computational complexity. *Communications in Nonlinear Science and Numerical Simulation*, 75:22 – 49, 2019.

[220] V. Vittaldev, R. P. Russell, and R. Linares. Spacecraft uncertainty propagation using Gaussian mixture models and polynomial chaos expansions. *Journal of Guidance, Control, and Dynamics*, Sep 2016.

[221] D. Wang and L.-H. Zhi. *Symbolic-numeric computation*. Springer Science & Business Media, 2007.

[222] G. A. Watson. The calculation of best restricted approximations. *SIAM J. Numer. Anal.*, 11(4):693–699, 1974.

[223] D. Widder. *The Laplace transform*. Princeton mathematical series. Princeton university press, 1946.

[224] D. Widder. *An introduction to transform theory*. Academic Press New York, 1971.

[225] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[226] J. Wimp. *Computation with Recurrence Relations*. Pitman, Boston, 1984.

[227] S. Wolfram. *The Mathematica Book*. Wolfram Media, Incorporated, 5 edition, 2003.

[228] M. Wyman. The asymptotic behaviour of the Laurent coefficients. *Canadian Journal of Mathematics*, 11:534–555, 1959.

[229] N. Yamamoto. A numerical verification method for solutions of boundary value problems with local uniqueness by Banach's fixed-point theorem. *SIAM Journal on Numerical Analysis*, 35(5):2004–2013, 1998.

[230] K. Yamanaka and F. Ankersen. New state transition matrix for relative motion on an arbitrary elliptical orbit. *Journal of Guidance, Control, and Dynamics*, 25(1):60–66, January 2002. http://dx.doi.org/10.2514/2.4875.

[231] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakata, and M. Nakata. Latest Developments in the SDPA Family for Solving Large-Scale SDPs. In M. F. Anjos and J. B. Lasserre, editors, *Handbook on Semidefinite, Conic and Polynomial Optimization*, volume 166 of *International Series in Operations Research & Management Science*, pages 687–713. Springer US, 2012.

[232] D. Zeilberger. A holonomic systems approach to special functions identities. *Journal of Computational and Applied Mathematics*, 32(3):321–368, 1990.

[233] Y.-K. Zhu and W. B. Hayes. Algorithm 908: Online exact summation of floating-point streams. *ACM Transactions on Mathematical Software (TOMS)*, 37(3):37, 2010.

Curriculum Vitae

Mioara Joldes
http://homepages.laas.fr/mmjoldes/
LAAS-CNRS
7, avenue du Colonel Roche
BP 54200
31031 Toulouse cedex 4

Tél: (+33) (0)5 61 33 69 26
E-mail: joldes@laas.fr

Doctoral School : ED 475, Mathématiques, Informatique, Télécommunications de Toulouse (Mathematics, Computer Science and Telecomunications, Toulouse)

Name : Joldes Surname : Mioara
Date of birth : 12/08/1984
Nationality : Romanian
Position : Chargée de Recherche CNRS, CRCN, Section 06 (CNRS Researcher),
MAC Team (Methods and Algorithms in Control), LAAS-CNRS, Toulouse, France.

PhD Thesis defended on September 26, 2011 at École Normale Supérieure de Lyon, France.

I - Education

2008 - 2011 **PhD in Computer Science** at École Normale Supérieure de Lyon. *Rigorous polynomial approximations and applications.* Advisors: N. Brisebarre and J.-M. Muller.

2007 - 2008 **Master Diploma in Computer Science** at École Normale Supérieure de Lyon. Advisors: N. Brisebarre et J.-M. Muller.

2003 - 2008 **Engineer Diploma** obtained in June 2008 from Technical University of Cluj-Napoca, Computer Science Department.

II - Work

Jan 2013 - CNRS Researcher (CRCN, Section 06) in MAC Team, LAAS-CNRS, Toulouse, France.

2011 - 2013 Postdoctorant, Computer-aided proofs in analysis (CAPA) Team, Uppsala Univ., Sweden.

III - Teaching

- October 2018: Invited class (2h), Master 2 level, *Approximation Theory and Proof Assistants: Certified Computations*, Fundamental Informatics, ÉNS de Lyon, France.
- June 2018: Invited class (2h), PhD level, CEA-EDF-INRIA Summer School, Paris, France.
- January 2018: Invited class (3h), Master/PhD level at Winter Workshop on Dynamics, Topology and Computations, Bedlewo, Poland.

- 2014, 2015: Small classes (Travaux dirigés et travaux pratiques 6+2x24h), Master 1 level, C programming language, at l'Université Toulouse 3, Paul Sabatier.
- 2014, 2015: Course (2x5h), L2 level, Automatic Control at ENSICA Toulouse.
- 2012 : Qualification MdC in CNU Section 26 (Applied Mathematics) ad 27 (Computer Science).
- 2008-2011: Teaching assistant at ÉNS Lyon and Université Lyon 1.

IV - Research

Rigorous/Validated Computing, Computer Arithmetic, Symbolic-Numeric Computing, Applications to Optimal Control and Aerospace

Invited conferences, seminaries (selection)

- **March 2019**: Invited talk at National Days of Informatics-Mathematics, (Journées du GDR IM), Orléans, France.
- **October 2018**: Invited talk at the conference Dynamics, Topology and Computations, Bedlewo, Poland.
- **October 2017**: Plenary talk at RAIM 2017: Rencontres Arithmétique de l'Informatique Mathematique, Lyon, France.
- **July 2017** : Semi-plenary talk at Foundations of Computational Mathematics, FOCM2017, Barcelona, Spain.
- **May 2017** : Invited talk at the Department of Mechanical & Aerospace Engineering, University of Strathclyde, Glasgow, UK.
- **September 2016** : Plenary talk at SCAN conference, 17th International Symposium on Scientific Computing, Computer Arithmetics and Verified Numerics, Sweden.
- **August 2016** : Talk at ACA, 20th IFAC Symposium on Automatic Control in Aerospace - 21-25 Août, 2016, Sherbrooke, Quebec, Canada.
- **July 2016** : Talk at 5th International Congress on Mathematical Software, Berlin.
- **April 2016** : Talk at Specfun Team Seminar Computations and Proofs, Inria Saclay, France.
- **March 2016** : Talk at Dali Team Seminar, Perpignan, France.
- **June 2015** : Talk at 2015 CMS Summer Meeting in Charlottetown, University of Prince Edward Island, Canada.
- **June 2015** : Talk at SIAM 13th International Symposium on Orthogonal Polynomials, Special Functions and Applications, Juin 1-5, NIST, Gaithersburg MD, USA.
- **June 2014** :Talk at ASAP conference, Zurich, Switzerland.
- **February 2014** : Talk at Mathematical Structures of Computation - Formal Proof, Symbolic Computation and Computer Arithmetic (SMC2014), Lyon, France.
- **November 2013** : Talk at "13ème forum des jeunes mathématicien-ne-s", Lyon, France.
- **Avril 2012**: Talk at National Institute of Aerospace, Hampton, Virginia, USA.

Software

- **CAMPARY** –CudA Multiple Precision ARithmetic librarY–. Multiple precision arithmetic routines based on Floating-Point Expansions for CPUs/GPUs http://homepages.laas.fr/mmjoldes/campary/. Developed with O. Marty, J.-M. Muller, V. Popescu et W. Tucker.

- **Unifapprox** Experimental Maple code for Rigorous Uniform Approximation of D-Finite Functions using Chebyshev Expansions; http://homepages.laas.fr/mmjoldes/Unifapprox/. Developed with A. Benoit and M. Mezzarobba.

- **ChebModels**, Maple package for rigorous univariate polynomial approximations; http://www.ens-lyon.fr/LIP/Arenaire/Ware/ChebModels/. Developed with N. Brisebarre.

V - Supervision

Students supervision: Starting 2013, I supervised or co-supervised 5 undergraduate internships, 5 master internships and 3 PhD theses.

PhD Students:

- 2016– F. Bréhard, PhD defense expected July 2019, *Tools for Certified Numeric Computations*: co-supervision (40%) with N. Brisebarre and D. Pous (LIP, ÉNS Lyon).

- 2015– 2018 P.R. Arantes Gilz, PhD defense on Octobre 17, 2018, *Embedded and validated control algorithms for the spacecraft rendezvous*: co-supervision (50%) with C. Louembet (MAC, LAAS).

- 2014–2017 V. Popescu, PhD defense on July 6, 2017, *Towards Fast and Certified Multiple Precision Arithmetic Libraries*: co-supervision (50%) with J.-M. Muller (LIP, ÉNS Lyon).

Master internships:

- 2018-2019 M.-F. Montaruli, 6 months Master 2 internship: co-supervision (33%) with D. Arzelier et S. Laurens (CNES).

- 2018 D. Guého, 4 months Master 1 internship: co-supervision (40%) with S. Laurens (CNES), D. Arzelier and A. Rondepierre (ROC, LAAS).

- 2017 L. Martire, 6 months Master 2 internship: co-supervision (33%) with D. Arzelier and A. Rondepierre.

- 2016 F. Bréhard, 6 months Master 2 internship: supervision (100%).

- 2014 V. Popescu, 6 months Master 2 internship: supervision (100%).

Undergraduate internships (1.5 - 3 months) :

- 2018 A.-M. Nanes, co-supervision (30%) with J.-M. Muller and N. Brisebarre.

- 2015 N. Deak, co-supervision (40%) with D. Arzelier and C. Louembet.

- 2015 B. Fulop, co-supervision (70%) with N. Brisebarre.

- 2014 O. Marty, supervision (100%).

- 2013 V. Popescu, co-supervision (70%) with W. Tucker.

VI - Collaborations, Research Projects, Scientific Transfer, Evaluation, Coordination

- March 2020 SQuaRE AIM (American Institute of Mathematics) project *Approximation Theory and Semidefinite Programming*. (Participant). The other participants are M. Dressler (Univ. of California, San Diego), E. de Klerk (Delft Univ.), J. B. Lasserre (LAAS-CNRS), Y. Xu (Univ. of Oregon). Project head: Simon Foucart (Texas A&M Univ.).
- 2014-2019 ANR FASTRELAX : *Fast and reliable approximation*. Project financed by French ANR. (Scientific responsible of LAAS-CNRS partner). Other partners: INRIA Grenoble Centre Rhône-Alpes, INRIA Saclay-Idf, INRIA–Centre Sophia Antipolis–Méditerranée, Université Pierre et Marie Curie (UPMC)-Paris 6. Project head: Bruno Salvy.
- 2016-2018 Research collaboration contract with Centre National d'Études Spatiales (CNES): *Global Collision Probability and Satellites Station Keeping*. (Project head). Other participants: D. Arzelier (LAAS-CNRS), J.-B. Lasserre (LAAS-CNRS), A. Rondepierre (LAAS-CNRS, IMT).
- 2013-2015 ANR VORACE : *Verification of fast optimization algorithms applied in critical embedded control.* (Participant). Other partners: INPT, ONERA, Rockwell Collins France. Project head: Marc Pantel (IRIT).
- 2012-2015 Research collaboration contract with Airbus Defence and Space: *Méthodes d'optimisation probabiliste pour l'évitement de collision spatiale.* (Participant). Project head: Denis Arzelier (LAAS-CNRS).

Reviewing: I have reviewed $\simeq$ 100 articles for the following journals: ACM Transactions on Math Software, Automatica, IEEE Transactions in Computers, Mathematics in Computer Science, Numerical Algorithms, VLSI Journal, Communications in Nonlinear Science and Numerical Simulation, Nonlinear Dynamics, Journal of Approximation Theory, as well as the following conferences: ASAP, Arith, ISSAC, SYNASC. I was member of the Program Committee of the following conferences: SYNASC2016, ISSAC2017, ARITH2019.

PhD Jurys:

- Reviewer (Principal Opponent) of F. M. Bartha's Phd Thesis, *Computer-aided proofs and algorithms in analysis*, Bergen Univ., Math Department, Norway, June 14, 2013;
- Member of R. Serra's Phd Thesis committee, *In-orbit Servicing: Collision Risk Assessment and Optimal Maneuvers for Collision Avoidance and Rendezvous*, defended 10-12-2015 in Toulouse, INSA, LAAS-CNRS;
- Member of G. Rance's Phd Thesis committee, *Commande H-infinie paramétrique et application aux viseurs gyrostabilisés*, defended 9-07-2018 at Centrale Supelec, France.

Conference Organization:

- 2018-2020 Co-organizer of Journées Nationales de Calcul Formel, CIRM, Luminy, France, http://www.jncf2019.uvsq.fr/.
- 2017 Publicity and Media Chair of IFAC –International Federation of Automatic Control 2017 World Congress.
- 2016 Organizer of the *Computer Arithmetic* session for RAIM 2016 : 8ème Rencontres Arithmétique de l'Informatique Mathématique, Banyuls-sur-mer (France).
- 2014 Organizer of the *Symbolic-Numeric* session at SEAMAC Days (MAC Team seminar).