Efficient and Validated Numerical Evaluation of Abelian Integrals

Florent Bréhard, Nicolas Brisebarre, Mioara Joldeş, Warwick Tucker

Efficient and Validated Numerical Evaluation of Abelian Integrals

FLORENT BRÉHARD, Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, France
NICOLAS BRISEBARRE, CNRS, LIP, Inria AriC, Université de Lyon, France
MIOARA JOLDES, LAAS-CNRS, Université de Toulouse, CNRS, France
WARWICK TUCKER, School of Mathematics, Monash University, Australia

Abelian integrals play a key role in the infinitesimal version of Hilbert's 16th problem. Being able to evaluate such integrals – with guaranteed error bounds – is a fundamental step in computer-aided proofs aimed at this problem. Using interpolation by trigonometric polynomials and quasi-Newton-Kantorovitch validation, we develop a validated numerics method for computing Abelian integrals in a quasi-linear number of arithmetic operations. Our approach is both effective, as exemplified on two practical perturbed integrable systems, and amenable to an implementation in a formal proof assistant, which is key to provide fully reliable computer-aided proofs.

CCS Concepts: • **Mathematics of computing** → **Quadrature**; **Ordinary differential equations**; **Approximation**; **Interval arithmetic**; *Interpolation*; Arbitrary-precision arithmetic.

Additional Key Words and Phrases: Hilbert's 16th problem, Abelian integral, limit cycles, rigorous numerics, trigonometric polynomial interpolation, Newton-like operator

ACM Reference Format:
Florent Bréhard, Nicolas Brisebarre, Mioara Joldes, and Warwick Tucker. 2023. Efficient and Validated Numerical Evaluation of Abelian Integrals. 1, 1 (December 2023), 38 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

After more than a century since the early stages of automated reasoning and mechanized theorem proving, notably with Hilbert's program in the beginning of the 20th century, it can be safely said that machines have *not* replaced mathematicians so far. Yet, for an increasing number of problems involving intense calculations, algorithms running on computers have already proven to be much more efficient than pen-and-paper work. Rather than fully machine-generated, the resulting proofs are *computer-assisted*. The field of dynamical systems, in particular, has benefited from automated techniques in computer algebra, numerical analysis and rigorous numerics over the past decades. Famous proofs highlighting these achievements are, for instance, the universality of the Feigenbaum constant [Lanford 1982], the existence of chaos in the Kuramoto–Sivashinsky equations [Wilczak 2003], and the (almost) finiteness of relative equilibria for the 5-body problem of celestial mechanics [Albouy and Kaloshin 2012].

The proliferation of computer-assisted proofs raises several central questions concerning their acceptability and utility by the mathematical community. 1. How efficient can algorithms be made to tackle hard problems from a computational point of view? 2. Can we trust the algorithms used to

This work was partially funded by the NuSCAP (ANR-20-CE48-0014) project of the French national research agency (ANR), by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR), and by the Swedish Research Council (VR-2018-04265).
Authors' addresses: Florent Bréhard, Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000, Lille, France, florent.brehard@univ-lille.fr; Nicolas Brisebarre, CNRS, LIP, Inria AriC, Université de Lyon, Lyon, France, nicolas.brisebarre@ens-lyon.fr; Mioara Joldes, LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, joldes@laas.fr; Warwick Tucker, School of Mathematics, Monash University, VIC 3800, Australia, warwick.tucker@monash.edu.

2023. XXXX-XXXX/2023/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

compute the solutions to these problems? 3. Can we moreover trust the implementations of these algorithms, written in practical programming languages and run on real world computers?

Having those questions in mind, this article deals with a computational problem originating from the infinitesimal Hilbert's 16th problem in dynamical systems, namely the evaluation of so-called Abelian integrals. The challenge is the following, with further details postponed to the rest of the introduction.

Problem 1.1. *For a polynomial potential function $H : \mathbb{R}^2 \to \mathbb{R}$, a rational rescaling factor $\mu : \mathbb{R}^2 \to \mathbb{R}$, a polynomial perturbation $(P, Q) : \mathbb{R}^2 \to \mathbb{R}^2$ and a regular[1] level value h, compute the Abelian integral along the oval $\Gamma(h)$ in the plane:*

$$\mathfrak{I}(h) = \int_{\Gamma(h)} \frac{P(x,y)\mathrm{d}y - Q(x,y)\mathrm{d}x}{\mu(x,y)}, \tag{1.1}$$

where an oval $\Gamma(h)$ *denotes a bounded connected component of the level set $H(x, y) = h$ for a regular value h[2].*

We propose a validated numerical algorithm, which, given polynomial or rational functions H, μ, P, Q, together with the level h, computes an interval that contains the exact value of $\mathfrak{I}(h)$. It is intended to be part of computer-assisted proofs around infinitesimal Hilbert's 16th problem, as in the examples at the end of this article. The method has to be both efficient – since a very large number of accurate digits may be necessary to return a tight enough enclosure of $\mathfrak{I}(h)$ – and strongly reliable – since the (sometimes quite ill-conditioned) numerical calculations are actual parts of the computer-assisted proof. To overcome this, we combine higher-order techniques based on Fourier series with fixed-point a posteriori validation to rigorously approximate the integration path and perform the integration. We obtain an algorithm with quasi-linear complexity in terms of arithmetic operations (see main Theorem 1.4), for which we provide an efficient implementation[3] in Julia[4] [Bezanson et al. 2017]. The method is sufficiently self-contained and is tailored to the specific needs of a formal proof. This will allow us to implement the algorithm within a proof assistant like Coq [Bertot and Castéran 2013] in the near future. The end goal is to provide an efficient and fully certified calculator for the infinitesimal Hilbert's 16th problem.

1.1 Hilbert's 16th problem

In 1900, at the International Congress of Mathematics held in Paris, David Hilbert presented ten open problems in mathematics, and later published a more comprehensive list of 23 problems [Hilbert 1900] aimed at challenging the mathematical community. Today, most of the Hilbert problems have been resolved (two of them were deemed to be unresolvable), but a few ones still remain open: one of these is Hilbert's 16th problem.

Hilbert's 16th problem has two distinct parts: one in real algebraic geometry, and one in dynamical systems. We shall address the latter which asks for $\mathcal{H}(n)$ – the maximal number of limit cycles (i.e., isolated periodic orbits [Christopher and Li 2007, §II.1.1]) the family of two-dimensional polynomial vector fields of (total) degree at most n can display. Note that the bound $\mathcal{H}(n)$ should be uniform, that is, it should not depend on the particular polynomial vector field, only on its degree n. As of today, this question is not resolved even in the simplest case $n = 2$. Even finding non-trivial *lower* bounds for $\mathcal{H}(n)$ appears to be very hard.

[1] It means that $\nabla H(x, y) \neq 0$ for all (x, y) s.t. $H(x, y) = h$.
[2] This makes $\Gamma(h)$ a Jordan curve, C^∞-diffeomorphic to the unit circle.
[3] The repositories for our code are available from https://gitlab.inria.fr/abintvalid.
[4] https://julialang.org/

In light of the lack of progress regarding bounds for $\mathcal{H}(n)$, in the mid-seventies, V.I. Arnold [Arnold 1977, 1990] proposed to study a restricted version of the original problem, now known as the *infinitesimal* (or *weak*, or *tangential*) Hilbert's 16th problem. Rather than considering the class of all polynomial vector fields of a certain degree, Arnold suggested that only small perturbations of Hamiltonian polynomial vector fields be considered. Thus, the corresponding question can be asked:

Problem 1.2. *Consider the differential system in* $\mathbb{R}^2$*:*

$$\begin{cases} \dot{x} = -\dfrac{\partial H}{\partial y}(x,y) + \varepsilon P(x,y), \\ \dot{y} = \dfrac{\partial H}{\partial x}(x,y) + \varepsilon Q(x,y), \end{cases} \tag{1.2}$$

where $H(x,y)$ *is a polynomial of degree at most* $n+1$*. The polynomials* P *and* Q *of degree at most* n*, and the positive number* $\varepsilon > 0$ *define the small perturbation of the Hamiltonian system.*

Is there a bound $\mathcal{Z}(n)$ *on the number of limit cycles the system (1.2) can have (for small* ε*), that only depends on the degree* n*?*

For the infinitesimal problem, significant progress has been made, notably the proof of the finiteness of $\mathcal{Z}(n)$ for all n [Binyamini et al. 2010], and the uniform bound for the quadratic case $\mathcal{Z}(2) = 2$ [Chen et al. 2006]

1.2 The Poincaré-Pontryagin theorem and Abelian integrals

The study of perturbed Hamiltonian (or even integrable) systems heavily depends on a theorem by Poincaré and Pontryagin, that makes a strong connection between the existence of a limit cycle and a zero of the *Abelian integral* (1.1).

The *Poincaré return map* is the key tool to understand this connection. Consider the unperturbed Hamiltonian system in a region made of a continuous foliation of periodic orbits. Take a transversal Σ, that is an open interval of a line crossing all orbits it encounters non-tangentially. This interval may be parameterized by the parameter h (the energy level) of the unperturbed system for some domain $h^- < h < h^+$. By continuity, for every $h \in (h^-, h^+)$ and sufficiently small ε, the trajectory originating from the point $\Sigma(h)$ on the transversal in the perturbed Hamiltonian system will cross Σ again. The Poincaré return map Π associates to h and ε the parameter $\Pi(h,\varepsilon)$ corresponding to the first point of return to Σ. We call $d(h,\varepsilon) = \Pi(h,\varepsilon) - h$ the *displacement function*. Clearly, the point $\Sigma(h)$ belongs to a periodic orbit of the perturbed system if and only if $d(h,\varepsilon) = 0$, and this is a limit cycle if and only if the zero of $d(\cdot,\varepsilon)$ is isolated.

The Poincaré-Pontryagin theorem roughly states that, for a Hamiltonian (or more generally, integrable) system perturbed by ε-small terms, the Abelian integral – which is nothing but the integral of the perturbation along a non-perturbed periodic orbit – is the first-order approximation in ε of the Poincaré return map. We now state the following adaptation of this result (see [Christopher and Li 2007] for a proof) that will prove useful in the sequel.

Theorem 1.3 (Generalized Poincaré-Pontryagin Theorem). *Let* $H : \mathbb{R}^2 \to \mathbb{R}$ *be a real analytic potential function,* $P, Q : \mathbb{R}^2 \to \mathbb{R}$ *real analytic functions,* $\varepsilon > 0$*, and* $\mu : \mathbb{R}^2 \to \mathbb{R}$ *an analytic* rescaling factor. *Consider the perturbed integrable system:*

$$\begin{cases} \dot{x} = -\mu(x,y)\dfrac{\partial H}{\partial y}(x,y) + \varepsilon P(x,y), \\ \dot{y} = \mu(x,y)\dfrac{\partial H}{\partial x}(x,y) + \varepsilon Q(x,y). \end{cases} \tag{1.3}$$

Let $\Gamma(h)$ be an oval of H of level h over which μ does not vanish. Then the displacement function $d(h,\varepsilon)$ is approximated as

$$d(h,\varepsilon) = \varepsilon\mathfrak{I}(h) + O(\varepsilon^2), \qquad \text{as } \varepsilon \to 0,$$

where the Abelian integral $\mathfrak{I}(h)$ was defined in (1.1).

In particular, $d(h,\varepsilon)$ and $\mathfrak{I}(h)$ have the same sign for small $\varepsilon > 0$. Hence, the number of isolated zeros of $\mathfrak{I}(h)$ where a change of sign occurs (in particular, simple zeros of $\mathfrak{I}(h)$) provides a lower bound for the number of limit cycles of (1.3) that exist for small $\varepsilon > 0$. Considering zeros of $\mathfrak{I}(h)$ of higher multiplicity it is possible to get an *upper* bound on the number of limit cycles that can bifurcate from the unperturbed periodic orbit(s) $\Gamma(h)$, see [Christopher and Li 2007].

A pessimistic, yet constructive, upper bound for $\mathcal{Z}(n)$ was obtained by the authors of [Binyamini et al. 2010] by bounding the number of zeros of $\mathfrak{I}(h)$ in terms of the degree n only, using the Picard-Fuchs differential equations satisfied by $h \mapsto \mathfrak{I}(h)$. On the other hand, counting the sign alternations of the Abelian integrals for well chosen integrable systems and perturbations can provide lower bounds for some $\mathcal{H}(n)$ [Li et al. 2009] or $\mathcal{Z}(n)$ [Johnson and Tucker 2010].

1.3 Rigorous computation of Abelian integrals: challenge and related works

In general, we lack closed forms for the Abelian integrals, and except for specific families of systems where the analytic behavior of these integrals was investigated by pure – but involved! – pen-and-paper techniques (e.g., [Li et al. 2009] for $\mathcal{H}(3) \geqslant 13$), the numerical evaluation of $\mathfrak{I}(h)$ requires a delicate strategy to certify a lower bound on the number of sign changes. Since these computations are a part of the proof, the results must come with strong guarantees, typically validated bounds. This is the field of *rigorous* (or *validated*) *numerics* [Moore 1966; Tucker 2011], where interval and higher order methods are used to enclose the actual value (number or function) in a guaranteed set-valued representation. Moreover, high precision is often necessary, even though only the *sign* of $\mathfrak{I}(h)$ needs to be validated in the end. This is because Abelian integrals with many zeros are often expressed as linear combinations of simpler integrals. These linear combinations are often subject to high cancellations. (see the example of Section 7.1 with Figure 2).

A first work to mention in this regard is the rigorous computation of Poincaré maps [Kapela et al. 2021][5] using the celebrated CAPD[6] [Kapela et al. 2020] library for validated numerics in connection with dynamical systems. Although CAPD's efficiency is well-established, we prefer *not* using rigorous (nonlinear) ODE integrators. Our aim is to keep the method as minimal as possible, in order to ease its implementation in a formal proof assistant.

Several works deal with the rigorous approximation of implicitly defined curves. The closest related one [Johnson and Tucker 2011], by one of the authors of this article, computes the Abelian integral by subdividing the interior of the oval $\Gamma(h)$ and enclosing its border using parallelotopes. A similar approach can be found in [Martin et al. 2013] for the rigorous continuation of one-dimensional varieties. Other techniques we can mention for rigorous path continuation, yet for a different problem, namely rigorous homotopy tracking, are [Beltrán and Leykin 2013; Xu et al. 2018]. A common feature of these methods is that the curve is represented by piecewise low-order approximations, with the accuracy controlled by the step size.

Since we target high precision without compromising the efficiency, we promote the use of global, high-order approximations for a suitable parameterization of the oval $\Gamma(h)$. Parameterizing algebraic curves is a deep topic in algebraic geometry and geometric modelling for computer-aided design. However, existence theorems for exact (polynomial, rational or trigonometric) parameterizations are

[5]See also the tutorial slides: https://kam.mff.cuni.cz/conferences/swim2015/slides/wilczak.pdf
[6]http://capd.ii.uj.edu.pl/

mostly restricted to curves of genus 0 [Abhyankar and Bajaj 1988; Hong and Schicho 1998], while approximate methods (piecewise rational, splines, etc.) developed for more general curves most of the time come without guaranteed error bounds (see e.g., [Bajaj and Xu 1997; Gao and Li 2004; Yang et al. 2010]). These methods usually target the standard binary64 precision (i.e. double precision), which is not sufficient for our problem. A notable exception is the computation of certain elliptic integrals [Bost and Mestre 1988], with quadratic convergence due to the arithmetic-geometric mean.

1.4 Our approach and contributions

We propose a higher-order method where the oval $\Gamma(h)$ is approximated not by parallelotopes, but by interpolation trigonometric polynomials. This allows us to take advantage of the excellent (typically, exponentially converging) approximation properties in Fourier analysis, while keeping a minimalist framework for rigorous computations, namely trigonometric polynomials, which we develop in Section 2. The result is a fully automated and rigorous Algorithm AbIntValid stated in Section 5, together with the following result, that we shall state more precisely as Theorem 5.2.

Theorem 1.4. *Algorithm AbIntValid computes an interval enclosure $\boldsymbol{I}$ for $\mathfrak{I}(h)$ (cf. (1.1)) in $O(N \log N)$ arithmetic operations, where N is the degree used for trigonometric polynomials to approximate the curve $\Gamma(h)$. As $N \to \infty$, the diameter of $\boldsymbol{I}$ tends exponentially fast to* 0.

The theorem says that in order to compute $\mathfrak{I}(h)$ rigorously to k digits of accuracy, one has to use a degree $N = O(k)$, where the constant hidden in this notation depends on the Hamiltonian system and h, but not on k.

Our approach is based on an *a posteriori validation scheme*, where we first approximate the oval $\Gamma(h)$ with a trigonometric parameterization, and afterwards validate a tube around it, over which the Abelian integral is eventually computed rigorously. The main algorithm AbIntValid is decomposed as the following subroutines:

(1) First, the oval $\Gamma(h)$ is approximated by the two routines OvalApproxInit and OvalApproxRefine (both described in Section 3). The result is a pair of degree N trigonometric polynomials $(\check{x}, \check{y})$, such that $t \in [0, 2\pi] \mapsto (\check{x}(t), \check{y}(t))$ is a smooth approximate parameterization of $\Gamma(h)$.
(2) Next, Algorithm OvalValid (described in Section 4) computes an a posteriori error bound for the distance between $(\check{x}, \check{y})$ and $\Gamma(h)$, thus defining a tube. This relies on the Newton-Kantorovitch validation principle summarized in Section 2.3.
(3) After that, Algorithm AbIntValidQuad (described in Section 5) rigorously computes an interval containing $\mathfrak{I}(h)$ by integrating the perturbation over the tube. Therefore, the smaller the obtained error bound is, the tighter the final enclosure is; hence the need for accurate validation techniques.

Finally, in Section 6, we present our implementation and compare it with alternative approaches, and we assess its efficiency on two practical examples borrowed from the literature in Section 7.

2 APPROXIMATION TOOLS FOR FOURIER ANALYSIS

In this section, we collect useful results for the sequel of the text. After recalling some basic facts about interpolation by trigonometric polynomials, we introduce rigorous trigonometric approximations, a tool for performing validated computations, and an arithmetic on these objects.

2.1 Interpolation of periodic functions

Throughout this paper, we shall need to approximate parameterizations of the ovals $\Gamma(h)$. Seeing that these parameterizations are continuous and 2π-periodic, a natural choice for approximating them is via trigonometric polynomials [Zygmund 2002].

Definition 2.1 (Trigonometric Polynomials). A degree N *Trigonometric Polynomial* (TP) $\check{f}$ is a real-valued trigonometric polynomial over $[0, 2\pi]$:

$$\check{f}(t) = a_0 + \sum_{k=1}^{N} \big(a_k \cos(kt) + b_k \sin(kt)\big),$$

where a_N or b_N is nonzero. We represent $\check{f}$ in terms of its $2N+1$ coefficients.

Let $\mathrm{TP}_N(\mathbb{R})$ denote the vector space of trigonometric polynomials with real coefficients and degree at most N, and let $(C^0_{2\pi}, \|\cdot\|_\infty)$ denote the vector space of real continuous 2π-periodic functions, equipped with the supremum norm: $\|f\|_\infty = \max_{0\leqslant x\leqslant 2\pi} |f(x)|$.

A usual way to obtain quasi-optimal trigonometric polynomial approximants is to consider the Fourier series expansion of the function $f \in C^0_{2\pi}$ truncated to order N. And yet, from a computational point of view, it is often more interesting to consider the degree (at most) N trigonometric polynomial that interpolates the function under consideration at the equispaced points $\left(\frac{2k\pi}{2N+1}\right)_{k=0}^{2N}$ [Wright et al. 2015]. We then define the interpolation operator

$$\begin{aligned} \mathcal{I}_N : C^0_{2\pi} &\to \mathrm{TP}_N(\mathbb{R}) \\ f &\mapsto \check{f} \quad \text{s.t.} \quad \check{f}\left(\frac{2k\pi}{2N+1}\right) = f\left(\frac{2k\pi}{2N+1}\right) \text{ for } k = 0, \dots, 2N. \end{aligned} \tag{2.1}$$

The Lebesgue constant Λ_N [Powell 1981] associated to the trigonometric interpolation operator $\mathcal{I}_N$ is the operator norm of $\mathcal{I}_N$, that is to say $\Lambda_N = \sup_{f\in C^0_{2\pi}, \|f\|_\infty=1} \|\mathcal{I}_N(f)\|_\infty$. A first nice feature of these interpolation trigonometric polynomials is that $\Lambda_N = O(\log N)$ [Ehlich and Zeller 1966], which makes them quasi-optimal approximations with respect to $\|\cdot\|_\infty$.

A second nice feature is that we can compute the coefficients of the interpolation polynomial by the Fast Fourier Transform, hence in $O(N\log(N))$ arithmetic operations. Indeed, if $\check{f}(t) = a_0 + \sum_{k=1}^{N} (a_k \cos(kt) + b_k \sin(kt))$ and $f_j = \check{f}\left(\frac{2j\pi}{2N+1}\right)$ for $j = 0, \dots, 2N$, we have, for $k = 1, \dots, N$,

$$a_0 = \frac{1}{2N+1} \sum_{j=0}^{2N} f_j, \quad a_k = \frac{1}{2N+1} \sum_{j=0}^{2N} f_j \cos\left(\frac{2jk\pi}{2N+1}\right), \quad b_k = \frac{1}{2N+1} \sum_{j=0}^{2N} f_j \sin\left(\frac{2jk\pi}{2N+1}\right).$$

Having this in mind, we define

$$\begin{aligned} \mathrm{FFT}_N : \quad \mathbb{R}^{2N+1} &\to \mathrm{TP}_N(\mathbb{R}) \\ (f_j)_{j=0}^{2N} &\mapsto \check{f} \text{ s.t. } \check{f}(t_j) = f_j \text{ for } 0 \leqslant j \leqslant 2N, \end{aligned}$$

and its inverse transform

$$\begin{aligned} \mathrm{IFFT}_N : \quad \mathrm{TP}_N(\mathbb{R}) &\to \mathbb{R}^{2N+1} \\ \check{f} &\mapsto (\check{f}(t_j))_{j=0}^{2N} \text{ where } t_j = \tfrac{2\pi j}{2N+1} \text{ for } 0 \leqslant j \leqslant 2N, \end{aligned}$$

which allows us to interpolate a function and to evaluate a TP on the equispaced grid in only $O(N\log(N))$ arithmetic operations. By combining both routines, we can compute the degree $2N$ product of two degree N TPs with the same asymptotic complexity $O(N\log(N))$. In practice, this outperforms the "naive" multiplication algorithm in $O(N^2)$ arithmetic operations for degrees N of the order of magnitude of 100 or more. We assume this $O(N\log(N))$ asymptotic complexity for the product of degree N TPs in what follows.

When working with real trigonometric approximations, a convenient way to bound the supremum norm of functions is to use the $\|\cdot\|_{\ell^1}$ norm directly defined from the Fourier coefficients a_k, b_k of f:

$$\|f\|_{\ell^1} = |a_0| + \sum_{k=1}^{\infty} \big(|a_k| + |b_k|\big).$$

This norm is well-defined as long as the sum converges, and then satisfies $\|f\|_{\ell^1} \geqslant \|f\|_\infty$. Note that the sum does not necessarily converge for $f \in C^0_{2\pi}$. However, this becomes true under very mild regularity assumptions, and a fortiori[7] for analytic f as assumed throughout this article. Finally, we have that $\|fg\|_{\ell^1} \leqslant \|f\|_{\ell^1}\|g\|_{\ell^1}$.

2.2 Rigorous trigonometric approximations

By analogy with intervals used to represent real numbers rigorously on computers [Moore 1966], (generalized) *rigorous polynomial approximations* are used in rigorous numerics to enclose mathematical functions in set-valued representations. Taking its roots in the so-called *ultra-arithmetics* developed in the early 80s [Epstein et al. 1982a,b], this concept has been extended and implemented in various settings, notably with Taylor [Makino and Berz 2003a,b] or Chebyshev [Brisebarre and Joldeş 2010] approximations. Since the functions considered throughout this article are univariate, real-valued and periodic, the trigonometric polynomials expressed in the Fourier trigonometric basis are the natural candidates to build rigorous, set-valued representations. For short, we call them *Rigorous Trigonometric Approximations* (RTA). Some previous works already made use of quite similar concepts (see, e.g., [Figueras and de la Llave 2017; Hungria et al. 2016]). However, for the sake of completeness, and in order to provide sound bases for the complexity analysis of the computation method for Abelian integrals presented in this article, we provide some elementary definitions, basic routines and lemmas concerning RTAs.

Definition 2.2 (Rigorous Trigonometric Approximations). A degree N Rigorous Trigonometric Approximation (RTA) is a pair $\boldsymbol{f} = (\check{f}, \varepsilon)$, with $\check{f}$ a degree N TP and $\varepsilon \geqslant 0$ representing the closed ε-ball around $\check{f}$ in the Banach space $C^0_{2\pi}$:

$$\overline{B}(\check{f}, \varepsilon) = \left\{ f \in C^0_{2\pi} \;\middle|\; \|f - \check{f}\|_\infty \leqslant \varepsilon \right\}.$$

Therefore, we say that a degree N RTA $\boldsymbol{f} = (\check{f}, \varepsilon)$ represents a function $f \in C^0_{2\pi}$ (or $\boldsymbol{f}$ contains f, or simply $f \in \boldsymbol{f}$) if there exists a function $s \in C^0_{2\pi}$ with $\|s\|_\infty \leqslant \varepsilon$ such that $f(t) = \check{f}(t) + s(t)$ for all $t \in \mathbb{R}$.

Linear operations are trivially defined on RTAs to match the (Banach) linear space structure of $(C^0_{2\pi}, \|\cdot\|_\infty)$: for all degree N RTAs $(\check{f}, \varepsilon)$, $(\check{g}, \eta)$ and $\lambda \in \mathbb{R}$, we have

$$\begin{aligned}(\check{f}, \varepsilon) + (\check{g}, \eta) &= (\check{f} + \check{g}, \varepsilon + \eta),\\ (\check{f}, \varepsilon) - (\check{g}, \eta) &= (\check{f} - \check{g}, \varepsilon + \eta),\\ \lambda(\check{f}, \varepsilon) &= (\lambda\check{f}, |\lambda|\varepsilon).\end{aligned}$$

Also, $\textsc{Bound}(\boldsymbol{f})$, defined as:

$$\textsc{Bound}((\check{f}, \varepsilon)) = \sum_{k=0}^{N} |a_k| + \sum_{k=1}^{N} |b_k| + \varepsilon.$$

[7] This is a straightforward corollary of the exponential decay of Fourier coefficients for analytic functions, see e.g. [Wright et al. 2015, Thm. 4.1].

satisfies $\textsc{Bound}(\boldsymbol{f}) \geqslant \|\check{f}\|_{\ell^1} + \varepsilon \geqslant \|\check{f}\|_\infty + \varepsilon$, so that $\|f\|_\infty \leqslant \textsc{Bound}(\boldsymbol{f})$ for all $f \in \boldsymbol{f}$.

The product of two degree N RTAs is the degree $2N$ RTA defined by

$$(\check{f}, \varepsilon) \times (\check{g}, \eta) = \big(\check{f}\check{g}, \textsc{Bound}(\check{f})\eta + \textsc{Bound}(\check{g})\varepsilon + \varepsilon\eta\big),$$

which can be computed in $O(N \log(N))$ arithmetic operations using the fast multiplication algorithm for TPs mentioned previously.

We shall also define the integral of $\boldsymbol{f} = (\check{f}, \varepsilon)$ with $\check{f} = a_0 + \sum_{k=1}^{N} (a_k \cos(kt) + b_k \sin(kt))$ over one period:

$$\int_0^{2\pi} \boldsymbol{f}(t)\mathrm{d}t = 2\pi\,(a_0 + [-\varepsilon, \varepsilon])\,.$$

Lemma 2.3. *If the RTAs $\boldsymbol{f} = (\check{f}, \varepsilon)$ and $\boldsymbol{g} = (\check{g}, \eta)$ represent functions f and g, then the RTA $\lambda\boldsymbol{f}$ represents λf for any $\lambda \in \mathbb{R}$, and the RTA $\boldsymbol{f} \star \boldsymbol{g}$ represents $f \star g$ for $\star \in \{+, -, \times\}$. Also, $\int_0^{2\pi} f(t)\mathrm{d}t \in \int_0^{2\pi} \boldsymbol{f}(t)\mathrm{d}t$ for all $f \in \boldsymbol{f}$.*

Proof. We prove the least trivial case only, namely the multiplication. By hypothesis, there exist functions u and v in $C^0_{2\pi}$ with $\|u\|_\infty \leqslant \varepsilon$ and $\|v\|_\infty \leqslant \eta$, such that $f = \check{f} + u$ and $g = \check{g} + v$. Therefore,

$$\begin{aligned}\|fg - \check{f}\check{g}\|_\infty &\leqslant \|\check{f}v + \check{g}u + uv\|_\infty \leqslant \|\check{f}\|_\infty\|v\|_\infty + \|\check{g}\|_\infty\|u\|_\infty + \|u\|_\infty\|v\|_\infty \\ &\leqslant \textsc{Bound}(\check{f})\eta + \textsc{Bound}(\check{g})\varepsilon + \varepsilon\eta,\end{aligned}$$

which by definition implies $fg \in \boldsymbol{f} \times \boldsymbol{g}$. □

Remark 2.4. *By analogy with a globally set precision for floating-point and interval arithmetic, we can choose to fix a global degree N used for TP / RTAs computations. Since some operations (e.g. multiplication) may increase the degree, the obtained result is truncated. In the case of RTAs, the higher-order terms must be added to the error component:*

$$\textsc{Trunc}(\boldsymbol{f}, N) = \Big(a_0 + \sum_{k=1}^{N} \big(a_k \cos(kt) + b_k \sin(kt)\big), \sum_{k=N+1}^{m} \big(|a_k| + |b_k|\big) + \varepsilon\Big),$$

for $\boldsymbol{f} = \big(a_0 + \sum_{k=1}^{m} \big(a_k \cos(kt) + b_k \sin(kt)\big), \varepsilon\big)$ when $m > N$. Clearly, if $f \in \boldsymbol{f}$, then $f \in \textsc{Trunc}(\boldsymbol{f}, N)$.

These elementary operations on RTAs are the building blocks of more complex ones. For instance, if $p(x)$ is a polynomial and $\boldsymbol{f}$ an RTA for f, then substituting x by $\boldsymbol{f}$ and applying an evaluation scheme of choice with arithmetic operations on RTAs gives an RTA $p(\boldsymbol{f})$ for $p(f)$.

Finally, we also need a routine IsPositive to determine whether, for an RTA $\boldsymbol{f}$, $f > 0$ holds for every $f \in \boldsymbol{f}$. Such a routine is discussed in Appendix A.

2.3 A posteriori validation using Newton-like operators

Many operations defined on functions do not preserve the ring of trigonometric polynomials. Examples include division, the square root, or the more complex operation performed by Algorithm OvalValid in Section 4 used to rigorously approximate a parameterization of the oval $\Gamma(h)$. More generally, the target function φ^* can be viewed as the solution of a functional equation $\mathcal{F}(\varphi) = 0$ for a suitable operator $\mathcal{F}$ mapping a function space to another one. For such cases, the principle of a posteriori validation is widely used in validated numerics:

- First, a TP $\check{\varphi}$ approximating the solution φ^* is computed using nonrigorous algorithms, e.g. by trigonometric interpolation as in Section 2.1.

- Next, an upper bound on the approximation error $\varepsilon \geqslant \|\check{\varphi} - \varphi^*\|_\infty$ is rigorously computed by expressing φ^* as a fixed point of a contracting operator $\mathcal{T}$, and applying the Banach fixed-point theorem (see Theorem 2.5 below). The result is an RTA $\boldsymbol{\varphi} = (\check{\varphi}, \varepsilon)$ such that $\varphi^* \in \boldsymbol{\varphi}$.

Theorem 2.5. *(see, e.g., [Yamamoto 1998, Thm. 2.1]) Let $(E, \|\cdot\|)$ be a Banach space, $\check{\varphi} \in E$, and $\mathcal{T} : E \to E$. If we can find $r, \lambda \geqslant 0$ such that:*

- *$\mathcal{T}$ is λ-contracting over the closed ball $\overline{B}(\check{\varphi}, r)$, i.e., $\lambda < 1$ and:*

$$\|\mathcal{T}(\varphi_1) - \mathcal{T}(\varphi_2)\| \leqslant \lambda \|\varphi_1 - \varphi_2\| \qquad \text{for all} \quad \varphi_1, \varphi_2 \in \overline{B}(\check{\varphi}, r);$$

- *We have $d + \lambda r \leqslant r$, where $d := \|\mathcal{T}(\check{\varphi}) - \check{\varphi}\|$ (called the defect);*

Then $\mathcal{T}$ admits a unique fixed point φ^* in the ball $\overline{B}(\check{\varphi}, r)$, and we have the following enclosure for the approximation error:

$$\frac{d}{1+\lambda} \leqslant \|\check{\varphi} - \varphi^*\| \leqslant \frac{d}{1-\lambda}.$$

Assume the functional equation is of the form $\mathcal{F}(\varphi) = 0$ with $\mathcal{F} : C^0_{2\pi} \to C^0_{2\pi}$ being of class C^1, and let $\check{\varphi}$ be an approximate zero of $\mathcal{F}$. A classical method in rigorous numerics (see e.g., [Hungria et al. 2016; Yamamoto 1998]) to obtain an equivalent fixed-point equation $\mathcal{T}(\varphi) = \varphi$ is to construct $\mathcal{T} : C^0_{2\pi} \to C^0_{2\pi}$ as a Newton-like operator:

$$\mathcal{T}(\varphi) = \varphi - \mathcal{A}(\mathcal{F}(\varphi)).$$

Here, $\mathcal{A} : C^0_{2\pi} \to C^0_{2\pi}$ is an invertible bounded linear operator approximating $(\mathrm{D}\mathcal{F}(\check{\varphi}))^{-1}$, the inverse of the differential of $\mathcal{F}$ at $\check{\varphi}$, making $\mathcal{T}$ a local contraction in a neighborhood of $\check{\varphi}$.

2.3.1 Newton-like validation in the case of polynomial equations. For the purpose of validating approximate parameterizations of planar algebraic curves, we apply the above strategy to *polynomial equations* of the form

$$c_r(t)\varphi(t)^r + c_{r-1}(t)\varphi(t)^{r-1} + \cdots + c_1(t)\varphi(t) + c_0(t) = 0, \qquad t \in [0, 2\pi],$$

with $c_0, \ldots, c_r \in C^0_{2\pi}$, that is to say, $\mathcal{F} \in C^0_{2\pi}[X]$ is a polynomial with coefficients in $C^0_{2\pi}$.

In this case, the differential $\mathrm{D}\mathcal{F}(\check{\varphi})$ coincides with the multiplication by $\mathcal{F}'(\check{\varphi}) \in C^0_{2\pi}$, where the prime symbol denotes the usual differentiation in a ring of polynomials, here $C^0_{2\pi}[X]$. Therefore, the linear operator $\mathcal{A}$ is set to be the multiplication by a TP $\check{a}$ constructed by interpolation to approximate $1/\mathcal{F}'(\check{\varphi})$:

$$\mathcal{A}(\varphi) = \check{a}\varphi, \qquad \text{with} \quad \check{a}(t) \approx \frac{1}{\mathcal{F}'(\check{\varphi})(t)}.$$

The following lemma provides a simple way to bound the Lipschitz constant of the resulting Newton-like operator $\mathcal{T}$.

Lemma 2.6. *If $\mathcal{F} \in C^0_{2\pi}[X]$, $\mathcal{A}(\varphi)(t) = \check{a}(t)\varphi(t)$, and $\mathcal{T}(\varphi) = \varphi - \mathcal{A}(\mathcal{F}(\varphi))$, then $\mathcal{T}$ is $\lambda(r)$-Lipschitz over $\overline{B}(\check{\varphi}, r)$ with $\lambda(r) = \lambda_0 + \alpha r \lambda_1(r)$, obtained from the (rigorously computed) bounds:*

$$\lambda_0 \geqslant \|1 - \check{a}\,\mathcal{F}'(\check{\varphi})\|_\infty, \qquad \alpha \geqslant \|\check{a}\|_\infty, \qquad \lambda_1(r) \geqslant \sup_{\varphi \in \overline{B}(\check{\varphi}, r)} \|\mathcal{F}''(\varphi)\|_\infty.$$

Proof. To get an upper bound for the Lipschitz constant for $\mathcal{T} \in C^0_{2\pi}[X]$ over $\overline{B}(\check{\varphi}, r)$, we simply bound $\|\mathcal{T}'(\varphi)\|_\infty$ for $\|\varphi - \check{\varphi}\|_\infty \leqslant r$ by the triangle inequality. First, $\|\mathcal{T}'(\check{\varphi})\|_\infty = \|1 - \check{a}\,\mathcal{F}'(\check{\varphi})\|_\infty \leqslant \lambda_0$ by definition of λ_0. Then, $\|\mathcal{T}'(\varphi) - \mathcal{T}'(\check{\varphi})\|_\infty = \|\check{a}\,(\mathcal{F}'(\varphi) - \mathcal{F}'(\check{\varphi}))\|_\infty \leqslant \alpha r \lambda_1(r)$ using the mean value theorem, by definition of α and λ_1, and since $\varphi \in \overline{B}(\check{\varphi}, r)$. Adding both bounds, we obtain the expected upper bound for the Lipschitz constant $\lambda(r)$. □

Note that $\mathcal{A}$ being invertible is a byproduct of the second assumption of Theorem 2.5: except in the trivial case $d = 0$, the inequality implies $\lambda_0 < 1$, so $\check{a}$ never vanishes.

Example 2.7 (Division of RTAs). *For two functions $g, h \in C^0_{2\pi}$ with $h(t) \neq 0$ for all t, represented by RTAs $\boldsymbol{g}$ and $\boldsymbol{h}$, Algorithm RTADiv$(\boldsymbol{g}, \boldsymbol{h}, N)$ computes a degree N TP $\check{\varphi}$ approximating the quotient $\varphi^* = g/h$ by trigonometric interpolation, and validates it using Lemma 2.6, since φ^* is the unique zero of $\mathcal{F} \in C^0_{2\pi}[X]$ defined by $\mathcal{F}(\varphi) = h\varphi - g$. When a global degree N for the RTA computations is fixed, we may use the notation $\boldsymbol{g}/\boldsymbol{h}$ for RTADiv$(\boldsymbol{g}, \boldsymbol{h}, N)$.*

Algorithm 1 RTADiv$(\boldsymbol{g}, \boldsymbol{h}, N)$

Input: RTAs $\boldsymbol{g} = (\check{g}, \delta)$ and $\boldsymbol{h} = (\check{h}, \eta)$, and approximation degree $N \in \mathbb{N}$
Output: Degree N RTA $\boldsymbol{\varphi}$ representing the quotient $\boldsymbol{g}/\boldsymbol{h}$

▷ *Build degree N candidate TP $\check{\varphi}$ by interpolation*
1: $(g_j)_{j=0}^{2N} \leftarrow \mathrm{IFFT}_N(\check{g})$ and $(h_j)_{j=0}^{2N} \leftarrow \mathrm{IFFT}_N(\check{h})$
2: $\check{\varphi} \leftarrow \mathrm{FFT}_N\left(\left(\frac{g_j}{h_j}\right)_{j=0}^{2N}\right)$
▷ *Build TP $\check{a}$ defining the Newton-like operator and bound the Lipschitz constant*
3: $\check{a} \leftarrow \mathrm{FFT}_N\left(\left(\frac{1}{h_j}\right)_{j=0}^{2N}\right)$
4: $\lambda \leftarrow \textsc{Bound}(1 - \check{a}\boldsymbol{h})$
▷ *Compute a posteriori error bound, if possible*
5: **if** $\lambda < 1$ **then**
6: $\quad d \leftarrow \textsc{Bound}\big(\check{a}\,(\boldsymbol{h}\check{\varphi} - \boldsymbol{g})\big)$
7: $\quad \varepsilon \leftarrow \frac{d}{1-\lambda}$
8: $\quad$ **return** $\boldsymbol{\varphi} = (\check{\varphi}, \varepsilon)$
9: **else**
10: $\quad$ **return** *FAIL*
11: **end if**

We now prove that the division operator, as implemented in Algorithm1, is correct, convergent, and has a good complexity.

Lemma 2.8. *Let $\boldsymbol{g} = (\check{g}, \delta)$ and $\boldsymbol{h} = (\check{h}, \eta)$ be RTAs. Then,*

(*i*) *For all $N \in \mathbb{N}$, if RTADiv$(\boldsymbol{g}, \boldsymbol{h}, N)$ returns an RTA $\boldsymbol{\varphi}$, then for all $g \in \boldsymbol{g}$ and $h \in \boldsymbol{h}$, $h(t) \neq 0$ for all t and $g/h \in \boldsymbol{\varphi}$.*

(*ii*) *If $\check{h}(t) \neq 0$ for all t and $\eta < \|\check{h}^{-1}\|_{\ell^1}^{-1}$, then there exists an $N_0 \in \mathbb{N}$ such that for all $N \geqslant N_0$, Algorithm RTADiv$(\boldsymbol{g}, \boldsymbol{h}, N)$ does not fail, and the remainder of $\boldsymbol{\varphi} = (\check{\varphi}, \varepsilon)$ satisfies*

$$\varepsilon \leqslant \frac{\|\check{h}^{-1}\|_{\ell^1}\delta + \|\check{g}/\check{h}^2\|_{\ell^1}\eta}{1 - \|\check{h}^{-1}\|_{\ell^1}\eta} + O(\kappa^{-N}), \qquad \text{as } N \to \infty,$$

for some $\kappa > 1$ depending on $\check{g}$ and $\check{h}$.

(*iii*) *RTADiv$(\boldsymbol{g}, \boldsymbol{h}, N)$ runs in $O(N' \log(N'))$ arithmetic operations, where $N' = \max(\deg \boldsymbol{g}, \deg \boldsymbol{h}, N)$.*

Proof. (*i*) *Soundness.* The Newton-like operator associated to $\mathcal{F}$ is defined by $\mathcal{T}(\varphi) = \varphi - \check{a}\,(h\varphi - g)$ with $\check{a}$ a degree N TP approximating $1/\mathcal{F}'(\check{\varphi}) = 1/h(t)$ (line 3). Since $\mathcal{F}$ (and hence $\mathcal{T}$) has degree 1,

$\mathcal{F}'' = 0$. Hence the bound $\lambda_1(r)$ in Lemma 2.6 is identically 0 , and we have $\lambda(r) = \lambda_0 = \|1 - \check{a}h\|_\infty$, rigorously upper-bounded by λ in line 4. Hence, as soon as $\lambda < 1$, we can set r arbitrary large in Theorem 2.5 and return the bound implemented in lines 6 and 7:

$$\|\check{\varphi} - \varphi^*\|_\infty \leqslant \frac{\|\mathcal{T}(\check{\varphi}) - \check{\varphi}\|_\infty}{1-\lambda} \leqslant \frac{\|\check{a}\,(h\check{\varphi} - g)\|_\infty}{1-\lambda} \leqslant \frac{\textsc{Bound}\big(\check{a}\,(\boldsymbol{h}\check{\varphi} - \boldsymbol{g})\big)}{1-\lambda}.$$

Therefore, the degree N RTA $\boldsymbol{\varphi} = (\check{\varphi}, \varepsilon)$ returned in line 8 contains g/h for all $g \in \boldsymbol{g}$ and $h \in \boldsymbol{h}$. Note moreover that the nonvanishing of all $h \in \boldsymbol{h}$ is a byproduct of the inequality $\|1 - \check{a}h\|_\infty \leqslant \lambda < 1$.

(*ii*) *Convergence.* Suppose that $\check{h}(t) \neq 0$ for all t, and that $\eta < \|\check{h}^{-1}\|_{\ell^1}^{-1}$. In particular, $\eta < \|\check{h}^{-1}\|_\infty^{-1} = \min_{t\in\mathbb{R}} |\check{h}(t)|$. Since $\check{h}^{-1}$ (resp. $\check{g}/\check{h}$) is analytic over $\mathbb{R}$, then $\check{a} = \mathcal{I}_N(\check{h}^{-1})$ (resp. $\check{\varphi} = \mathcal{I}_N(\check{g}/\check{h})$) converges to $\check{h}^{-1}$ (resp. $\check{g}/\check{h}$) w.r.t. $\|\cdot\|_{\ell^1}$ exponentially fast as $N \to \infty$ (see e.g. [Wright et al. 2015, Thm. 4.2][8]).

Now for the bound λ computed in line 4, we have for all $h \in \boldsymbol{h}$,

$$\|1 - \check{a}h\|_{\ell^1} \leqslant \|1 - \check{a}\check{h}\|_{\ell^1} + \|\check{a}\|_{\ell^1}\|h - \check{h}\|_{\ell^1} \leqslant \|\check{a} - \check{h}^{-1}\|_{\ell^1}\|\check{h}\|_{\ell^1} + \|\check{a}\|_{\ell^1}\eta \leqslant \|\check{h}^{-1}\|_{\ell^1}\eta + O(\kappa^{-N}),$$

for some $\kappa > 1$, as $N \to \infty$. In particular, Algorithm RTADiv always terminates for sufficiently large N. Similarly, for the bound d computed in line 6, for all $g \in \boldsymbol{g}$ and $h \in \boldsymbol{h}$,

$$\|\check{a}(h\check{\varphi} - g)\|_{\ell^1} \leqslant \underbrace{\|\check{a}(\check{h}\check{\varphi} - \check{g})\|_{\ell^1}}_{\to 0} + \underbrace{\|\check{a}\|_{\ell^1}}_{\to \|\check{h}^{-1}\|_{\ell^1}}\delta + \underbrace{\|\check{a}\check{\varphi}\|_{\ell^1}}_{\to \|\check{g}/\check{h}^2\|_{\ell^1}}\eta \leqslant \|\check{h}^{-1}\|_{\ell^1}\delta + \|\check{g}/\check{h}^2\|_{\ell^1}\eta + O(\kappa^{-N}),$$

for some $\kappa > 1$, as $N \to \infty$. This finally yields the expected estimate for ε computed in line 7.

(*iii*) *Complexity.* The asymptotic complexity is determined by the FFT_N/IFFT_N routines used for evaluation on the equispaced grid, interpolation and TP/RTA multiplications. □

Remark 2.9. *In the case where the polynomial operator $\mathcal{F}$ has degree greater than* 1 *(which will be the case for Algorithm* OvalValid *in Section 4 as long as the potential function $H(x, y)$ has degree at least* 2*), an explicit radius r satisfying the hypothesis of Theorem 2.5 must be rigorously computed. Given the function $r \mapsto \lambda(r)$, this amounts to locating the smallest positive zero of $(1 - \lambda(r))r - d$. A simple bisection method* NewtonBall *is proposed in Appendix C for this purpose.*

3 APPROXIMATION OF THE OVAL $\Gamma(h)$ AND NUMERICAL INTEGRATION

This section is devoted to the computation of a smooth approximate parameterization $t \in [0, 2\pi] \to (\check{x}(t), \check{y}(t))$ of the oval $\Gamma(h)$ with $\check{x}, \check{y}$ TPs of degree N.

First, an implicit parameterization for $\Gamma(h)$ is given by the unscaled, unperturbed Hamiltonian system itself (i.e., System (1.2) with $\varepsilon = 0$):

$$\begin{cases} \dot{x} = -\dfrac{\partial H}{\partial y}(x, y), \\ \dot{y} = \dfrac{\partial H}{\partial x}(x, y). \end{cases} \tag{3.1}$$

For a level h on the portion of Σ of interest, the trajectory of (3.1) starting at $(x_{ini}, y_{ini}) = \Sigma \cap H^{-1}(h)$ is periodic since it exactly follows the oval $\Gamma(h)$. Call t_{end} the time of first return onto Σ. By a linear change of the independent variable t, we may assume that $t_{\text{end}} = 2\pi$, giving a periodic analytic parameterization

$$t \in [0, 2\pi] \mapsto (x^*(t), y^*(t)) \text{ of } \Gamma(h), \tag{3.2}$$

with a winding number equal to ± 1 with respect to any point inside $\Gamma(h)$.

[8]This theorem provides convergence bounds w.r.t. the $\|\cdot\|_\infty$ norm, but the same bounds actually hold for the $\|\cdot\|_{\ell^1}$ norm since the proof consists in bounding the Fourier coefficients.

The approximation of $\Gamma(h)$ consists of two steps:

- First, Algorithm OvalApproxInit presented in Section 3.1 computes an initial approximation $(x^\circ, y^\circ) \in (C^0_{2\pi})^2$ of (x^*, y^*), with TPs x°, y° of a low degree N_0. The accuracy of this initial guess (and therefore N_0) should ensure the convergence of the Newton iterations in the next routine OvalApproxRefine. It only depends on the system under consideration (i.e., H, Σ and h) and not on the (possibly very high) final precision that we target. Hence, the uniform convergence of the output of Algorithm OvalApproxInit (without asymptotic estimate) is sufficient for our complexity analysis.
- Next, Algorithm OvalApproxRefine refines this initial guess by computing TPs $\check{x}, \check{y}$ of higher degree N. As explained in Section 3.2, this is done by applying Newton iterations to the initial guess so as to "project" it onto $\Gamma(h)$. We provide a complexity analysis and an asymptotic convergence estimate of $(\check{x}, \check{y})$ to $\Gamma(h)$ in terms of the degree N.

We emphasize the fact that the convergence results in this section are asymptotic only. Indeed, our objective is to prove the quasi-linear arithmetic complexity of this approximation procedure with respect to the number of correct digits. On the other hand, computing effective error bounds is the role of the algorithms presented in the Sections 4 and 5.

3.1 Algorithm OvalApproxInit: initial approximation of the curve

Algorithm OvalApproxInit is given an initial point $(x_{ini}, y_{ini}) \in \Gamma(h) \cap \Sigma$ on the transversal. It first calls a numerical ODE solver to approximately solve the differential system (3.1) until the first return onto Σ. It provides us with a continuous function $t \mapsto (\tilde{x}(t), \tilde{y}(t))$, which we assume to be rescaled over the time interval $[0, 2\pi]$. Finally, in order to obtain smooth approximations to (x^*, y^*), cf. (3.2), we interpolate $(\tilde{x}, \tilde{y})$ at evenly spaced points and obtain low degree TPs x°, y° (lines 2–5) representing our analytic initial guess for $\Gamma(h)$. The proof of the following Proposition is postponed to the end of this section.

Proposition 3.1. *Under Assumption 3.2, Algorithm OvalApproxInit$(H, \Sigma, x_{ini}, y_{ini}, N_0)$ computes degree N_0 TPs x° and y° that converge uniformly to the rescaled parameterization $(x^*, y^*) \in (C^0_{2\pi})^2$ of $\Gamma(h)$:*

$$\|x^\circ - x^*\|_\infty, \|y^\circ - y^*\|_\infty \to 0 \quad as \quad N_0 \to \infty.$$

Algorithm 2 OvalApproxInit$(H, \Sigma, x_{ini}, y_{ini}, N_0)$

Input: Potential function H, transversal Σ, initial point (x_{ini}, y_{ini}), and approximation degree $N_0 \in \mathbb{N}$
Output: Degree N_0 TPs x°, y° approximating $\Gamma(h)$

▷ *Approximation of $\Gamma(h)$, computed with precision parameter N_0^{-1}*
1: $(\tilde{x}, \tilde{y}) \leftarrow \mathrm{DSolve}(H, \Sigma, x_{ini}, y_{ini}, N_0^{-1})$
2: **for** $j = 0$ **to** $2N_0$ **do**
3: $\quad x^\circ_j \leftarrow \tilde{x}\left(\frac{2j\pi}{2N_0+1}\right), y^\circ_j \leftarrow \tilde{y}\left(\frac{2j\pi}{2N_0+1}\right)$
4: **end for**
▷ *Interpolate the points*
5: $x^\circ \leftarrow \mathrm{FFT}_{N_0}\left((x^\circ_j)_{j=0}^{2N_0}\right), y^\circ \leftarrow \mathrm{FFT}_{N_0}\left((y^\circ_j)_{j=0}^{2N_0}\right)$
6: **return** (x°, y°)

3.1.1 The numerical solver. We need a numerical routine $\textsc{DSolve}(H, \Sigma, x_{ini}, y_{ini}, \delta)$ that integrates the ODE system (3.1) with some precision parameter δ, starting from the initial point $(x_{ini}, y_{ini}) \in \Gamma(h) \cap \Sigma$. This routine has to detect numerically the first return onto Σ and stop at the corresponding time t_{end}. Moreover, we assume that this routine rescales the time interval $[0, t_{\text{end}}]$ to $[0, 2\pi]$, in order to work on a fixed domain of definition.

Most scientific programming languages offer libraries with sophisticated explicit or implicit iterative schemes to compute a time-discretized solution to (3.1), with excellent timings for the moderate precision, e.g., the standard double precision in most cases, needed at this stage. Here, the precision parameter δ may represent a maximal time step. Moreover, the values between two consecutive time steps are usually automatically interpolated, so as to define a continuous function $t \mapsto (\tilde{x}(t), \tilde{y}(t))$.

3.1.2 Detection of the first return onto Σ. Another feature often proposed by numerical libraries is the so-called *event detection*: the solver stops when a user-defined continuous function of the dependent and independent variables vanishes (the "event"). In our case, intersecting the transversal line Σ is trivially recast as a linear equation in the Cartesian coordinates. Note, however, that the event is already satisfied at $t = 0$ (up to numerical errors), since $(x_{ini}, y_{ini}) \in \Sigma$, which may cause the solver to stop before detecting the actual first return. A more robust solution is to use a polar parameterization with respect to a reference point (x_r, y_r) of choice inside $\Gamma(h)$, that is the ODE system:

$$\begin{cases} x(t) = x_r + \rho(t)\cos\theta(t), \\ y(t) = y_r + \rho(t)\sin\theta(t), \end{cases} \quad \text{with} \quad \begin{cases} \dot{\rho} = \sin\theta \dfrac{\partial H}{\partial x}(x,y) - \cos\theta \dfrac{\partial H}{\partial y}(x,y), \\ \dot{\theta} = \dfrac{\cos\theta \frac{\partial H}{\partial x}(x,y) + \sin\theta \frac{\partial H}{\partial y}(x,y)}{\rho}. \end{cases}$$

The stopping condition is when $\theta(t)$ reaches $\theta(0) + \varsigma 2\pi$, with $\varsigma = \pm 1$ the winding number of (x^*, y^*) with respect to (x_r, y_r).

3.1.3 Convergence of the ODE solver and a proof of Proposition 3.1. To prove Proposition 3.1, we need the following assumption on the numerical routine DSolve to ensure that the numerical solution $(\tilde{x}, \tilde{y})$, computed with a precision parameter $\delta = 1/N_0$, converges fast enough to (x^*, y^*) to balance the overestimation factor due to the degree N_0 trigonometric interpolation. In practice, such a mild convergence assumption in $O(\delta)$ is guaranteed by most iterative schemes, even a simple Euler scheme with time step δ.

Assumption 3.2. *The numerical routine $\textsc{DSolve}(H, \Sigma, x_{ini}, y_{ini}, \delta)$ computes a continuous approximation $t \in [0, 2\pi] \mapsto (\tilde{x}(t), \tilde{y}(t))$ of the rescaled parameterization (x^*, y^*), cf. (3.2), of $\Gamma(h)$, such that:*

$$\|\tilde{x} - x^*\|_\infty, \|\tilde{y} - y^*\|_\infty = O(\delta) \quad \textit{as} \quad \delta \to 0.$$

We can now prove the convergence of Algorithm OvalApproxInit.

Proof of Proposition 3.1. Let the symbol z denote either x or y. Let $\eta = \tilde{z} - z^*$ be the error between z^* and the approximate solution $\tilde{z}$ computed by DSolve in line 1. Thanks to the linearity of the operator $\mathcal{I}_{N_0}$ (see (2.1)), the error between the interpolant z° computed in lines 2–5 and z^* can be bounded from above by the error of interpolation for z^*, plus the interpolation of the error η :

$$\|z^\circ - z^*\|_\infty = \|\mathcal{I}_{N_0}(\tilde{z}) - z^*\|_\infty \leqslant \|\mathcal{I}_{N_0}(z^*) - z^*\|_\infty + \|\mathcal{I}_{N_0}(\eta)\|_\infty.$$

• *Error of interpolation.* Since z^* is 2π-periodic and continuously differentiable, the trigonometric interpolants $\mathcal{I}_{N_0}(z^*)$ converge uniformly to z^* [Zygmund 2002, Chap. X].

• *Interpolation of error.* By Assumption 3.2, η is a continuous function over $[0, 2\pi]$ and $\|\eta\|_\infty$ converges uniformly to 0 in $O(N_0^{-1})$ as the precision parameter $\delta = N_0^{-1}$ tends to 0. From Section 2.1, we have

$$\|\mathcal{I}_{N_0}(\eta)\|_\infty \leqslant \Lambda_{N_0}\|\eta\|_\infty = O(\log(N_0)/N_0) \to 0 \text{ as } N_0 \to \infty. \qquad \square$$

3.2 Algorithm OvalApproxRefine: refining the curve by projection

As its name suggests, the role of Algorithm OvalApproxRefine is to refine the initial guess (x°, y°) with TPs $(\check{x}, \check{y})$ of higher degree $N \geqslant N_0$, yielding an analytic approximate parameterization of $\Gamma(h)$ of very high accuracy.

By solving algebraic equations only, it is in general not possible to recover the parameterization $(x^*, y^*) \in (C^0_{2\pi})^2$ satisfying the differential system (3.1). However, the value of $\mathfrak{I}(h)$ is independent of the parameterization used for $\Gamma(h)$. Therefore, we define another one, $(x^\#, y^\#) \in (C^0_{2\pi})^2$, called the *projected parameterization*, which Algorithm OvalApproxRefine approximates using Newton's method on points sampled on the initial curve (x°, y°).

Definition 3.3. Let $(x^\circ, y^\circ) \in (C^0_{2\pi})^2$ be an approximate parameterization of $\Gamma(h)$, $(u^\circ, v^\circ) \in (C^0_{2\pi})^2$, and $\varepsilon > 0$. If the equation

$$H(x^\circ(t) + s(t)\,u^\circ(t), y^\circ(t) + s(t)\,v^\circ(t)) = h, \qquad t \in [0, 2\pi], \tag{3.3}$$

has a unique solution $s^\# \in C^0_{2\pi}$ with $\|s^\#\|_\infty \leqslant \varepsilon$, called the *shift*, then the curve $(x^\#, y^\#) \in (C^0_{2\pi})^2$,

$$\begin{cases} x^\#(t) = x^\circ(t) + s^\#(t)\,u^\circ(t), \\ y^\#(t) = y^\circ(t) + s^\#(t)\,v^\circ(t), \end{cases} \tag{3.4}$$

is called the *projection* of (x°, y°) onto $\Gamma(h)$ w.r.t. the direction (u°, v°).

The following Lemma, whose proof is postponed to Appendix B, establishes the good properties of the projected parameterization when the initial guess (x°, y°) returned by OvalApproxInit is sufficiently close to $\Gamma(h)$.

Lemma 3.4. *There exists $\eta > 0$ (depending on H and h) such that if (x°, y°) and (u°, v°) are analytic and η-close*[9] *to (x^*, y^*) and $\frac{\nabla H(x^*,y^*)}{\|\nabla H(x^*,y^*)\|_2}$, respectively, in $(C^0_{2\pi})^2$, then:*

(i) $(x^\#, y^\#) \in (C^0_{2\pi})^2$ is well-defined and it is an analytic parameterization of $\Gamma(h)$ with same orientation as (x^*, y^*).

(ii) For all $t \in [0, 2\pi]$, letting $(x, y) = (x^\circ(t), y^\circ(t))$ and $(u, v) = (u^\circ(t), v^\circ(t))$, the Newton operator associated to Equation (3.3) at t is well-defined:

$$\mathcal{N}_{x,y,u,v}(s) = s - \frac{H(x + su, y + sv) - h}{u\frac{\partial H}{\partial x}(x + su, y + sv) + v\frac{\partial H}{\partial y}(x + su, y + sv)}, \tag{3.5}$$

and the Newton iterations $s^{(k)} = \mathcal{N}^k_{x,y,u,v}(0)$ converge quadratically fast to $s^\#(t)$.

Based on those properties, Algorithm OvalApproxRefine discretizes the initial guess (x°, y°) over the equispaced grid (t_j) of size $2N + 1$ (line 1), then applies several Newton iterations (3.5), implemented in lines 6–8, and finally interpolates the obtained values s_j in order to construct a degree N TP $\check{s}$ approximating the shift $s^\#$. This defines the approximate parameterization $(\check{x}, \check{y})$ of $\Gamma(h)$:

$$\begin{cases} \check{x}(t) = x^\circ(t) + \check{s}(t)\,u^\circ(t), \\ \check{y}(t) = y^\circ(t) + \check{s}(t)\,v^\circ(t), \end{cases} \qquad t \in [0, 2\pi]. \tag{3.6}$$

[9] It means that $\max(\|x^\circ - x^*\|_\infty, \|y^\circ - y^*\|_\infty) \leqslant \eta$.

Algorithm 3 OvalApproxRefine$(H, h, x^\circ, y^\circ, N)$

Input: Potential function H, level h, initial approximation (x°, y°) and degree $N \in \mathbb{N}$
Output: Degree N TPs $u^\circ, v^\circ, \check{s}$ defining the approximate parameterization $(\check{x}, \check{y})$ of $\Gamma(h)$ as in Eq. (3.6)

▷ *Sample initial points*
1: $(x_j)_{j=0}^{2N} \leftarrow \mathrm{IFFT}_N(x^\circ), \quad (y_j)_{j=0}^{2N} \leftarrow \mathrm{IFFT}_N(y^\circ)$
▷ *Apply* $\lceil \log N \rceil$ *Newton iterations*
2: **for** $j = 0$ **to** $2N$ **do**
3: $\quad (u_j, v_j) \leftarrow \frac{\nabla H(x_j, y_j)}{\|\nabla H(x_j, y_j)\|_2}$
4: $\quad s_j \leftarrow 0$
5: $\quad$ **for** $k = 1$ **to** $\lceil \log N \rceil$ **do**
6: $\quad\quad \delta \leftarrow \dfrac{h - H(x_j, y_j)}{u_j \frac{\partial H}{\partial x}(x_j, y_j) + v_j \frac{\partial H}{\partial y}(x_j, y_j)}$
7: $\quad\quad s_j \leftarrow s_j + \delta$
8: $\quad\quad (x_j, y_j) \leftarrow (x_j, y_j) + \delta\,(u_j, v_j)$
9: $\quad$ **end for**
10: **end for**
▷ *Interpolate*
11: $u^\circ \leftarrow \mathrm{FFT}_N\left((u_j)_{j=0}^{2N}\right), \quad v^\circ \leftarrow \mathrm{FFT}_N\left((v_j)_{j=0}^{2N}\right)$
12: $\check{s} \leftarrow \mathrm{FFT}_N\left((s_j)_{j=0}^{2N}\right)$
13: **return** $(u^\circ, v^\circ), \check{s}$

The following proposition establishes the exponential convergence of this method w.r.t. the approximation degree N and its quasi-linear arithmetic complexity.

Proposition 3.5. *There exists $\eta > 0$ (depending on H and h) such that if the analytic initial guess $(x^\circ, y^\circ) \in (C^0_{2\pi})^2$ is η-close to (x^*, y^*), then there is $\kappa > 1$ such that for any $N \in \mathbb{N}$, Algorithm OvalApproxRefine$(H, h, x^\circ, y^\circ, N)$ computes a degree N TP $\check{s}$ in $O(N \log N)$ arithmetic operations, and the corresponding curve $(\check{x}, \check{y})$, cf. (3.6), converges exponentially fast to the projected parameterization $(x^\#, y^\#)$ of $\Gamma(h)$:*

$$\|\check{s} - s^\#\|_\infty, \|\check{x} - x^\#\|_\infty, \|\check{y} - y^\#\|_\infty = O(\kappa^{-N}) \quad \text{as} \quad N \to \infty.$$

Proof. Let $\eta' > 0$ denote the η of Lemma 3.4. Clearly, one can chose an $\eta \in (0, \eta']$ such that if (x°, y°) is η-close to (x^*, y^*), then $(u^\circ, v^\circ) = \mathcal{I}_N\left(\frac{\nabla H(x^\circ, y^\circ)}{\|\nabla H(x^\circ, y^\circ)\|_2}\right)$ (lines 3 and 11) is η'-close to $\frac{\nabla H(x^*, y^*)}{\|\nabla H(x^*, y^*)\|_2}$, since by assumption ∇H does not vanish over $\Gamma(h)$. Then the analytic shift $s^\#$ and the projected parameterization $(x^\#, y^\#)$ are well-defined. As in the proof of Proposition 3.1, we bound the total error with the sum of the error of interpolation and the interpolation of the error:

$$\|\check{s} - s^\#\|_\infty \leqslant \|\mathcal{I}_N(s^\#) - s^\#\|_\infty + \Lambda_N \sup_{0 \leqslant j \leqslant 2N} |\check{s}(t_j) - s^\#(t_j)|,$$

where $t_j = \frac{2j\pi}{2N+1}$ are the equispaced interpolation points.

• *Error of interpolation.* Since $s^\# \in C^0_{2\pi}$ is 2π-periodic and analytic over $\mathbb{R}$ by Lemma 3.4, there is a $\rho > 0$ such that the open horizontal strip $\mathbb{R} + (-\rho, \rho)i$ avoids all possible singularities of $s^\#$ in the

complex plane. Then the interpolants $\mathcal{I}_N(s^\#)$ converge uniformly to $s^\#$ over $\mathbb{R}$ in $O(\kappa^{-N})$ for any $\kappa \in (1, e^\rho)$ (see [Wright et al. 2015, Thm. 4.2]).

• *Interpolation of the error.* Since the interpolation in line 12 uses approximations s_j of the exact points $s^\#(t_j)$, the resulting "interpolation of the error" depends on the accuracy reached after $k = \lceil \log N \rceil$ Newton iterations (lines 2–10). By Eq. (B.1) in the proof of Lemma 3.4, we have:

$$|s_j - s^\#(t_j)| \leqslant |\mathcal{N}^k_{x_j,y_j,u_j,v_j}(0) - s^\#(t_j)| \leqslant \frac{|s^\#(t_j)|}{2^{2^k-1}} \leqslant 2^{-N}\frac{\|s^\#\|_\infty}{2}.$$

• *Combining both errors,* we conclude that $\|s - s^\#\|_\infty = O(\kappa^{-N})$ for any $\kappa \in (1, \min(e^\rho, 2))$. It is also clear from Eq. (3.4) and (3.6) that the same convergence happens for $(\check{x}, \check{y})$ to $(x^\#, y^\#)$.

• *Complexity.* The sampling of initial points and reinterpolation through FFT_N/IFFT_N requires $O(N \log N)$ arithmetic operations. For the projection step (lines 2–10), each of the $2N + 1$ points is refined through $\lceil \log N \rceil$ Newton iterations, where the cost of a single iteration is $O(1)$. This justifies the overall complexity in $O(N \log N)$ arithmetic operations. □

4 A POSTERIORI VALIDATION OF THE OVAL $\Gamma(h)$

The purpose of Algorithm OvalValid presented in this section is to enclose the oval $\Gamma(h)$ in a validated tube $T(\check{x}, \check{y}, u^\circ, v^\circ, \varepsilon)$ around the approximate parameterization $(\check{x}, \check{y})$ (3.6) containing 2π-periodic curves:

$$T(\check{x}, \check{y}, u^\circ, v^\circ, \varepsilon) = \left\{ (x, y) \in (C^0_{2\pi})^2 \;\middle|\; \exists s \in C^0_{2\pi}, \|s\|_\infty \leqslant \varepsilon \text{ and } \begin{cases} x(t) = \check{x}(t) + s(t)u^\circ(t) \\ y(t) = \check{y}(t) + s(t)v^\circ(t) \end{cases} \right\}.$$

The width ε of such a tube is rigorously computed by using the fixed-point based a posteriori approach presented in Section 2.3 to bound the distance between $\check{s}$ computed by OvalApproxRefine and the exact shift $s^\#$ defining the parameterization $(x^\#, y^\#)$ of $\Gamma(h)$ (see Definition 3.3). Under a few additional conditions also checked by OvalValid (winding number, etc.[10]), this ensures that $(x^\#, y^\#)$ exists, satisfies Lemma 3.4, and lies in the tube $T(\check{x}, \check{y}, u^\circ, v^\circ, \varepsilon)$.

Using Equation (3.3), we construct a polynomial *Newton-like operator* $\mathcal{N} \in C^0_{2\pi}[X]$ acting on a function $s \in C^0_{2\pi}$:

$$\mathcal{N}(s) = s - \check{a}\ (H\,(x^\circ + s\,u^\circ, y^\circ + s\,v^\circ) - h)\,, \tag{4.1}$$

for which $s^\#$ is the unique fixed point in some small neighborhood of $\check{s}$. Here,

$$\check{a}(t) \approx \frac{1}{u^\circ(t)\,\frac{\partial H}{\partial x}(\check{x}(t), \check{y}(t)) + v^\circ(t)\,\frac{\partial H}{\partial y}(\check{x}(t), \check{y}(t))},$$

is a TP built by interpolation to make the derivative of $\mathcal{N}$,

$$\mathcal{N}'(s) = 1 - \check{a}\left(u^\circ\,\frac{\partial H}{\partial x}(x^\circ + s\,u^\circ, y^\circ + s\,v^\circ) + v^\circ\,\frac{\partial H}{\partial y}(x^\circ + s\,u^\circ, y^\circ + s\,v^\circ)\right),$$

small in the neighborhood of $s = \check{s}$ (that is, geometrically, in the neighborhood of the approximate curve $(\check{x}, \check{y})$).

Contrary to the a posteriori validation for the division presented in Section 2.3.1, the Newton-like operator $\mathcal{N}$ in (4.1) is not of degree 1 in s unless H itself has degree 1. As a consequence, the value of $\|\mathcal{N}'(s)\|_\infty$ depends on s, so that the conditions of Theorem 2.5 must be satisfied with a finite radius $r > 0$. We propose a generic subroutine NewtonBall (not specific to the oval validation problem considered here) in Appendix C to automate this task. Specifically, given:

[10]Although such kinds of properties may not always be checked by other existing approaches, it turns out to be necessary in order to move towards formally checked computed-assisted proofs.

- a procedure computing an upper bound $\lambda(r)$ for $\sup_{\|s-\check{s}\|_\infty \leqslant r} \|\mathcal{N}'(s)\|_\infty$ for any $r \geqslant 0$,
- an upper bound d for the defect $\|\mathcal{N}(\check{s}) - \check{s}\|_\infty$,

NewtonBall(d, λ) either computes a radius r and a Lipschitz constant $\lambda = \lambda(r)$ satisfying the hypotheses of Theorem 2.5, or returns an error.

Algorithm 4 OvalValid$(H, h, x_{ini}, y_{ini}, \varepsilon_0, x_r, y_r, \varsigma, x^\circ, y^\circ, u^\circ, v^\circ, \check{s}, N)$

Input: $H, h, \varsigma = \pm 1$, $(x_{ini}, y_{ini}) \in \mathbb{R}^2$, $\varepsilon_0 > 0$, $(x_r, y_r) \in \mathbb{R}^2$, and degree N TPs $x^\circ, y^\circ, u^\circ, v^\circ, \check{s}$
Output: a bound $\varepsilon > 0$ defining a tube $T(\check{x}, \check{y}, u^\circ, v^\circ, \varepsilon)$ around $(\check{x}, \check{y})$ containing $(x^\#, y^\#)$

▷ *Build TP $\check{a}$ defining the Newton-like operator $\mathcal{N}$*
$(x_j)_{j=0}^{2N} \leftarrow \mathrm{IFFT}_N(\check{x})$ and $(y_j)_{j=0}^{2N} \leftarrow \mathrm{IFFT}_N(\check{y})$
$(u_j)_{j=0}^{2N} \leftarrow \mathrm{IFFT}_N(u^\circ)$ and $(v_j)_{j=0}^{2N} \leftarrow \mathrm{IFFT}_N(v^\circ)$
$\check{a} \leftarrow \mathrm{FFT}_N\left(\left(\dfrac{1}{u_j \frac{\partial H}{\partial x}(x_j, y_j) + v_j \frac{\partial H}{\partial y}(x_j, y_j)}\right)_{j=0}^{2N}\right)$
▷ *Bound operator norm of $\mathcal{N}$, using rigorous operations with intervals and RTAs*
$\boldsymbol{x} \leftarrow x^\circ + \check{s}\, u^\circ$ and $\boldsymbol{y} \leftarrow y^\circ + \check{s}\, v^\circ$ ▷ $(\boldsymbol{x}, \boldsymbol{y})$ *RTAs for* $(\check{x}, \check{y})$
$\boldsymbol{x}' \leftarrow x^{\circ\prime} + \check{s}'\, u^\circ + \check{s}\, u^{\circ\prime}$ and $\boldsymbol{y}' \leftarrow y^{\circ\prime} + \check{s}'\, v^\circ + \check{s}\, v^{\circ\prime}$
$d \leftarrow \textsc{Bound}(\boldsymbol{s})$ where $\boldsymbol{s} \leftarrow \check{a}\,(h - H(\boldsymbol{x}, \boldsymbol{y}))$ ▷ *defect*
$\lambda_0 \leftarrow \textsc{Bound}(\boldsymbol{\ell}_0)$ where $\boldsymbol{\ell}_0 \leftarrow 1 - \check{a}\left(u^\circ \frac{\partial H}{\partial x}(\boldsymbol{x}, \boldsymbol{y}) + v^\circ \frac{\partial H}{\partial y}(\boldsymbol{x}, \boldsymbol{y})\right)$
$\alpha \leftarrow \textsc{Bound}(\check{a})$
$X \leftarrow \textsc{Bound}(\boldsymbol{x}) \cdot [-1, 1]$ and $Y \leftarrow \textsc{Bound}(\boldsymbol{y}) \cdot [-1, 1]$
$U \leftarrow \textsc{Bound}(u^\circ) \cdot [-1, 1]$ and $V \leftarrow \textsc{Bound}(v^\circ) \cdot [-1, 1]$
function $\lambda(r)$ ▷ *Lipschitz constant of $\mathcal{N}$ over $\overline{B}(\check{s}, r)$ in $C^0_{2\pi}$*
  $X_r \leftarrow X + rU$ and $Y_r \leftarrow Y + rV$
  $\lambda_1 \leftarrow \mathrm{mag}\left(U^2 \frac{\partial^2 H}{\partial x^2}(X_r, Y_r) + 2UV \frac{\partial^2 H}{\partial x \partial y}(X_r, Y_r) + V^2 \frac{\partial^2 H}{\partial y^2}(X_r, Y_r)\right)$
  return $\lambda_0 + \alpha r \lambda_1$
end function
$(r^+, \lambda^+) \leftarrow \textsc{NewtonBall}(d, \lambda)$ ▷ *Find stable ball*
▷ *Validated error bound ε*
$\varepsilon \leftarrow d/(1 - \lambda^+)$
▷ *Check initial conditions*
if $(\boldsymbol{x}(0) - \varepsilon u^\circ(0) - x_{ini})^2 + (\boldsymbol{y}(0) - \varepsilon v^\circ(0) - y_{ini})^2 > \varepsilon_0^2$ **or**
$(\boldsymbol{x}(0) + \varepsilon u^\circ(0) - x_{ini})^2 + (\boldsymbol{y}(0) + \varepsilon v^\circ(0) - y_{ini})^2 > \varepsilon_0^2$, **then return** *FAIL* **end if**
▷ *Check the winding number of the curve*
$\boldsymbol{r2} \leftarrow (\boldsymbol{x} - x_r)^2 + (\boldsymbol{y} - y_r)^2$
if $\neg\, \textsc{IsPositive}\left(\boldsymbol{r2} - \varepsilon^2 \textsc{Bound}(u^{\circ 2} + v^{\circ 2})\right)$ **then return** *FAIL* **end if**
$\varsigma \leftarrow \dfrac{1}{2\pi} \displaystyle\int_0^{2\pi} \frac{(\boldsymbol{x}(t) - x_r)\boldsymbol{y}' - (\boldsymbol{y}(t) - y_r)\boldsymbol{x}'}{\boldsymbol{r2}(t)}\, \mathrm{d}t$
if $\varsigma \cap \mathbb{Z} \neq \{\varsigma\}$ **then return** *FAIL* **end if**
return ε

Proposition 4.1. *Given a point (x_r, y_r) inside the oval $\Gamma(h)$, and a ball $\overline{B}((x_{ini}, y_{ini}), \varepsilon_0)$ enclosing initial conditions such that it intersects $H^{-1}(h)$ in the $\Gamma(h)$ component only, Algorithm* OvalValid$(H, h,$

$x_{ini}, y_{ini}, \varepsilon_0, x_r, y_r, x^\circ, y^\circ, u^\circ, v^\circ, \check{s}, N$), if it does not fail, computes ε such that $s^\#$ of Definition 3.3 exists, satisfies Lemma 3.4 with winding number around (x_r, y_r) equal to ς, and $\|\check{s} - s^\#\|_\infty \leqslant \varepsilon$.

Moreover, for a sufficiently large degree N, the process never fails and returns a bound ε that converges exponentially fast to 0 w.r.t. N, in $O(N \log(N))$ arithmetic operations.

Proof. We apply Theorem 2.5 over the space $C^0_{2\pi}$ to compute an upper bound ε for $\|s^\#\|_\infty$, where $s^\# \in C^0_{2\pi}$ is solution to (3.3). First (lines 1–3), the Newton-like operator $\mathcal{N}$ is built as in Equation (4.1) by computing a TP $\check{a}$ interpolating the function which represents the inverse differential. Next, the algorithm computes a rigorous bound d on the defect $\|\mathcal{N}(\check{s}) - \check{s}\|_\infty$ (line 6), and produces a procedure λ s.t. $\lambda(r)$ is a Lipschitz constant for $\mathcal{N}$ over $\overline{B}(\check{s}, r)$ in $C^0_{2\pi}$ w.r.t. to the $\|\cdot\|_\infty$ norm (lines 7–15). The formula for $\lambda(r)$ in line 14 comes from Lemma 2.6, with U, V enclosures for the range of u°, v°, and X_r, Y_r enclosures for the range of $x^\circ + su^\circ = \check{x} + (s - \check{s})u^\circ$ and $y^\circ + sv^\circ = \check{y} + (s - \check{s})v^\circ$, for all $s \in \overline{B}(\check{s}, r)$. After that, the auxiliary routine NewtonBall (line 16) described in Appendix C rigorously computes a suitable radius r^+ satisfying the hypotheses of Theorem 2.5, if possible. We thus deduce the existence and uniqueness of $s^\# \in C^0_{2\pi}$ (and hence $(x^\#, y^\#)$) together with a validated bound ε (line 17).

• *Initial conditions.* By hypothesis, the ball $\overline{B}((x_{ini}, y_{ini}), \varepsilon_0)$ in $\mathbb{R}^2$ intersects $H^{-1}(h)$ in its component $\Gamma(h)$ only. Hence, by continuity, checking that $(x^\#(0), y^\#(0))$ lies in that ball is sufficient to ensure that $(x^\#(t), y^\#(t))$ belongs to the component $\Gamma(h)$ for all $t \in [0, 2\pi]$. Line 19 does this by checking that both endpoints of the transversal segment of the tube at time 0 belong to that ball.

• *Winding number.* First, lines 20–21 ensure that the trajectories in the tube $T(\check{x}, \check{y}, u^\circ, v^\circ, \varepsilon)$ all avoid the point (x_r, y_r) inside $\Gamma(h)$. Then, the winding number of $(\check{x}, \check{y})$ is rigorously computed in line 22 and compared to the reference value ς at line 23. Since the tube avoids (x_r, y_r), the winding number remains the same for all curves in it (so in particular $(x^\#, y^\#)$).

• *Analyticity.* The Banach fixed-point based argument used previously only guarantees the continuity of $(x^\#, y^\#)$. To gain more regularity, we need to consider a larger function space, namely the Banach space of continuous 2π-periodic functions defined over a horizontal strip $\mathcal{S}_\rho := \mathbb{R} + [-\rho, \rho]i$ of $\mathbb{C}$ for some small $\rho > 0$, equipped with the norm: $\|\varphi\|_{\mathcal{S}_\rho} = \sup_{z \in \mathcal{S}_\rho} |\varphi(z)|$[11]. The a posteriori validation argument based on Theorem 2.5 applies the same way, except that we need to replace $\|\cdot\|_\infty$ with $\|\cdot\|_{\mathcal{S}_\rho}$ in all the bounds defining d and $r \mapsto \lambda(r)$. If we choose ρ small enough, the inequalities $\lambda(r^+) < 1$ and $d + r^+\lambda(r^+) < r^+$ guaranteed by NewtonBall still hold for the same r^+, which means that Theorem 2.5 now proves the existence of a unique continuous periodic solution to (3.3) extending $s^\#$ over $\mathcal{S}_\rho$. Moreover, the iterations $\mathcal{N}^k(0)$ are all trigonometric polynomials (hence analytic) and they converge uniformly to $s^\#$ over $\mathcal{S}_\rho$ as $k \to \infty$. As a consequence, the limit $s^\#$ is analytic over $\mathbb{R}$.

• *Convergence.* Here we outline the main arguments and leave the technical details to the reader. First, we know from Proposition 3.5 that the input $\check{s}$ computed by OvalApproxRefine tends exponentially fast to $s^\#$. This holds for the $\|\cdot\|_\infty$ norm, but also for the $\|\cdot\|_{\ell^1}$ norm since they are TPs of degree $O(N)$. As a consequence, the computed bounds d and λ_0 tend to 0 exponentially fast, while the other bounds α, X, Y, U, V and λ_1 (for given r) tend to the mathematical values obtained by replacing the approximations $\check{s}, \check{x}, \check{y}$ by $s^\#, x^\#, y^\#$. Therefore, in the routine NewtonBall, for N sufficiently large, the initial radius $r_{\max}$ computed in line 1 is bounded away from 0 and satisfies $d + \lambda(r_{\max})r_{\max} < r_{\max}$. Hence, NewtonBall necessarily returns an r satisfying $d + \lambda(r)r < r$, and then OvalValid computes $\varepsilon = d/(1 - \lambda^+) \leqslant 2d$, which tends exponentially fast to 0. To conclude, similar arguments show that the remaining tests (initial conditions and winding number) never fail for N sufficiently large.

[11] Note that this argument is purely theoretical: there is no need to reimplement Algorithm OvalValid to actually compute with this norm $\|\cdot\|_{\mathcal{S}_\rho}$.

• *Complexity.* The asymptotic complexity w.r.t. the degree N is determined by the FFT_N/IFFT_N routines used for evaluation, interpolation, RTA multiplications, the RTA positivity test in line 21 and the RTA division in line 22. Note that the execution time of NewtonBall does not depend on N, since the function $r \mapsto \lambda(r)$ (lines 11–15) involves intervals rather than RTAs. □

5 RIGOROUS ENCLOSURE OF THE ABELIAN INTEGRAL $\mathfrak{I}(h)$

According to Proposition 4.1, the tube $T(\check{x}, \check{y}, u^\circ, v^\circ, \varepsilon)$ computed by OvalValid contains a unique analytic parameterization $(x^\#, y^\#)$ of $\Gamma(h)$ with correct winding number, so that we have:

$$\mathfrak{I}(h) = \int_0^{2\pi} \frac{P(x^\#(t), y^\#(t))y^{\#\prime}(t) - Q(x^\#(t), y^\#(t))x^{\#\prime}(t)}{\mu(x^\#(t), y^\#(t))}\,\mathrm{d}t. \tag{5.1}$$

In order to provide an interval enclosure for $\mathfrak{I}(h)$, we must bound the difference between the exact value given by Equation (5.1) and the corresponding integral taken along the approximate curve $(\check{x}, \check{y})$ defined by $\check{s}$:

$$I_{\check{s}} = \int_0^{2\pi} \frac{P(\check{x}(t), \check{y}(t))\check{y}'(t) - Q(\check{x}(t), \check{y}(t))\check{x}'(t)}{\mu(\check{x}(t), \check{y}(t))}\,\mathrm{d}t, \tag{5.2}$$

which can be rigorously computed using operations on RTAs. To do so, we make use of Stokes' theorem and bound the resulting term:

$$c(x, y)\,\mathrm{d}x \wedge \mathrm{d}y = \left[\frac{\partial(P/\mu)}{\partial x}(x, y) + \frac{\partial(Q/\mu)}{\partial y}(x, y)\right] \mathrm{d}x \wedge \mathrm{d}y, \tag{5.3}$$

over the region located between the two curves $(\check{x}, \check{y})$ and $(x^\#, y^\#)$, which is safely overapproximated by the tube $T(\check{x}, \check{y}, u^\circ, v^\circ, \varepsilon)$.

Algorithm 5 AbIntValidQuad$(P, Q, \mu, x^\circ, y^\circ, u^\circ, v^\circ, \check{s}, \varepsilon, N)$

Input: P, Q, μ as in Problem 1.1, degree N TPs $x^\circ, y^\circ, u^\circ, v^\circ, \check{s}$, and $\varepsilon > 0$ computed by OvalValid
Output: interval enclosure $\boldsymbol{I}$ for $\mathfrak{I}(h)$ (1.1)

▷ *Compute rigorous integral along approximate curve using RTAs*
1: $\boldsymbol{x} \leftarrow x^\circ + \check{s}\,u^\circ$ and $\boldsymbol{y} \leftarrow y^\circ + \check{s}\,v^\circ$ ▷ $(\boldsymbol{x}, \boldsymbol{y})$ *RTAs for* $(\check{x}, \check{y})$
2: $\boldsymbol{x}' \leftarrow x^{\circ\prime} + \check{s}'\,u^\circ + \check{s}\,u^{\circ\prime}$ and $\boldsymbol{y}' \leftarrow y^{\circ\prime} + \check{s}'\,v^\circ + \check{s}\,v^{\circ\prime}$
3: $\boldsymbol{P} \leftarrow P(\boldsymbol{x}, \boldsymbol{y})$, $\boldsymbol{Q} \leftarrow Q(\boldsymbol{x}, \boldsymbol{y})$, and $\boldsymbol{\mu} \leftarrow \mu(\boldsymbol{x}, \boldsymbol{y})$
4: $\boldsymbol{E} \leftarrow (\boldsymbol{P}\,\boldsymbol{y}' - \boldsymbol{Q}\,\boldsymbol{x}') \,/\, \boldsymbol{\mu}$
5: $\boldsymbol{I}_{\check{s}} \leftarrow \int_0^{2\pi} \boldsymbol{E}(t)\,\mathrm{d}t$
▷ *Compute integration error in the tube using Stokes' theorem*
6: $\boldsymbol{x}_\varepsilon \leftarrow (x^\circ + \check{s}\,u^\circ,\ \varepsilon\,\mathrm{Bound}(u^\circ)))$ and $\boldsymbol{y}_\varepsilon \leftarrow (y^\circ + \check{s}\,v^\circ,\ \varepsilon\,\mathrm{Bound}(v^\circ)))$
7: $c \leftarrow \frac{\partial(P/\mu)}{\partial x} + \frac{\partial(Q/\mu)}{\partial y}$ ▷ *symbolically as a rational fraction*
8: $C \leftarrow \mathrm{Bound}(c(\boldsymbol{x}_\varepsilon, \boldsymbol{y}_\varepsilon))$
9: $B_1 \leftarrow \mathrm{Bound}(\boldsymbol{x}'v^\circ - \boldsymbol{y}'u^\circ)$
10: $B_2 \leftarrow \mathrm{Bound}(u^{\circ\prime}v^\circ - v^{\circ\prime}u^\circ)$
11: $\delta \leftarrow 2\pi\varepsilon(2B_1 + \varepsilon B_2)C$
▷ *Return interval enclosure*
12: **return** $\boldsymbol{I} \leftarrow \boldsymbol{I}_{\check{s}} + [-\delta, \delta]$

Proposition 5.1. *Let P, Q, μ be as in Problem 1.1. If* AbIntValidQuad$(P, Q, \mu, x^\circ, y^\circ, u^\circ, v^\circ, \check{s}, \varepsilon, N)$ *with $\varepsilon > 0$ computed by* OvalValid *does not fail, then it returns an interval enclosure $\boldsymbol{I}$ of $\mathfrak{I}(h)$.*

Moreover, if N is sufficiently large, then it does never fail and computes an interval $\boldsymbol{I}$ whose diameter tends exponentially fast to 0 w.r.t. N, in $O(N \log(N))$ arithmetic operations.

Proof. First, an interval enclosure $\boldsymbol{I}_{\check{s}}$ of the integral $I_{\check{s}}$ (5.2) along the approximate curve $(\check{x}, \check{y})$ is computed rigorously by using RTA representations and operations for the integrand (lines 3 and 4), and then integrating it over $[0, 2\pi]$ (line 5). Next, lines 6–11 compute a bound δ for the error between $I_{\check{s}}$ and the exact value $\mathfrak{I}(h)$ taken along $(x^\#, y^\#)$ of $\Gamma(h)$, so that AbIntValidQuad returns a validated enclosure $\boldsymbol{I} = \boldsymbol{I}_{\check{s}} + [-\delta, \delta]$ for $\mathfrak{I}(h)$. The strategy to obtain δ is detailed below.

• *Error bound δ with Stokes' theorem.* Let (x, y) denote the standard coordinates of the plane $\mathbb{R}^2$, and add a third one, z, living in $\mathbb{R}/2\pi\mathbb{Z}$, corresponding to the time t. The approximate parameterization $(\check{x}, \check{y})$ traces a smooth simple closed curve $t \in \mathbb{R}/2\pi\mathbb{Z} \mapsto (x^\circ(t) + \check{s}(t)u^\circ(t), y^\circ(t) + \check{s}(t)v^\circ(t), t)$ in $\mathbb{R}^2 \times \mathbb{R}/2\pi\mathbb{Z}$. Similarly, $(x^\#, y^\#)$ traces $t \in \mathbb{R}/2\pi\mathbb{Z} \mapsto (x^\circ(t) + s^\#(t)u^\circ(t), y^\circ(t) + s^\#(t)v^\circ(t), t)$. Since $s^\#$ is analytic, it only has a finite number of zeros in $[0, 2\pi]$. Hence, since (u°, v°) never vanishes, the two curves only intersect each other a finite number of times. The "strip" $\mathcal{S}$ between them,

$$\mathcal{S} = \left\{ \left(x^\circ(t) + su^\circ(t), y^\circ(t) + sv^\circ(t), t\right), \quad t \in \mathbb{R}/2\pi\mathbb{Z},\ s \in [\check{s}(t), s^\#(t)] \text{ or } [s^\#(t), \check{s}(t)] \right\},$$

is a finite collection of two-dimensional cells (with two singularities each). Choosing the appropriate orientation for each cell, the difference between $\mathfrak{I}(h)$ and $I_{\check{s}}$ is equal to the integral of the 1-form $(P/\mu)\mathrm{d}y - (Q/\mu)\mathrm{d}x$ along the oriented boundary $\partial\mathcal{S}$ of $\mathcal{S}$:

$$\mathfrak{I}(h) - I_{\check{s}} = \int_{\partial\mathcal{S}} \frac{P(x,y)\mathrm{d}y - Q(x,y)\,\mathrm{d}x}{\mu(x,y)}.$$

Using Stokes' theorem [Lee 2013, Thm. 14.9], and since P and Q do not depend on z, we obtain:

$$|\mathfrak{I}(h) - I_{\check{s}}| \leqslant \int_{\mathcal{S}} |c(x,y)|\ \mathrm{d}x \wedge \mathrm{d}y,$$

with $c(x, y)$ defined as in Equation (5.3).

A bound C for the integrand $|c(x, y)|$ is computed in lines 6–8. Moreover, the integration domain $\mathcal{S}$ can be safely overapproximated by the geometric realization T_ε of the tube $T(\check{x}, \check{y}, u^\circ, v^\circ, \varepsilon)$ in $\mathbb{R}^2 \times \mathbb{R}/2\pi\mathbb{Z}$, which is the two-dimensional manifold parameterized by:

$$\begin{array}{rccc} \sigma : & \mathbb{R}/2\pi\mathbb{Z} \times [-\varepsilon, \varepsilon] & \longrightarrow & T_\varepsilon \\ & (t, s) & \longmapsto & \begin{pmatrix} \check{x}(t) + s\,u^\circ(t) \\ \check{y}(t) + s\,v^\circ(t) \\ t \end{pmatrix}. \end{array}$$

Hence, the difference $\mathfrak{I}(h) - I_{\check{s}}$ is bounded by:

$$\begin{aligned} |\mathfrak{I}(h) - I_{\check{s}}| &\leqslant C \int_{T_\varepsilon} |\mathrm{d}x \wedge \mathrm{d}y| \\ &= C \int_0^{2\pi} \int_{-\varepsilon}^{\varepsilon} \left| \begin{vmatrix} \check{x}'(t) & u^\circ(t) \\ \check{y}'(t) & v^\circ(t) \end{vmatrix} + s \begin{vmatrix} u^{\circ\prime}(t) & u^\circ(t) \\ v^{\circ\prime}(t) & v^\circ(t) \end{vmatrix} \right| \mathrm{d}s\,\mathrm{d}t \\ &\leqslant 2\pi\varepsilon\,(2B_1 + \varepsilon B_2)\,C = \delta, \end{aligned}$$

$$\text{with} \quad B_1 \geqslant \sup_{0 \leqslant t \leqslant 2\pi} \left| \begin{vmatrix} \check{x}'(t) & u^\circ(t) \\ \check{y}'(t) & v^\circ(t) \end{vmatrix} \right| \quad \text{and} \quad B_2 \geqslant \sup_{0 \leqslant t \leqslant 2\pi} \left| \begin{vmatrix} u^{\circ\prime}(t) & u^\circ(t) \\ v^{\circ\prime}(t) & v^\circ(t) \end{vmatrix} \right|,$$

computed in lines 9 and 10. Hence, $\mathfrak{I}(h) = I_{\check{s}} + (\mathfrak{I}(h) - I_{\check{s}}) \in \boldsymbol{I}_{\check{s}} + [-\delta, \delta] = \boldsymbol{I}$.

• *Convergence.* When N is large enough, $(\check{x}, \check{y})$ is sufficiently close to $\Gamma(h)$, and since μ does not vanish in a neighborhood of this oval, all the functions involved in the computation of $\boldsymbol{I}_{\check{s}}$ (lines 3–5) are analytic; hence the diameter of this interval tends exponentially fast to 0 w.r.t. N.

The same arguments hold for the computation of Stokes' error bound δ in lines 6–11. Note that B_1, B_2 and C are bounded since they converge to the corresponding $\|\cdot\|_{\ell^1}$ norms with $(\check{x}, \check{y})$ and (u°, v°) replaced by $(x^\#, y^\#)$ and $\frac{\nabla H(x^\circ, y^\circ)}{\|\nabla H(x^\circ, y^\circ)\|_2}$, respectively. Since ε tends to 0 exponentially fast w.r.t. N according to Proposition 4.1, so does δ (line 11), and consequently the total diameter of $\boldsymbol{I}$ in line 12.

• *Complexity.* The asymptotic complexity w.r.t. the globally set RTA degree N is determined by the FFT_N/IFFT_N routines used for TP/RTA multiplications and divisions. □

Now we state our main algorithm AbIntValid and our main theorem, which is a straightforward consequence of Propositions 3.1, 3.5, 4.1 and 5.1.

Algorithm 6 AbIntValid$(H, h, \Sigma, P, Q, \mu, x_{ini}, y_{ini}, \varepsilon_0, x_r, y_r, \varsigma, N_0, N)$

Input: Potential H, level h, transversal Σ, P, Q, μ as in Problem 1.1, $(x_{ini}, y_{ini}) \in \mathbb{R}^2$, $\varepsilon_0 > 0$, $(x_r, y_r) \in \mathbb{R}^2$, $\varsigma = \pm 1$, $N_0 \in \mathbb{N}$ and $N \geqslant N_0$
Output: interval enclosure $\boldsymbol{I}$ of $\mathfrak{I}(h)$

1: $(x^\circ, y^\circ) \leftarrow$ OvalApproxInit$(H, \Sigma, x_{ini}, y_{ini}, N_0)$
2: $(u^\circ, v^\circ), \check{s} \leftarrow$ OvalApproxRefine$(H, h, x^\circ, y^\circ, N)$
3: $\varepsilon \leftarrow$ OvalValid$(H, h, x_{ini}, y_{ini}, \varepsilon_0, x_r, y_r, \varsigma, x^\circ, y^\circ, u^\circ, v^\circ, \check{s}, N)$
4: **return** $\boldsymbol{I} \leftarrow$ AbIntValidQuad$(P, Q, \mu, x^\circ, y^\circ, u^\circ, v^\circ, \check{s}, \varepsilon, N)$

Theorem 5.2. *Let H, P, Q, μ be as in Problem 1.1, h be a regular level value on a transversal Σ, the ball of center (x_{ini}, y_{ini}) and radius ε_0 contain a unique point in $H^{-1}(h) \cap \Sigma$, the point (x_r, y_r) be in the interior of the oval $\Gamma(h)$, and $\varsigma \in \{-1, 1\}$ be the winding number of $\Gamma(h)$ w.r.t. (x_r, y_r) in the orientation prescribed by the Hamiltonian. Then for a sufficiently large degree N_0 (depending on the preceding input data but not on the target accuracy), as the approximation degree $N \to \infty$, Algorithm AbIntValid$(H, h, \Sigma, P, Q, \mu, x_{ini}, y_{ini}, \varepsilon_0, x_r, y_r, \varsigma, N_0, N)$ never fails and returns an interval enclosure $\boldsymbol{I}$ of $\mathfrak{I}(h)$ in $O(N \log N)$ arithmetic operations. The diameter of $\boldsymbol{I}$ tends to 0 exponentially fast, i.e., there exists $\kappa > 1$ depending on $H, h, \Sigma, P, Q, \mu, x_{ini}, y_{ini}, \varepsilon_0, x_r, y_r, \varsigma, N_0$ such that $\mathrm{diam}(\boldsymbol{I}) = O(\kappa^{-N})$.*

6 IMPLEMENTATION DETAILS AND CONCURRENT APPROACHES

We now present the Julia implementation of our method and review concurrent approaches. The repositories for our code are available from https://gitlab.inria.fr/abintvalid.

6.1 A brief overview of our Julia implementation

Two main reasons motivating our choice of the Julia language for the prototype implementation accompanying this article are: the convenience of the interactive mode for experimenting, with performances close to compiled languages such as C; and the existing libraries for numerical computing, notably approximations with Fourier series. We developed two Julia packages: `RigorousFourier.jl` to provide RTA data structures and operations used throughout this article, and `AbelianIntegral.jl` to implement Algorithm AbIntValid and its subroutines.

`RigorousFourier.jl (v.0.0.1)`. We built this package on top of the existing `ApproxFun.jl` package[12] which provides a very modular framework to approximate and manipulate functions in

[12] https://juliaapproximation.github.io/ApproxFun.jl/latest/

various bases, including Fourier: A degree N TP is stored as the length $2N+1$ vector of its coefficients represented by floating-point numbers of arbitrary precision (the `BigFloat` type in Julia is a wrapper for the standard GNU MPFR C library[13]). Since this data structure is parameterized by the type of the coefficients, TPs can also be defined with interval coefficients[14]. Therefore, in `RigorousFourier.jl`, RTAs are implemented as pairs made up of a TP with interval coefficients together with a floating-point error bound. Operations on them, described in Section 2, are implemented using interval operations on the interval coefficients, and floating-point operations with the suitable rounding mode activated for the error component. Finally, a global approximation degree N is set by the user so that TPs and RTAs are automatically truncated to degree N after each operation. Note that the degree $N' = 2^{\lceil \log_2(N) \rceil} \geqslant N$ is used internally for FFT based operations (such as multiplication) for the sake of efficiency.

Remark 6.1. *Using floating-point coefficients for TPs $\check{f}$ adds rounding errors to the approximation error w.r.t. to the represented functions f. Similarly, using floating-point interval coefficients for RTAs $\mathbf{f}$ enlarges the set of functions contained in $\mathbf{f}$, due to the growth of the interval widths. However, it can be easily proved that with $p \approx N$ bits for the underlying floating-point arithmetic, the total contribution of rounding errors in the evaluation of a fixed expression involving TPs or RTAs and basic operations (addition, subtraction, coefficientwise multiplication, integration, etc.) of Section 2.2 is bounded by $O(N^c 2^{-N})$ for some $c > 0$. Therefore, these errors will not affect the convergence result of Theorem 5.2.*

The next remark concerns FFT-based operations: interpolation, fast multiplication, division, composition, etc.

Remark 6.2. *Directly using an FFT scheme on intervals rather than floating-point numbers may lead to gross overestimations. Some works addressed the use of FFT for convolutions in a rigorous setting with a focus on several related issues (e.g., aliasing errors [Cyranka 2014; Lessard 2018], overestimation of tail coefficients [Lessard 2018], wrapping effect when input coefficients are given as large intervals [Liu and Kreinovich 2010]). However, the intrinsic growth of the intervals during the execution of the FFT algorithm remains an open and insufficiently documented problem in the literature (see e.g., [Cyranka 2014, §3.4]). We currently propose two approaches to alleviate this issue when performing fast and rigorous RTA multiplication:*

- *(a) An efficient and numerically stable way is to first "*flatten*" RTAs (i.e., interval coefficients are shrunk to their floating-point midpoint, with the error component updated accordingly), then to apply a floating-point FFT scheme, and finally to update the error component with known rigorous estimates on the resulting rounding errors (see e.g., [Brisebarre et al. 2020]). Using $p \approx N$ bits of precision, the claimed bounds are in $O(N^2 2^{-N})$, which, again, does not affect the convergence result of Theorem 5.2.*
  In our current prototype implementation, though, this last step is still missing, because adapting these error estimates to the precise FFT scheme used in `ApproxFun.jl` requires some additional work.
- *(b) A more direct approach is to apply an interval FFT scheme on the flattened coefficients. Although quite simple, this method may generate larger error bounds and leads to an overhead factor of roughly 5 to 8 in timings compared to method (a). We however also provide the timings for this method (b) since it is the only fully rigorous one in our prototype code.*

[13] https://www.mpfr.org/

[14] Intervals are provided by `IntervalArithmetic.jl`, https://juliaintervals.github.io/IntervalArithmetic.jl/latest/intro/.

AbelianIntegral.jl (v.0.0.1). In this package, we implemented all routines related to the evaluation of Abelian integrals, described in Sections 3 to 5. Its main dependencies are:

- the `TypedPolynomials.jl` package[15] for multivariate polynomials used in the data structures representing integrable systems, oval families and perturbations;
- the `DifferentialEquations.jl` package[16] that provides a numerical ODE solver to instantiate the DSolve routine used by Algorithm OvalApproxInit to compute the initial guess (x°, y°). As of today, it uses by default a Tsitouras 5/4 Runge-Kutta method with 4th order interpolant [Tsitouras 2011]. This package also offers event detection features, which we use to end the integration when the first return onto the transversal Σ is detected. The resulting implementation of DSolve is thus compliant with Assumption 3.2.

Remark 6.3. *Since Assumption 3.2 is satisfied in our implementation, and considering the rounding error estimates given in Remarks 6.1 and 6.2 (assuming option (a)), the convergence results of the preceding sections in $O(\kappa^{-N})$ for some $\kappa \in (1,2)$ still hold for the floating-point implementation of AbIntValid. Combining this with the bit complexity of $O(N \log N)$ for arithmetic operations on N-bit floating-point numbers [Harvey and van der Hoeven 2021], a refined version of Theorem 5.2 says that Algorithm AbIntValid, using degree N TPs/RTAs and floating-point arithmetic with N bits of precision (neglecting considerations about the exponent range), computes a rigorous enclosure of $\mathfrak{I}(h)$ with $O(N)$ bits of accuracy in $O(N^2 \log^2 N)$ bit operations.*

6.2 Concurrent approaches and implementations

Despite the constraint of keeping the mathematical background of the validation routine as minimal as possible to ease the future formal proof implementation, our algorithm remains sufficiently efficient for the rigorous high precision evaluation of Abelian integrals. Timings are compared with the following available software:

- The subdivision algorithm [Johnson and Tucker 2011], by one of the authors of this article, is, to our knowledge, the only fully algorithmic work dedicated to the rigorous evaluation of Abelian integrals. The integration domain (the interior of the oval, by using the Green formula) is safely approximated using boxes. Such a strategy would be a natural candidate for a formal proof implementation. However, contrary to the exponential convergence of our method, this routine has only finite order, which makes it unsuitable for high precision evaluation.
- The CAPD library, developed in C++, is well established among the community of computer assisted proofs in dynamical systems, both for its efficiency and reliability. Trajectories are rigorously computed using various higher order methods, including Taylor forms [Neumaier 2003]. Our strategy to evaluate an Abelian integral using CAPD is to augment the plane (x, y) with a third dimension I (the integral), and then add a third component to the vector field, representing $\dot{I}$, which is equal to the integrand of (1.1), where dx and dy are replaced by the values of $\dot{x}$ and $\dot{y}$ given by the first two components. However, implementing such a method in Coq would require significant work to formalize some theory of differential equations (e.g., the Picard-Lindelöf theorem) and a rigorous ODE solver.
- The Arb library[17] [Johansson 2017], developed in C, is a general purpose rigorous numerics library providing extremely efficient basic routines, including one for arbitrary-precision numerical integration with rigorous error bounds [Johansson 2018]. Although it cannot rigorously approximate general algebraic functions yet (in order to approximate the oval),

[15] https://github.com/JuliaAlgebra/TypedPolynomials.jl.
[16] https://diffeq.sciml.ai/v2.0/
[17] https://arblib.org/

we used it, as a SageMath[18] external package, for the first example of Section 7 where explicit parameterizations can be used. In this case, timings easily outperform our method and CAPD.

We also wish to mention [Lairez et al. 2019]. This approach should be quite effective to address Problem 1.1 but we did not include it in our comparisons due to the following two issues: as of today, there is no ready-to-use implementation and, overall, the algorithms rely on tools that are currently out of reach for the proof assistant Coq.

7 WORKED EXAMPLES, DISCUSSION AND CONCLUSION

We now apply our algorithm on two problems regarding the isolation of limit cycles in near-integrable polynomial planar vector fields:

- The first example (Section 7.1) is an integrable quartic system introduced in [Johnson 2011] to claim $\mathcal{H}(4) \geqslant 26$. Unfortunately, an implementation error led to incorrect evaluation of the Abelian integrals. An analysis led by the authors of this article (see [Bréhard 2011, Chap. 6]) showed that 24 limit cycles could still be obtained from this example. The particular form of the potential function H makes it possible to divide the ovals into arcs $x \mapsto y(x)$ and $y \mapsto x(y)$ with simple formulas, so that $\mathfrak{I}(h)$ can easily be computed as a classical integral over a segment of $\mathbb{R}$ with an explicit integrand.
- The second example (Section 7.2) is a cubic Hamiltonian system used in [Li et al. 2009] to demonstrate $\mathcal{Z}(3) \geqslant 13$. Contrary to our first example, here the ovals of the unperturbed system cannot be easily parameterized using explicit formulas. Therefore, the evaluation of the Abelian integrals necessarily requires a method working on the implicit representation of the ovals.

The tests were executed on an Intel i7-6600U 2.60GHz CPU with a 64-bit Linux-based system.

7.1 Johnson's symmetric quartic system revisited

In [Johnson 2011], T. Johnson constructed a perturbed quartic pseudo-Hamiltonian vector field of the form:

$$\begin{cases} \dot{x} = -4y^2(y^2 - Y_0), \\ \dot{y} = 4xy(x^2 - X_0) + \varepsilon g(x, y), \end{cases} \tag{7.1}$$

where $X_0 = \frac{9}{10}$, $Y_0 = \frac{11}{10}$, and g is a degree 4 perturbation,

$$g(x, y) = \alpha_{00} + \alpha_{20}x^2 + \alpha_{22}x^2y^2 + \alpha_{40}x^4 + \alpha_{04}y^4, \tag{7.2}$$

with well-chosen coefficients α_{ij}. Using a rigorous validation integration routine, he claimed to prove the existence of 26 limit cycles, thus surpassing the previously known record $\mathcal{H}(4) \geqslant 22$ [Christopher and Li 2007]. Unfortunately, a bug in his implementation led him to observe more zeros in the Abelian integral than what actually exists. In [Bréhard 2011, Chap. 6], the authors of this article however showed that by using different values for the coefficients α_{ij}, one can prove the following Theorem establishing $\mathcal{H}(4) \geqslant 24$[19].

Theorem 7.1. *The quartic system* (7.1) *with* $X_0 = \frac{9}{10}$, $Y_0 = \frac{11}{10}$, *and the degree* 4 *perturbation* g *defined as in* (7.2) *with coefficients:*

$$\alpha_{00} = -0.78622148667854837664, \quad \alpha_{20} = 0.87723523612653436051, \quad \alpha_{22} = 1,$$
$$\alpha_{40} = 0.23742713894293038223, \quad \alpha_{04} = -0.21823846173078863753,$$

[18] https://www.sagemath.org/

[19] In the meantime, Prohens and Torregrosa [Prohens and Torregrosa 2019] showed $\mathcal{H}(4) \geqslant 28$ by a different kind of computer-aided proof (and for a different system).

has at least 24 limit cycles.

A particularity of the potential function H associated to this system is that the ovals can be parameterized explicitly using arcs $x(y)$ and $y(x)$ (see Section 7.1.1 below). This was the strategy adopted in [Johnson 2011] and [Bréhard 2011, Chap. 6] for the rigorous evaluation of the Abelian integral. In Section 7.1.2, we compare our approach to the other methods described above. Finally, in Section 7.1.3, we redo the computations using the algorithm presented in this article (which does not require explicit parameterizations), which gives a rigorous proof of Theorem 7.1.

7.1.1 Potential function, ovals and symmetries. The quartic system (7.1) admits the following potential function:

$$H(x,y) = (x^2 - X_0)^2 + (y^2 - Y_0)^2, \tag{7.3}$$

together with the rescaling factor $\mu(x,y) = y$.

The level set associated to the parameter $h \geqslant 0$, represented in the (x^2, y^2) plane, is the portion of the circle of center (X_0, Y_0) and radius $\sqrt{h}$ located in the positive quadrant (see Figure 1a). In the (x,y) plane, this corresponds to the ovals depicted in Figure 1b. Only those not intersecting the x-axis (over which μ vanishes) must be considered. We call them *small* and *big* ovals, respectively:

- When $h \in (0, X_0^2)$, the circle in the (x^2, y^2)-plane entirely lies in the positive quadrant. In the (x,y) plane, this results into four symmetric *small ovals*:

$$\Gamma^{++}(h) = H^{-1}(h) \cap \mathbb{R}_{\geqslant 0} \times \mathbb{R}_{\geqslant 0}, \qquad \Gamma^{+-}(h) = H^{-1}(h) \cap \mathbb{R}_{\geqslant 0} \times \mathbb{R}_{\leqslant 0},$$
$$\Gamma^{-+}(h) = H^{-1}(h) \cap \mathbb{R}_{\leqslant 0} \times \mathbb{R}_{\geqslant 0}, \qquad \Gamma^{--}(h) = H^{-1}(h) \cap \mathbb{R}_{\leqslant 0} \times \mathbb{R}_{\leqslant 0}.$$

- When $h \in (X_0^2, Y_0^2)$, a portion of this circle in the (x^2, y^2)-plane crosses the y-axis, yielding two symmetric *big ovals*:

$$\Gamma^{+}(h) = H^{-1}(h) \cap \mathbb{R} \times \mathbb{R}_{\geqslant 0}, \qquad \Gamma^{-}(h) = H^{-1}(h) \cap \mathbb{R} \times \mathbb{R}_{\leqslant 0}.$$

In what follows, the notation $\Gamma(h)$ stands for the small oval $\Gamma^{++}(h)$ when $h \in (0, X_0^2)$, and for the big oval $\Gamma^{+}(h)$ when $h \in (X_0^2, Y_0^2)$. The corresponding Abelian integral is:

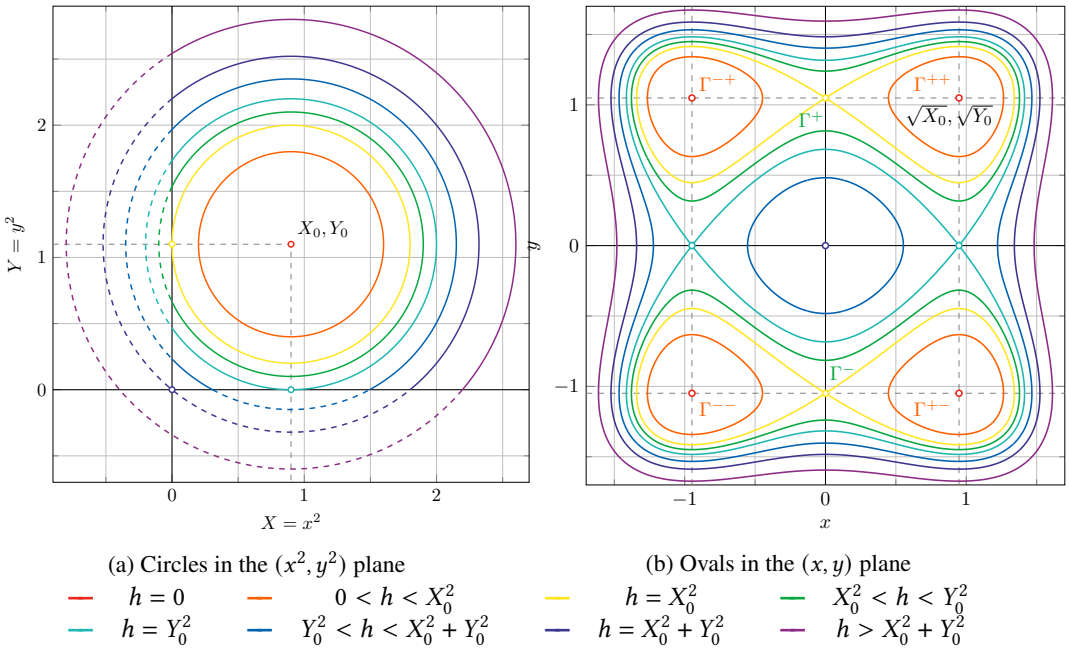
$$\mathfrak{I}(h) = -\int_{\Gamma(h)} \frac{g(x,y)}{y}\,\mathrm{d}x.$$

By symmetry and the fact that $g(x,y)$ contains only even powers of x and y, it is clear that the Abelian integrals taken along the other ovals of $H^{-1}(h)$ are equal, up to the sign.

Due to the very specific form of the potential function H in (7.3), we can divide the ovals into four arcs $x \mapsto y_{\mathrm{up}}(x)$, $x \mapsto y_{\mathrm{down}}(x)$, $y \mapsto x_{\mathrm{left}}(y)$ and $y \mapsto x_{\mathrm{right}}(y)$, parameterized by explicit algebraic functions involving square roots only (see [Bréhard 2011, §6.2.3]). Therefore, explicit rigorous quadrature routines, e.g., the one provided by Arb, can be used to address this example.

7.1.2 Timings. In Table 1, we present the execution times of our implementation and its three alternatives. The timings obtained with our method are comparable to those of the reference CAPD implementation, which is very encouraging for a future formal proof implementation in Coq.

Concerning the implementations of [Johnson 2011] and in Arb/Sage, they both rely on an explicit parameterization of the ovals, which is possible only due to the very specific form of the potential function. While Arb/Sage exhibits remarkable timings due to the particularly efficient implementation of this library, the code of [Johnson 2011] is limited by the binary64 (double precision) format and the finite order quadrature used to compute $\mathfrak{I}(h)$.



(a) Circles in the (x^2, y^2) plane (b) Ovals in the (x, y) plane

Fig. 1. Level curves of the potential function H

	our method			other methods		
p	N	(a)	(b)	CAPD	[Johnson 2011]	Arb/Sage
4	55	0.32	1.6	0.44	7.9	0.012
6	70	0.59	3.3	0.54	19.0	0.013
8	100	0.60	4.8	0.67	49.4	0.014
16	150	1.5	9.0	2.0	—	0.018
24	225	1.5	9.1	2.2	—	0.021
32	275	3.8	18.7	3.6	—	0.031
48	425	3.8	18.9	6.4	—	0.040
64	570	8.0	42.1	11.1	—	0.043
96	870	9.4	46.1	23.8	—	0.078
128	1165	18.8	96.2	41.7	—	0.084

Table 1. Timings in seconds to rigorously evaluate $\mathfrak{I}(0.25)$ associated to (7.1) with relative error at most 10^{-p}, using our method with degree N RTAs and floating-point FFT (a) or interval FFT (b) for RTA multiplication (see Remark 6.2), and other software: the CAPD library, the original code of [Johnson 2011] and the Arb library.

7.1.3 Computer-assisted proof of $\mathcal{H}(4) \geqslant 24$. In Table 2, we compute rigorous interval enclosures $\boldsymbol{I}_N$ for $\mathfrak{I}(h)$, for specific values of h, using our algorithm with RTAs of degree N as small as possible, as long as the resulting interval guarantees the sign of $\mathfrak{I}(h)$. The existence of sufficiently many simple zeros needed to prove Theorem 7.1 results from the obtained sign alternation. Figures 2 and 3 provide a graphical representation of the sign alternations on small and big ovals.

ovals	r	h	N	$\boldsymbol{I}_N$	$\boldsymbol{I}_{512}$	sign($\mathfrak{I}(h)$)
small	0.5	0.25	17	[3.5963e-5, 9.6953e-5]	[6.6457e-5, 6.6458e-5]	+
	0.78	0.6084	37	[-1.3730e-4, -1.6580e-5]	[-7.6939e-5, -7.6938e-5]	−
	0.88	0.7744	93	[1.6407e-9, 3.1821e-8]	[1.6730e-8, 1.6731e-8]	+
	0.89	0.7921	100	[-3.8327e-8, -1.1066e-9]	[-1.9717e-8, -1.9716e-8]	−
	0.895	0.801025	102	[9.5936e-10, 1.1267e-07]	[5.6812e-8, 5.6813e-8]	+
	0.8987	0.80766169	111	[-5.1962e-7, -5.2103e-8]	[-2.8586e-7, -2.8585e-7]	−
big	0.901	0.811801	197	[-1.2620e-5, -3.4042e-7]	[-6.4798e-6, -6.4797e-6]	−
	0.93	0.8649	128	[2.0334e-5, 6.2143e-4]	[3.2088e-4, 3.2089e-4]	+
	0.95	0.9025	140	[-1.9667e-4, -7.0461e-6]	[-1.0186e-4, -1.0185e-4]	−

Table 2. Rigorous evaluation of $\mathfrak{I}(h)$ with our algorithm and resulting sign alternations on small and big ovals. Computations are carried out with 256 bits of floating-point precision and the smallest degree N for RTAs such that the rigorous enclosure $\boldsymbol{I}_N$ guarantees the sign of $\mathfrak{I}(h)$. Tighter enclosures $\boldsymbol{I}_{512}$ using a high degree 512 are also provided.

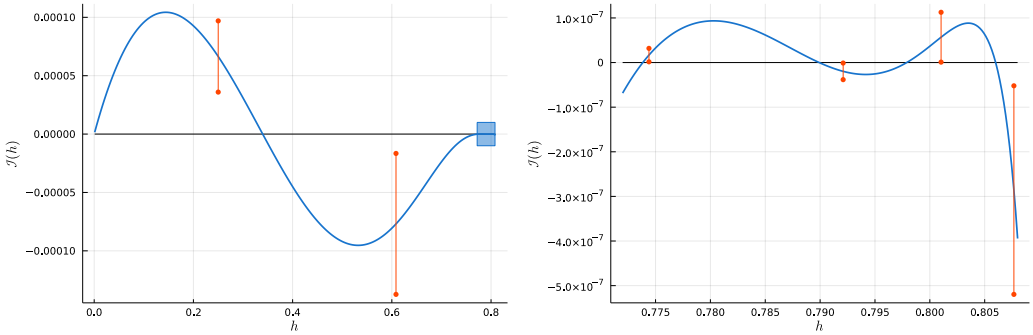


Fig. 2. Rigorously computed interval enclosures of $\mathfrak{I}(h)$ on small ovals, for $h \in \{0.25, 0.6084, 0.7744, 0.7921, 0.801025, 0.80766169\}$, proving the existence of 5 zeros. The right plot is a zoom of the blue box in the left plot.

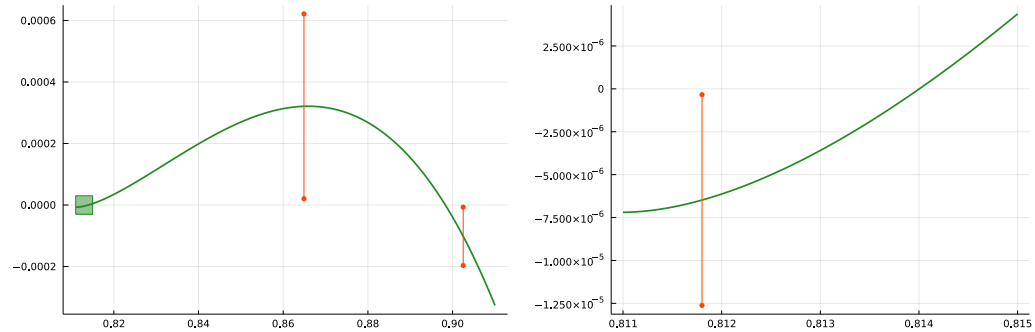


Fig. 3. Rigorously computed interval enclosures of $\mathfrak{I}(h)$ on big ovals, for $h \in \{0.811801, 0.8649, 0.9025\}$, proving the existence of 2 zeros. The right plot is a zoom of the green box in the left plot.

Proof of Theorem 7.1. Let h_i ($1 \leqslant i \leqslant 9$) denote the 9 values of h in Table 2, taken in increasing order. The rigorous intervals enclosures given in this table are sufficient to guarantee the sign of $\mathfrak{I}(h_i)$. According to Theorem 1.3, there exists for each h_i, an $\varepsilon_i > 0$ such that $d(h_i, \varepsilon)$ and $\mathfrak{I}(h_i)$ share the same (strict) sign whenever $0 < \varepsilon \leqslant \varepsilon_i$. Hence, with $\varepsilon^* = \min_{0 \leqslant i \leqslant 9} \varepsilon_i > 0$, we have that the displacement function $h \mapsto d(h, \varepsilon)$ alternates sign at least 5 times on $(0, X_0^2)$ and at least 2 times on (X_0^2, Y_0^2), for each fixed $0 < \varepsilon \leqslant \varepsilon^*$, giving respectively at least 5 and 2 isolated zeros in these intervals. Using the symmetries on the four small ovals and the two big ovals, we deduce the existence of at least $5 \times 4 + 2 \times 2 = 24$ limit cycles in the quartic system (7.1) whenever $0 < \varepsilon \leqslant \varepsilon^*$. □

7.2 Li, Liu and Yang's cubic system with 13 limit cycles

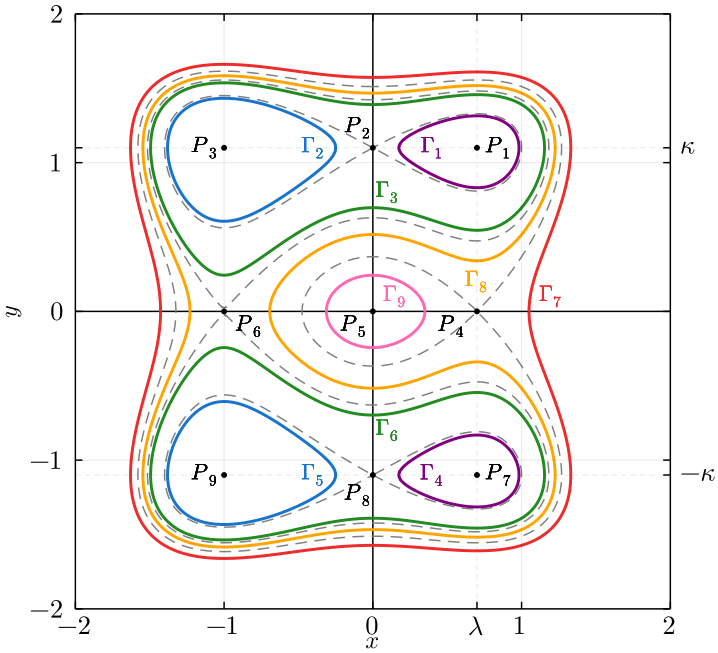


Fig. 4. Ovals of potential function H (7.5), with $\lambda = 0.7$ and $\kappa = 1.1$.

In [Li et al. 2009], C. Li, C. Liu and J. Yang showed that $\mathcal{Z}(3) \geqslant 13$ using the following perturbed cubic Hamiltonian system:

$$\begin{cases} \dot{x} = -y^3 + \kappa^2 y, \\ \dot{y} = x^3 + (1-\lambda)x^2 - \lambda x + \varepsilon y(\alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 y^2), \end{cases} \tag{7.4}$$

with well-chosen[20] $0 < \lambda < 1$, $\kappa > 1$, and coefficients $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}$ for the ε-small perturbation. The critical points of the corresponding first integral,

$$H(x,y) = \underbrace{\frac{x^4}{4} + \frac{1-\lambda}{3}x^3 - \frac{\lambda}{2}x^2}_{F(x)} + \frac{y^4}{4} - \frac{\kappa^2 y^2}{2}, \tag{7.5}$$

depicted in Figure 4, are ranked according to their level value:

$$\begin{array}{ccccccccccc} H(P_3) & < & H(P_1) & < & H(P_2) & < & H(P_6) & < & H(P_4) & < & H(P_5). \\ \| & & \| & & \| & & & & & & \| \\ H(P_9) & & H(P_7) & & H(P_8) & & & & & & 0 \end{array}$$

The resulting families of ovals can be considered as a (horizontally) dissymmeterized version of the ovals of [Johnson 2011] discussed in the previous section: Γ_2 (and symmetrically Γ_5) when $h \in (H(P_3), H(P_2))$, Γ_1 (and sym. Γ_4) when $h \in (H(P_1), H(P_2))$, Γ_3 (and sym. Γ_6) when $h \in (H(P_2), H(P_6))$, Γ_8 when $h \in (H(P_6), H(P_4))$, Γ_9 when $h \in (H(P_4), H(P_5))$, and Γ_7 when $h > H(P_4)$.

The Abelian integrals under consideration are:

$$\mathfrak{I}_j(h) = \int_{\Gamma_j(h)} y(\alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 y^2)\,\mathrm{d}x, \qquad \text{for } 1 \leqslant j \leqslant 9.$$

In Section 7.2.1, we first benchmark our algorithm on the more complex geometry of these ovals and compare the results with competing software. After that, we show in Section 7.2.2 how our approach based on numerical experiments and a posteriori validation provides an easy-to-check instance of $\mathcal{Z}(3) \geqslant 13$ with effective values for the coefficients $\lambda, \kappa, \alpha_1, \alpha_2, \alpha_3, \alpha_4$ in System (7.4).

Theorem 7.2. *Consider System (7.4) with parameters:*

$$\lambda = 0.17, \qquad \alpha_1 = \ \ 8.808855593098, \qquad \alpha_3 = -2.599597169555 \cdot 10^{-5},$$
$$\kappa = 20, \qquad \alpha_2 = -2.078279433211 \cdot 10^{-5}, \qquad \alpha_4 = -7.340712733831 \cdot 10^{-3}.$$

Then, over their respective interval of definition:

- $\mathfrak{I}_2$ *(and symmetrically* $\mathfrak{I}_5$*) has at least one simple zero;*
- $\mathfrak{I}_3$ *(and symmetrically* $\mathfrak{I}_6$*) has at least 5 simple zeros;*
- $\mathfrak{I}_7$ *has at least one simple zero.*

As a result, the cubic Hamiltonian system (7.4) has at least 13 limit cycles.

Such an approach complements the detailed but harder-to-check existence proof for such coefficients in [Li et al. 2009], based on the careful analysis of limit cycles for Liénard systems by F. Dumortier and C. Li [Dumortier and Li 2003].

7.2.1 Timings. We benchmark our implementation on ovals Γ_3 (non symmetric, non convex) and Γ_8 (since it is not even star-shaped). We set $\lambda = \frac{7}{10}$ and $\kappa = \frac{11}{10}$ (the same values were used to plot Figure 4), and $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1$. Let Q_1 (resp. Q_2) denote the topmost intersection of the homoclinic connection of P_6 (resp. P_4) with the y-axis. Then, for Γ_3, we consider the periodic orbit starting from $\frac{1}{10}P_2 + \frac{9}{10}Q_1$, and for Γ_8, we take as initial point $\frac{1}{2}Q_1 + \frac{1}{2}Q_2$ (this corresponds exactly to the ovals Γ_3 and Γ_8 depicted in Figure 4).

The execution times in terms of the target relative accuracy for the Abelian integral are given in Table 3. They show that our method is particularly well-suited for high-precision evaluation of Abelian integrals, no matter the geometry of the oval. Note however that CAPD still performs better

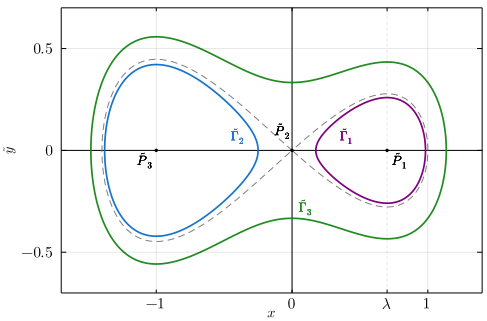
[20]Note that it is assumed in [Li et al. 2009] that $\kappa > 10$.

for low precision (up to standard double precision), probably due to its time interval subdivision strategy that we have not implemented yet in our method.

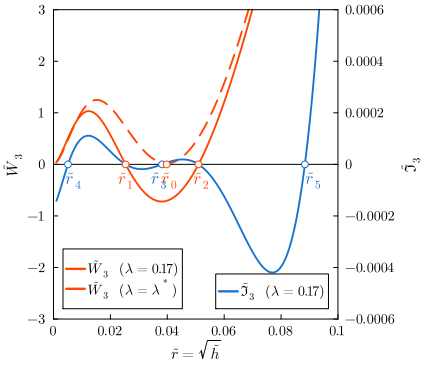
Finally, we also tried this example with the algorithm of [Johnson and Tucker 2011]. The resulting timings were not included in Table 3 since they already exceed 5 minutes for a targeted 4 digit accuracy. The reason is that the underlying oval subdivision strategy converges too slowly to provide more than a few digits in a reasonable time.

oval Γ_3					oval Γ_8				
p	N	(a)	(b)	CAPD	p	N	(a)	(b)	CAPD
4	100	0.45	2.2	0.25	4	220	0.90	4.6	0.51
6	125	0.46	2.2	0.38	6	280	1.7	11.6	0.83
8	170	1.0	5.8	0.54	8	385	1.8	11.7	1.2
16	400	2.1	11.6	1.7	16	860	4.4	23.7	3.8
24	625	4.3	23.5	4.1	24	1330	8.9	47.5	9.8
32	855	5.0	25.5	8.3	32	1805	10.8	52.7	25.3
48	1310	11.0	52.0	22.1	48	2750	22.3	122.6	64.5
64	1765	11.1	55.4	46.2	64	3695	22.8	136.2	139.5
96	2675	27.4	136.3	158.5	96	5585	59.7	295.8	485.3
128	3580	30.0	137.1	361.6	128	7475	60.7	298.8	1084.0

Table 3. Timings in seconds to rigorously evaluate an Abelian integral along ovals from the Γ_3 and Γ_8 family with relative error at most 10^{-p}, using our implementation with degree N RTAs and floating-point FFT (a) or interval FFT (b) for RTA multiplication (see Remark 6.2), and CAPD.



(a) Ovals of potential function $\widetilde{H}$ (7.6).



(b) Wronskian $\widetilde{W}_3$ of system $\{\widetilde{\mathfrak{I}}_{3i}\}_{1\leqslant i\leqslant 4}$ (7.7) and a linear combination $\widetilde{\mathfrak{I}}_3$ with $\lambda = 0.17$ and appropriate β_i realizing 5 zeros $\{\tilde{r}_i\}_{1\leqslant i\leqslant 5}$.

Fig. 5. Obtaining numerical β_i realizing 5 zeros for $\widetilde{\mathfrak{I}}_3$ in Dumortier and Li's system [Dumortier and Li 2003].

7.2.2 Computer-assisted proof of $\mathcal{Z}(3) \geqslant 13$. Before proving Theorem 7.2, we present a heuristic process, directly inspired by [Li et al. 2009], that yields potential relevant parameters. To do so, we focus on $\mathfrak{I}_2$ and $\mathfrak{I}_3$ (once the parameters fixed, the result on $\mathfrak{I}_7$ will be straightforward to establish

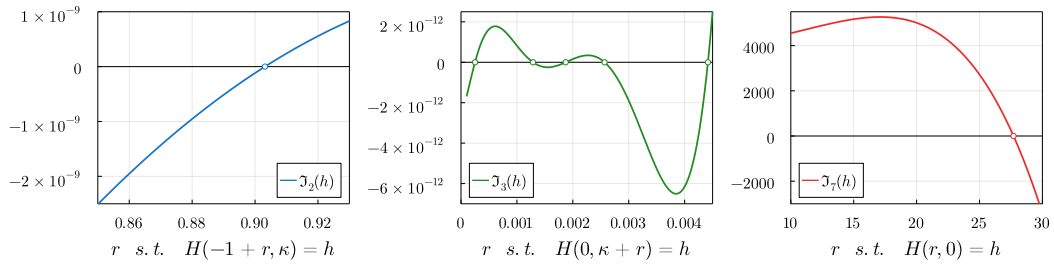


Fig. 6. Abelian integrals $\mathfrak{I}_2$, $\mathfrak{I}_3$, $\mathfrak{I}_7$ in Li, Liu and Yang's system (7.4) with the parameters of Theorem 7.2, having 1, 5 and 1 zeros, respectively.

oval	r	N	$\mathfrak{I}(h)$	δ	sign
Γ_2	0.88	2000	[-9.54e-10, -9.53e-10]	5.45e-22	−
	0.92	2000	[5.55e-10, 5.56e-10]	3.92e-21	+
Γ_3	0.0002	8000	[-5.21e-13, -5.20e-13]	4.16e-22	−
	0.001	8000	[8.45e-13, 8.46e-13]	3.24e-26	+
	0.0015	4000	[-2.16e-13, -2.15e-13]	5.11e-21	−
	0.0022	4000	[3.52e-13, 3.53e-13]	2.27e-22	+
	0.004	4000	[-6.13e-12, -6.12e-12]	6.76e-25	−
	0.0045	2000	[2.54e-12, 2.55e-12]	1.86e-20	+
Γ_7	25	100	[2718.12, 2718.13]	3.63e-6	+
	30	100	[-3419.83, -3419.82]	6.00e-7	−

Table 4. Rigorous evaluations of $\mathfrak{I}_2(h)$, $\mathfrak{I}_3(h)$ and $\mathfrak{I}_7(h)$ with the parameters of Theorem 7.2 using AbIntValid. The ovals $\Gamma_2(h)$, $\Gamma_3(h)$ and $\Gamma_7(h)$ are parameterized with a variable r denoting a position on a transversal, with $h = H(-1+r,\kappa)$, $h = H(0,\kappa+r)$, and $h = H(r,0)$, respectively. N is the degree used for RTAs, and δ is the diameter of the computed interval enclosure for $\mathfrak{I}(h)$.

with our approach). In this case, [Li et al. 2009] uses the change of variable $\tilde{y} = \frac{y^2-\kappa^2}{2}$ to obtain a simpler potential function,

$$\widetilde{H}(x,\tilde{y}) = \tilde{y}^2 + F(x), \tag{7.6}$$

whose families of ovals $\widetilde{\Gamma}_1$, $\widetilde{\Gamma}_2$ and $\widetilde{\Gamma}_3$ are depicted in Figure 5a. The corresponding Abelian integrals $\widetilde{\mathfrak{I}}_j$ for $1 \leqslant j \leqslant 3$ are:

$$\widetilde{\mathfrak{I}}_j(h) \;=\; \int_{\widetilde{\Gamma}_j(h)} \tilde{y}(\beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 \tilde{y}^2)\,\mathrm{d}x \;=\; \beta_1 \widetilde{\mathfrak{I}}_{j1}(h) + \beta_2 \widetilde{\mathfrak{I}}_{j2}(h) + \beta_3 \widetilde{\mathfrak{I}}_{j3}(h) + \beta_4 \widetilde{\mathfrak{I}}_{j4}(h). \tag{7.7}$$

The authors of [Li et al. 2009] establish, as soon as κ is large enough, an equivalence between the numbers of zeros of $\mathfrak{I}_2$ and $\widetilde{\mathfrak{I}}_2$, resp. $\mathfrak{I}_3$ and $\widetilde{\mathfrak{I}}_3$.

Zeros of $\widetilde{\mathfrak{I}}_2$ and $\widetilde{\mathfrak{I}}_3$. By combining the main existence result of [Dumortier and Li 2003] with some asymptotic analysis, C. Li, C. Liu and J. Yang prove [Li et al. 2009, Lemma 1] that if λ is close to a certain critical value λ^*, then there exist coefficients β_1, β_2, β_3, β_4 such that $\widetilde{\mathfrak{I}}_2$ has at least one simple zero and $\widetilde{\mathfrak{I}}_3$ has at least 5 simple zeros. The reason why 5 zeros can be obtained from the 4-term linear combination $\widetilde{\mathfrak{I}}_3$ is that for this critical value λ^*, the Wronskian $\widetilde{W}_3(h) = \det\left(\widetilde{\mathfrak{I}}_{3j}^{(i-1)}(h)\right)_{1\leqslant i,j\leqslant 4}$ has a

double zero $\tilde{h}_0 = \tilde{r}_0^2$ (see Figure 5b). Then, an appropriate perturbation of λ around λ^* creates two simple zeros for $\widetilde{W}(h)$, from which one hopes to deduce the existence of values for the β_i realizing 5 zeros[21].

In contrast with the non-constructive existence proof in [Li et al. 2009], we confirm this heuristic in a constructive way, by determining candidate values for λ and the β_i purely numerically. To do so, we compute Chebyshev approximations of the Wronskian $r \mapsto \widetilde{W}_3(r^2)$[22] for several values of λ and count the number of zeros. This is done by interpolating the Abelian integrals on a large number of Chebyshev nodes, with very high precision for each value[23]. The value $\lambda = 0.17$ is found to fulfill those requirements, that is 2 zeros $\tilde{h}_1 = \tilde{r}_1^2 < \tilde{r}_2^2 = \tilde{h}_2$ for $\widetilde{W}_3$ (at least numerically, see Figure 5b), with $\tilde{r}_1$ and $\tilde{r}_2$ not too close for better numerical stability.

Now, to obtain values for the β_i, we fix $\beta_1 = 1$, and numerically solve for β_2, β_3 and β_4 the equations $\widetilde{\mathfrak{I}}_3(\tilde{r}_i^2) = 0$ for $\tilde{r}_1$, $\tilde{r}_2$ and the midpoint $\tilde{r}_3 = \frac{\tilde{r}_1+\tilde{r}_2}{2}$. As desired, the numerical plot of the resulting combination $r \mapsto \widetilde{\mathfrak{I}}_3(r)$ in Figure 5b exhibits 2 additional zeros $\tilde{r}_4$ and $\tilde{r}_5$, making a total of 5 zeros for $\widetilde{\mathfrak{I}}_3$. Finally, a numerical plot suggests that $\widetilde{\mathfrak{I}}_2$ also has one simple zero.

Zeros of $\mathfrak{I}_2$, $\mathfrak{I}_3$ and $\mathfrak{I}_7$. As claimed by [Li et al. 2009, Lemma 2], by selecting a sufficiently large parameter $\kappa > 1$ in (7.4) and defining the coefficients α_i from the β_i using the following formulas:

$$\alpha_1 = \frac{\beta_1}{\kappa^5} + \frac{3\beta_4}{2\kappa}, \qquad \alpha_2 = \frac{\beta_2}{\kappa^5}, \qquad \alpha_3 = \frac{\beta_3}{\kappa^5}, \qquad \alpha_4 = -\frac{\beta_4}{2\kappa^3},$$

the 5 zeros of $\widetilde{\mathfrak{I}}_3$ and the single zero of $\widetilde{\mathfrak{I}}_2$ are recovered in $\mathfrak{I}_3$ and $\mathfrak{I}_2$. Moreover, with these coefficients, an additional zero can be found for $\mathfrak{I}_7$, according to [Li et al. 2009, Lemma 3].

Fixing $\kappa = 20$ (based on some experimenting) leads to the desired number of zeros (see plots in Figure 6), while being not too large, to avoid numerical issues. For the final values of the α_i, given in Theorem 7.2 and used for the computer-assisted proof, no less than 13 digits are necessary in the decimal truncation to keep the expected number of zeros.

We now conclude with a rigorous numerics based proof of the desired sign alternations of the Abelian integrals, hence of $\mathcal{Z}(3) \geqslant 13$.

Proof of Theorem 7.2. We perform rigorous evaluations of Abelian integrals $\mathfrak{I}_2$, $\mathfrak{I}_3$ and $\mathfrak{I}_7$ on the points given in Table 4 using Algorithm AbIntValid, with the values of λ, κ and the α_i given in the statement of the theorem. By continuity of these three functions, it follows that $\mathfrak{I}_2$, $\mathfrak{I}_3$ and $\mathfrak{I}_7$ have at least 1, 5 and 1 simple zeros, respectively. Moreover, by the symmetry $(x, y) \mapsto (x, -y)$, $\mathfrak{I}_5$ and $\mathfrak{I}_6$ have at least 1 and 5 simple zeros, respectively. Finally, by the same argument used in the proof of Theorem 7.1 and based on Theorem 1.3 (Poincaré-Pontryagin), we obtain that for sufficiently small $\varepsilon > 0$, System (7.4) has at least 13 limit cycles. □

From a performance point of view, it is noteworthy that rather large degrees for RTAs are necessary for $\mathfrak{I}_2$ and $\mathfrak{I}_3$ (see Table 4), mainly due to the winding number check in OvalValid since the denominator in line 22 becomes small when the curve gets close to the point (x_r, y_r). Alternative methods for this should be investigated in the future to avoid this phenomenon.

[21] Indeed, the case of a single (simple) zero for the Wronskian (together with additional non-vanishing conditions on lower-dimensional Wronskians of the system) would imply that $\widetilde{\mathfrak{I}}_3$ cannot exhibit more than 4 zeros (see [Novaes and Torregrosa 2017, Cor. 1.4]).

[22] For convenience, we use $r = \sqrt{h}$ instead of h as the independent variable.

[23] For this we use a plain floating-point evaluation scheme, adapted from AbIntValid by removing all the rigorous numerics for more efficiency, since error bounds are not necessary at this stage.

7.3 Conclusion

We have presented a set of algorithms that makes it possible to evaluate Abelian integrals in a fast and rigorous way. Next steps are the development of a rigorous FFT following [Brisebarre et al. 2020], cf. (a) in Remark 6.2, and the formalization in Coq of our method. Our hope is that, in particular, it will be used in the dynamical system community as a basic brick in the quest for new records for the $\mathcal{H}(n)$ and $\mathcal{Z}(n)$ bounds.

ACKNOWLEDGMENTS

We wish to warmly thank J.-L. Figueras, G. Hanrot, S. Olver, R. M. Slevinsky and D. Wilczak for helpful discussions and advise. The first author wishes to thank Uppsala University and the CAPA group where part of this work was done. The fourth author wishes to thank Collegium de Lyon and the INRIA international chair programme for hosting and funding him during the time spent in Lyon while working on this paper.

REFERENCES

S. S. Abhyankar and C. L. Bajaj. 1988. Computations with Algebraic Curves. In *Symbolic and Algebraic Computation, International Symposium ISSAC'88, Rome, Italy, July 4-8, 1988, Proceedings (Lecture Notes in Computer Science, Vol. 358)*, Patrizia M. Gianni (Ed.). Springer, 274–284. https://doi.org/10.1007/3-540-51084-2_26

A. Albouy and V. Kaloshin. 2012. Finiteness of central configurations of five bodies in the plane. *Ann. of Math. (2)* 176, 1 (2012), 535–588. https://doi.org/10.4007/annals.2012.176.1.10

V. I. Arnold. 1977. Loss of stability of self-induced oscillations near resonance, and versal deformations of equivariant vector fields. *Funkcional. Anal. i Priložen.* 11, 2 (1977), 1–10, 95. Translated in English in Functional Anal. Appl. **11** (1977), no. 2, 85–92.

V. I. Arnold. 1990. *Ten problems*. Adv. Soviet Math., Vol. 1. Amer. Math. Soc., Providence, RI. 1–8 pages.

C. L. Bajaj and G. L. Xu. 1997. Piecewise rational approximations of real algebraic curves. *J. Comput. Math.* 15, 1 (1997), 55–71.

C. Beltrán and A. Leykin. 2013. Robust certified numerical homotopy tracking. *Found. Comput. Math.* 13, 2 (2013), 253–295. https://doi.org/10.1007/s10208-013-9143-2

Y. Bertot and P. Castéran. 2013. *Interactive Theorem Proving and Program Development: Coq'Art: The Calculus of Inductive Constructions*. Springer Science & Business Media.

J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. 2017. Julia: a fresh approach to numerical computing. *SIAM Rev.* 59, 1 (2017), 65–98. https://doi.org/10.1137/141000671

G. Binyamini, D. Novikov, and S. Yakovenko. 2010. On the number of zeros of Abelian integrals. *Invent. Math.* 181, 2 (2010), 227–289. https://doi.org/10.1007/s00222-010-0244-0

J.-B. Bost and J.-F. Mestre. 1988. Moyenne arithmético-géométrique et périodes des courbes de genre 1 et 2. *Gaz. Math.* 38 (1988), 36–64.

N. Brisebarre and M. Joldeş. 2010. Chebyshev interpolation polynomial-based tools for rigorous computing. In *ISSAC 2010—Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*. ACM, New York, 147–154. https://doi.org/10.1145/1837934.1837966

N. Brisebarre, M. Joldeş, J.-M. Muller, A.-M. Naneş, and J. Picot. 2020. Error Analysis of Some Operations Involved in the Cooley-Tukey Fast Fourier Transform. *ACM Trans. Math. Softw.* 46, 2, Article 11 (may 2020), 27 pages. https://doi.org/10.1145/3368619

F. Bréhard. 2011. *Certified numerics in function spaces: Polynomial approximations meet computer algebra and formal proof*. Ph. D. Dissertation. École normale supérieure de Lyon – Université de Lyon, Lyon, France. https://tel.archives-ouvertes.fr/tel-02337901

H. Cartan. 1995. *Elementary theory of analytic functions of one or several complex variables*. Dover Publications, Inc., New York. 228 pages. Translated from the French, Reprint of the 1973 edition.

F. Chen, C. Li, J. Llibre, and Z. Zhang. 2006. A unified proof on the weak Hilbert 16th problem for $n = 2$. *Journal of Differential Equations* 221, 2 (2006), 309–342. https://doi.org/10.1016/j.jde.2005.01.009

C. Christopher and C. Li. 2007. *Limit cycles of differential equations*. Birkhäuser Verlag, Basel. viii+171 pages.

J. Cyranka. 2014. Efficient and generic algorithm for rigorous integration forward in time of dPDEs: Part I. *J. Sci. Comput.* 59, 1 (2014), 28–52. https://doi.org/10.1007/s10915-013-9749-1

F. Dumortier and C. Li. 2003. Perturbation from an elliptic Hamiltonian of degree four. IV. Figure eight-loop. *J. Differential Equations* 188, 2 (2003), 512–554. https://doi.org/10.1016/S0022-0396(02)00111-0

H. Ehlich and K. Zeller. 1966. Auswertung der Normen von Interpolationsoperatoren. *Math. Ann.* 164 (1966), 105–112. https://doi.org/10.1007/BF01429047

C. Epstein, W. L. Miranker, and T. J. Rivlin. 1982a. Ultra-arithmetic. I. Function data types. *Math. Comput. Simulation* 24, 1 (1982), 1–18. https://doi.org/10.1016/0378-4754(82)90045-3

C. Epstein, W. L. Miranker, and T. J. Rivlin. 1982b. Ultra-arithmetic. II. Intervals of polynomials. *Math. Comput. Simulation* 24, 1 (1982), 19–29. https://doi.org/10.1016/0378-4754(82)90046-5

J.-L. Figueras and R. de la Llave. 2017. Numerical computations and computer assisted proofs of periodic orbits of the Kuramoto-Sivashinsky equation. *SIAM J. Appl. Dyn. Syst.* 16, 2 (2017), 834–852.

X.-S. Gao and M. Li. 2004. Rational quadratic approximation to real algebraic curves. *Comput. Aided Geom. Design* 21, 8 (2004), 805–828. https://doi.org/10.1016/j.cagd.2004.07.009

D. Harvey and J. van der Hoeven. 2021. Integer multiplication in time $O(n\log n)$. *Annals of Mathematics* 193, 2 (2021), 563 – 617. https://doi.org/10.4007/annals.2021.193.2.4

D. Hilbert. 1900. Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Congress zu Paris 1900. *Nachr. Ges. Wiss. Göttingen, Math.-Phys. Kl.* 1900 (1900), 253–297.

H. Hong and J. Schicho. 1998. Algorithms for trigonometric curves (simplification, implicitization, parameterization). *J. Symbolic Comput.* 26, 3 (1998), 279–300. https://doi.org/10.1006/jsco.1998.0212

A. Hungria, J.-P. Lessard, and J. D. Mireles James. 2016. Rigorous numerics for analytic solutions of differential equations: the radii polynomial approach. *Math. Comp.* 85, 299 (2016), 1427–1459.

F. Johansson. 2017. Arb: efficient arbitrary-precision midpoint-radius interval arithmetic. *IEEE Trans. Comput.* 66, 8 (2017), 1281–1292. https://doi.org/10.1109/TC.2017.2690633

F. Johansson. 2018. Numerical integration in arbitrary-precision ball arithmetic. In *Mathematical Software – ICMS 2018*. Springer Lecture Notes in Computer Science, 255–263. https://doi.org/10.1007/978-3-319-96418-8_30

T. Johnson. 2011. A quartic system with twenty-six limit cycles. *Exp. Math.* 20, 3 (2011), 323–328.

T. Johnson and W. Tucker. 2010. An improved lower bound on the number of limit cycles bifurcating from a quintic Hamiltonian planar vector field under quintic perturbation. *Internat. J. Bifur. Chaos Appl. Sci. Engrg.* 20, 1 (2010), 63–70. https://doi.org/10.1142/S0218127410025405

T. Johnson and W. Tucker. 2011. On a computer-aided approach to the computation of Abelian integrals. *BIT* 51, 3 (2011), 653–667. https://doi.org/10.1007/s10543-011-0318-4

T. Kapela, M. Mrozek, D. Wilczak, and P. Zgliczyński. 2020. CAPD::DynSys: a flexible C++ toolbox for rigorous numerical analysis of dynamical systems. *Commun. Nonlinear Sci. Numer. Simul.* (2020), 105578.

T. Kapela, D. Wilczak, and P. Zgliczyński. 2021. Recent advances in rigorous computation of Poincaré maps. arXiv:2104.08046 [math.NA]

P. Lairez, M. Mezzarobba, and M. Safey El Din. 2019. Computing the Volume of Compact Semi-Algebraic Sets. In *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation*. ACM, New York, NY, USA, 259–266. https://doi.org/10.1145/3326229.3326262

O. E. Lanford, III. 1982. A computer-assisted proof of the Feigenbaum conjectures. *Bull. Amer. Math. Soc. (N.S.)* 6, 3 (1982), 427–434. https://doi.org/10.1090/S0273-0979-1982-15008-X

J. M. Lee. 2013. *Introduction to smooth manifolds* (second ed.). Graduate Texts in Mathematics, Vol. 218. Springer, New York. xvi+708 pages.

J.-P. Lessard. 2018. Computing discrete convolutions with verified accuracy via Banach algebras and the FFT. *Appl. Math.* 63, 3 (2018), 219–235. https://doi.org/10.21136/AM.2018.0082-18

C. Li, C. Liu, and J. Yang. 2009. A cubic system with thirteen limit cycles. *J. Differential Equations* 246, 9 (2009), 3609–3619. https://doi.org/10.1016/j.jde.2009.01.038

G. Liu and V. Kreinovich. 2010. Fast convolution and fast Fourier transform under interval and fuzzy uncertainty. *J. Comput. System Sci.* 76, 1 (2010), 63–76. https://doi.org/10.1016/j.jcss.2009.05.006

K. Makino and M. Berz. 2003a. Taylor models and other validated functional inclusion methods. *Int. J. Pure Appl. Math.* 4, 4 (2003), 379–456.

K. Makino and M. Berz. 2003b. Taylor models and other validated functional inclusion methods. *Int. J. Pure Appl. Math.* 6, 3 (2003), 239–316.

B. Martin, A. Goldsztejn, L. Granvilliers, and C. Jermann. 2013. Certified parallelotope continuation for one-manifolds. *SIAM J. Numer. Anal.* 51, 6 (2013), 3373–3401. https://doi.org/10.1137/130906544

R. E. Moore. 1966. *Interval analysis*. Vol. 4. Prentice-Hall Englewood Cliffs.

A. Neumaier. 2003. Taylor forms—use and limits. *Reliab. Comput.* 9, 1 (2003), 43–79.

D. D. Novaes and J. Torregrosa. 2017. On extended Chebyshev systems with positive accuracy. *J. Math. Anal. Appl.* 448, 1 (2017), 171–186. https://doi.org/10.1016/j.jmaa.2016.10.076

M. J. D. Powell. 1981. *Approximation theory and methods*. Cambridge University Press.

R. Prohens and J. Torregrosa. 2019. New lower bounds for the Hilbert numbers using reversible centers. *Nonlinearity* 32, 1 (2019), 331–355. https://doi.org/10.1088/1361-6544/aae94d
C. Tsitouras. 2011. Runge-Kutta pairs of order 5(4) satisfying only the first column simplifying assumption. *Comput. Math. Appl.* 62, 2 (2011), 770–775. https://doi.org/10.1016/j.camwa.2011.06.002
W. Tucker. 2011. *Validated numerics: a short introduction to rigorous computations*. Princeton University Press.
D. Wilczak. 2003. Chaos in the Kuramoto-Sivashinsky equations—a computer-assisted proof. *J. Differential Equations* 194, 2 (2003), 433–459. https://doi.org/10.1016/S0022-0396(03)00104-9
G. B. Wright, M. Javed, H. Montanelli, and L. N. Trefethen. 2015. Extension of Chebfun to periodic functions. *SIAM J. Sci. Comput.* 37, 5 (2015), C554–C573. https://doi.org/10.1137/141001007
J. Xu, M. Burr, and C. Yap. 2018. An Approach for Certifying Homotopy Continuation Paths: Univariate Case. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation* (New York, NY, USA) *(ISSAC '18)*. Association for Computing Machinery, New York, NY, USA, 399–406.
N. Yamamoto. 1998. A numerical verification method for solutions of boundary value problems with local uniqueness by Banach's fixed-point theorem. *SIAM J. Numer. Anal.* 35, 5 (1998), 2004–2013.
H. Yang, B. Jüttler, and L. Gonzalez-Vega. 2010. An evolution-based approach for approximate parameterization of implicitly defined curves by polynomial parametric spline curves. *Math. Comput. Sci.* 4, 4 (2010), 463–479.
A. Zygmund. 2002. *Trigonometric series. Vol. I, II* (third ed.). Cambridge University Press, Cambridge. xii; Vol. I: xiv+383 pp.; Vol. II: viii+364 pages.

A ALGORITHM ISPOSITIVE: CHECKING POSITIVITY OF RTAS

In rigorous numerics, it is common to determine if a function f is positive over a given interval using interval subdivision and branch and bound techniques [Tucker 2011, Chap. 5.1]. In our case, however, evaluating RTAs of high degree over non-thin intervals yields extremely large overapproximations, making this process ineffective for our purpose.

The following routine IsPositive is based on an alternative approach. The RTA $\boldsymbol{f} = (\check{f}, \varepsilon)$ is preconditioned by an approximate inverse $\check{g} \approx \frac{1}{\check{f}}$. If $\boldsymbol{f}$ does not vanish, then the product $\check{g}\boldsymbol{f}$ is likely to be close to 1, which is measured by the quantity $\|1 - \check{g}\boldsymbol{f}\|_{\ell^1}$.

Algorithm 7 IsPositive$(\boldsymbol{f}, N)$

Input: RTA $\boldsymbol{f} = (\check{f}, \varepsilon)$ and approximation degree N
Output: Boolean **true** or **false**, with **true** guaranteeing that $f > 0$ for all $f \in \boldsymbol{f}$

$a_0 \leftarrow$ constant coefficient of $\check{f}$
if $a_0 \leqslant 0$ **then**
    return false
else if Bound$(\boldsymbol{f} - a_0) < a_0$ **then**
    return true
else
    $\check{g} \leftarrow \mathrm{FFT}_N((\frac{1}{f_j})_{j=0}^{2N})$, where $(f_j)_{j=0}^{2N} \leftarrow \mathrm{IFFT}_N(\check{f})$
    if $\check{g}$ is well defined and Bound$(1 - \check{g}\boldsymbol{f}) < 1$ **then**
        return true
    else
        return false
    end if
end if

Lemma A.1. *Let $\boldsymbol{f} = (\check{f}, \varepsilon)$ be an RTA. Then,*

(*i*) *For any approximation degree N, if IsPositive$(\boldsymbol{f}, N)$ returns **true**, then $f > 0$ for all $f \in C^0_{2\pi}$ represented by $\boldsymbol{f}$.*

(*ii*) *If moreover* $\check{f} > 0$ *and* $\varepsilon < \|\check{f}^{-1}\|_{\ell^1}^{-1}$ *(which implies* $f > 0$ *for all* $f \in \boldsymbol{f}$*), then there is an* N_0 *such that* IsPositive$(\boldsymbol{f}, N)$ *returns* ***true*** *for all* $N \geqslant N_0$.

(*iii*) IsPositive$(\boldsymbol{f}, N)$ *runs in* $O(N' \log(N'))$ *arithmetic operations, where* $N' = \max(\deg \boldsymbol{f}, N)$.

Proof. (*i*) *Soundness.* Assume IsPositive$(\boldsymbol{f}, N)$ returns **true**. A first possibility is that the condition in line 4 is true, implying that $a_0 > 0$ and that any function $f \in \boldsymbol{f}$ can be written as $a_0 + h$ with $\|h\|_\infty < a_0$, and is therefore positive. The other possibility (line 8) is that we obtained a $\check{g}$ such that for any $f \in \boldsymbol{f}$, $\|1 - \check{g}f\|_\infty < 1$, implying that $f(t) \neq 0$ for all t. By continuity, f has constant sign over $\mathbb{R}$. In particular, this is true for $\check{f}$. Since the condition in line 2 was false, i.e., $a_0 = \int_0^{2\pi} \check{f}\,\mathrm{d}t > 0$, we get $\check{f} > 0$. Finally, since the ball in $C^0_{2\pi}$ denoted by $\boldsymbol{f}$ is convex, the same statement $f > 0$ is true for all $f \in \boldsymbol{f}$.

(*ii*) *Completeness.* Suppose conversely that $\check{f} > 0$ and that $\varepsilon < \|\check{f}^{-1}\|_{\ell^1}^{-1}$. In particular, $\varepsilon < \|\check{f}^{-1}\|_\infty^{-1} = \min_{t\in\mathbb{R}} \check{f}(t)$, implying that $f > 0$ for all $f \in \boldsymbol{f} = (\check{f}, \varepsilon)$. We prove that IsPositive$(\boldsymbol{f}, N)$ returns **true** for N sufficiently large.

First, the condition in line 2 is false since $a_0 = \int_0^{2\pi} \check{f}\,\mathrm{d}t > 0$. Then, the algorithm either returns **true** if the condition in line 4 is true, or it computes $\check{g} \leftarrow \mathcal{I}_N(\check{f}^{-1})$ in line 7. Since $1/\check{f}$ is analytic, $\|\check{g} - \check{f}^{-1}\|_{\ell^1}$ converges to 0 as the approximation degree N tends to infinity. Hence,

$$\textsc{Bound}(1 - \check{g}\boldsymbol{f}) = \|1 - \check{g}\check{f}\|_{\ell^1} + \varepsilon\textsc{Bound}(\check{g}) \longrightarrow 0 + \varepsilon\|\check{f}^{-1}\|_{\ell^1} < 1 \quad \text{as } N \longrightarrow \infty.$$

Therefore, there is an N_0 such that for all $N \geqslant N_0$, the degree N TP $\check{g}$ computed by trigonometric interpolation makes the condition in line 8 true, so that IsPositive$(\boldsymbol{f}, N)$ returns **true**.

(*iii*) *Complexity.* The asymptotic complexity is determined by the FFT_N/IFFT_N routines used for evaluation on the equispaced grid, interpolation and TP/RTA multiplication. □

B PROOF OF LEMMA 3.4 FOR THE PROJECTED PARAMETERIZATION

Proof. The proof consists of three steps. First, we prove the existence of an analytic function $\hat{s} : U \longrightarrow \mathbb{R}$ over a neighborhood $U \subseteq \mathbb{R}^4$ of

$$\hat{\Gamma}(h) = \left\{ (x, y, u, v) \in \mathbb{R}^4 \text{ s.t. } (x, y) \in \Gamma(h) \text{ and } (u, v) = \frac{\nabla H(x, y)}{\|\nabla H(x, y)\|_2} \right\},$$

that serves as a shift to project any point (x, y) onto $\Gamma(h)$ in the direction given by (u, v). Then, we prove that over a possibly smaller neighborhood $U' \subseteq U$ of $\hat{\Gamma}(h)$, the Newton iterations $\mathcal{N}^k_{x,y,u,v}(0)$ (see Eq. (3.5)) converge quadratically fast to $\hat{s}(x, y, u, v)$ for all $(x, y, u, v) \in U'$. Finally, we compose by $t \in [0, 2\pi] \longrightarrow (x^\circ(t), y^\circ(t))$ to complete the proof.

• *Analytic projection.* Consider the equation:

$$F(x, y, u, v, s) = H\,(x + su, y + sv) - h = 0.$$

By definition, $F(x, y, u, v, 0) = 0$ for all $(x, y, u, v) \in \hat{\Gamma}(h)$. Moreover, the derivative with respect to s,

$$\frac{\partial F}{\partial s}(x, y, u, v, s) = u\frac{\partial H}{\partial x}\,(x + su, y + sv) + v\frac{\partial H}{\partial y}\,(x + su, y + sv)$$

satisfies $\frac{\partial F}{\partial s}(x, y, u, v, 0) = \|\nabla H(x, y)\|_2 > 0$ for all $(x, y, u, v) \in \hat{\Gamma}(h)$, since h is a regular value of H. By the analytic implicit function theorem [Cartan 1995, Chap. IV, Prop. 6.1], there exists a neighborhood $U \subseteq \mathbb{R}^4$ of $\hat{\Gamma}(h)$, a neighborhood $V \subseteq \mathbb{R}^5$ of $\hat{\Gamma}(h) \times \{0\}$, and a unique analytic function $\hat{s} : U \longrightarrow \mathbb{R}$ such that:

$$((x, y, u, v, s) \in V \ \wedge\ F(x, y, u, v, s) = 0) \ \Leftrightarrow\ ((x, y, u, v) \in U \ \wedge\ s = \hat{s}(x, y, u, v))\,.$$

In particular, $\hat{s}(x, y, u, v) = 0$ if and only if $(x, y) \in \Gamma(h)$.

• *Convergence of Newton iterations.* Since $\hat{\Gamma}(h)$ is compact, we may assume (up to restricting the neighborhoods U and V) that $\frac{\partial F}{\partial s}(x,y,u,v,s) \geqslant m_1$ for some $m_1 > 0$, and $|\frac{\partial^2 F}{\partial s^2}(x,y,u,v,s)| \leqslant M_2$ for some $M_2 > 0$, for all $(x,y,u,v,s) \in V$. Now for any $(x,y,u,v,s) \in V$, the classical error analysis of Newton's method [Tucker 2011, §5.1.2] yields in our case:

$$|\mathcal{N}_{x,y,u,v}(s) - \hat{s}(x,y,u,v)| \leqslant \frac{M_2}{2m_1}|s - \hat{s}(x,y,u,v)|^2.$$

Therefore, by restricting the neighborhood U of $\hat{\Gamma}(h) \subseteq \hat{s}^{-1}(0)$ to:

$$U' = \left\{(x,y,u,v) \in U, |\hat{s}(x,y,u,v)| \leqslant \frac{m_1}{M_2}\right\},$$

we deduce that the Newton iterations $\mathcal{N}^k_{x,y,u,v}(0)$ converge quadratically fast to the desired limit $\hat{s}(x,y,u,v)$:

$$|\mathcal{N}^k_{x,y,u,v}(0) - \hat{s}(x,y,u,v)| \leqslant \left(\frac{M_2}{2m_1}|\hat{s}(x,y,u,v)|\right)^{2^k-1}|\hat{s}(x,y,u,v)| \leqslant \frac{|\hat{s}(x,y,u,v)|}{2^{2^k-1}}. \tag{B.1}$$

• *Composition with the initial guess.* Let us fix $\eta > 0$ such that any (x°, y°) and (u°, v°) η-close to (x^*, y^*) and $\frac{\nabla H(x^*,y^*)}{\|\nabla H(x^*,y^*)\|_2}$ in $(C^0_{2\pi})^2$ satisfy $(x^\circ(t), y^\circ(t), u^\circ(t), v^\circ(t)) \in U'$ for all $t \in [0, 2\pi]$. By the property of composition of analytic functions, the map $t \mapsto s^\#(t) = \hat{s}(x^\circ(t), y^\circ(t), u^\circ(t), v^\circ(t))$ is analytic, and so is $(x^\#, y^\#)$ by Eq. (3.4). Also, by choosing $\eta > 0$ sufficiently small, we can make $(x^\#, y^\#)$ sufficiently close to (x^*, y^*), which ensures that the winding number with respect to a given point inside $\Gamma(h)$ remains the same (± 1 depending on the orientation). This concludes the proof that $(x^\#, y^\#)$ is an analytic parameterization of $\Gamma(h)$ with same orientation as (x^*, y^*), although not equal to (x^*, y^*) in general. Finally, the quadratic convergence of the Newton iteration scheme for any point $(x^\circ(t), y^\circ(t))$ w.r.t. direction $(u^\circ(t), v^\circ(t))$ follows from the preceding analysis. □

C ALGORITHM NEWTONBALL

In order to make effective the a posteriori validation process presented in Section 2.3, we must be able to automatically determine a radius r and a Lipschitz constant λ satisfying the hypotheses of Theorem 2.5. To this aim, given a function $r \mapsto \lambda(r)$ providing a guaranteed upper bound for the Lipschitz constant for the fixed-point operator $\mathcal{T}$ over $\overline{B}(\check{\varphi}, r)$, and an upper bound for the defect $d \geqslant \|\mathcal{T}(\check{\varphi}) - \check{\varphi}\|$, Algorithm NewtonBall implements a simple bisection method to identify the smallest zero of $r \mapsto f(r) := (1 - \lambda(r))r - d$. Specifically, it maintains a stack of subintervals $[r^-, r^+]$ to be investigated, each of them satisying $f(r^-) \leqslant 0$. Under the reasonable assumption that $\lambda(r)$ is an increasing function of r, the range of f over $[r^-, r^+]$ can be overapproximated by $[(1 - \lambda^+)r^- - d, (1 - \lambda^-)r^+ - d]$ where $\lambda^\pm := \lambda(r^\pm)$, thus giving a simple test (line 7) to detect the possible existence of a zero of f in $[r^-, r^+]$. If moreover $f(r^+) \geqslant 0$, then $[r^-, r^+]$ necessarily contains a zero, and the stack of remaining intervals is cleared (line 8). When the process successfully terminates, it returns a pair (r^+, λ^+) satisfying the hypotheses of Theorem 2.5.

Remark C.1. *Algorithm NewtonBall ensures a slightly stronger property than Theorem 2.5, namely the* strict *inequality $d + \lambda^+ r^+ < r^+$. By doing so, we make sure that this inequality remains valid under small perturbations of $\lambda(r)$, which turns out to be essential in the proof of Proposition 4.1 to guarantee that the unique fixed point is not only continuous, but also analytic.*

Algorithm 8 NewtonBall(d, λ)

Input: defect $d \geqslant \|\mathcal{T}(\breve{\varphi}) - \breve{\varphi}\|$ and procedure λ s.t. $\mathcal{T}$ is $\lambda(r)$-Lipschitz over $\overline{B}(\breve{\varphi}, r)$
Output: (r^+, λ^+) s.t. $\mathcal{T}$ is λ^+-contracting over $\overline{B}(\breve{\varphi}, r^+)$ and $d + \lambda^+ r^+ < r^+$

compute $r_{\max}$ s.t. $\lambda(r_{\max}) \approx 0.5$
push $((0, \lambda(0)), (r_{\max}, \lambda(r_{\max})))$ on a new empty stack S
▷ *Bisection loop*
while S not empty **do**
    $((r^-, \lambda^-), (r^+, \lambda^+)) \leftarrow \mathrm{pop}(S)$
    ▷ *When interval subdivision is below user-defined* δ_λ
    if $\lambda^+ - \lambda^- \leqslant \delta_\lambda$ **and** $d + \lambda^+ r^+ < r^+$ **then**
        return (r^+, λ^+)
    ▷ *Check whether interval* $[r^-, r^+]$ *may contain a zero*
    else if $\lambda^+ - \lambda^- > \delta_\lambda$ **and** $d + \lambda^- r^+ < r^+$ **then**
        if $d + \lambda^+ r^+ \leqslant r^+$ **then** clear(S) **end if** ▷ $[r^-, r^+]$ *necessarily contains a zero*
        ▷ *Subdivide* $[r^-, r^+]$ *and push resulting intervals on the stack*
        $r_m \leftarrow (r^- + r^+)/2$
        $\lambda_m \leftarrow \lambda(r_m)$
        $\mathrm{push}(((r_m, \lambda_m), (r^+, \lambda^+)), S)$
        $\mathrm{push}(((r^-, \lambda^-), (r_m, \lambda_m)), S)$
    end if
end while
▷ *Empty stack*
return *FAIL*