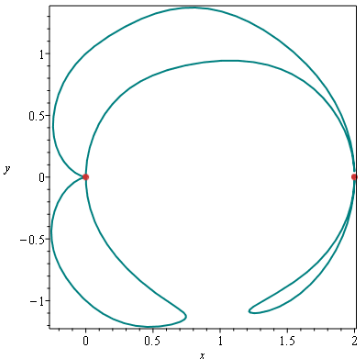
Validated Numerical Software For Algebraic Curves With Singularities



François Boulier, Adrien Poteaux, Florent Bréhard — CFHP, CRIStAL, Univ. Lille

Validated numerics is the art of designing efficient numerical algorithms, yet *reliable* ones, i.e. with guaranteed error bounds encompassing all sources of errors: uncertain data, rounding errors, discretization, etc. The goal is to provide scientists in a broad sense with a "certified pocket calculator". This includes engineers working on safety-critical applications, but also a novel generation of mathematicians using computers to prove their theorems.

The goal of this Ph.D. is to design and implement validated algorithms to compute with *algebraic curves*, which arise in many branches of science. More specifically, we are interested by the difficult case of *singularities*, which may cause catastrophic numerical errors if not dealt properly with. These achievements will allow us to treat currently unreachable applications in computer algebra, physics and robotics.

I. Scientific context: Numerical computation with singular algebraic curves

Motivation. Real and complex algebraic curves are a fundamental object with surprisingly many applications in computer science, mathematics and physics, ranging from automatic geometric theorem proving (see e.g. [16] and references therein) or pure complex algebraic geometry (the crucial Abel-Jacobi map of a Riemann surface [5]) to computer-aided geometric design [15], nonlinear waves models in physics (e.g., nonlinear Schrödinger, Korteweg-de Vries and Kadomtsev-Petviashvili equations [2, 23]), computer-aided design and robotics (connectivity queries for motion planning [9, 14]). Although they are simply defined *implicitly* by polynomial relations in their coordinates:

$$P_1(x_1,\dots,x_n) = \dots = P_r(x_1,\dots,x_n) = 0, \qquad P_j \in \mathbb{K}[x_1,\dots,x_n] \quad \text{with } \mathbb{K} = \mathbb{Q}, \mathbb{R}, \mathbb{C}\dots,$$

manipulating them efficiently and *explicitly* (parametrization, intersection, topology, etc.) requires sophisticated algorithms that have been continuously developed over last decades in computational algebraic geometry [10, 1], either purely symbolically or with the use of numerics [27, 18].

Singularities are the points where the curve is not similar to a line, like a pinch or a crossing of two branches (see the two red dots in the figure above). They occur in many situations like the plane projection of a space curve, or when a robot passes through a singular position (drop in the number of degrees of freedom). Particular care is needed at singularities since algorithms designed for regular curves may exhibit critical behavior at those points, like division by zero or numerical instability. This is a challenge for applications where maximum confidence is required, such as safety-critical engineering or computer-aided proofs in mathematics: a surgical robot is *not* safe up to erratic numerical behavior, *nor* is a geometry theorem true up to rounding errors.

Validated numerics [22, 30] aims at computing with numerical set-valued representations (real intervals, complex balls, set of functions described by a polynomial approximation and an error bound, etc.), thus exploiting the efficiency of floating-point arithmetic while guaranteeing actual mathematical statements: the solution is contained in the computed set. Such techniques have been successfully employed for critical applications and computer-assisted proofs, particularly in the case of differential equations [29, 31, 7].

The goal of this Ph.D. thesis is to treat singularities of algebraic curves using symbolic-numeric methods and validated numerics, towards both efficiently and maximum reliability from algorithm design to implementation.

II. Objectives of the thesis: Validated numerical algorithms for singular algebraic curves, implementations and applications

The Ph.D. thesis consists of three main objectives detailed below, in chronological order. First, one needs to design a symbolic-numeric algorithm to separate and parameterize the branches of an algebraic curve at singularities **(O1)**, combining of the efficiency of floating-point arithmetic and rigorous error bounds obtained from computer algebra and a posteriori validation. After that, a neat implementation **(O2)** of the resulting algorithms will be realized in Julia using validated numerics and dedicated libraries, to finally address some of the challenging applications **(O3)** mentioned in the introduction.

(O1) Symbolic-numeric Newton-Puiseux algorithm using a posteriori validation. The Newton-Puiseux algorithm computes parametrizations of the branches of a curve implicitly defined by $P(X,Y)=0$ at a singularity x_0 under the form of a Puiseux series (i.e., power series with fractional exponents), with *algebraic* coefficients a_k:

$$Y(X)=\sum_{k\geqslant k_0} a_k(X-x_0)^{\frac{k}{e}}, \qquad e\in\mathbb{N}^*, \quad k_0\in\mathbb{Z}, \quad a_k\in\mathbb{C} \quad (k\geqslant k_0).$$

Despite significant improvements made on Newton-Puiseux over the last years (see [25, 26] and references therein), the intrinsic representation size of the algebraic numbers a_k makes this symbolic algorithm not competitive, even for problems $P(X,Y)=0$ of moderate degree (10 – 100). Consequently, in presence of singularities, many algorithms avoid the use of Puiseux series and prefer *turning around* such points when possible. Doing so, however, they ignore the crucial geometrical information encoded by singularities, and they increase the risk of numerical instability when working close to a singular point without exploiting it.

Yet, in many situations, computing accurate numerical approximations of the a_k together with rigorous and tight error bounds, rather than exact representations, is sufficient. Therefore, we propose to design a validated symbolic-numeric Newton-Puiseux algorithm, following these steps:

1. Compute the coefficients a_k numerically *following the structure* of the Puiseux series (i.e., its exponents), obtained using [24]. Note that, without this structure information, such a numerical method applied close to a singularity would be highly unstable. This notably requires designing a numerical Hensel lifting modified with the theory of valuations [26].

2. Finally, design a validation method to compute rigorous and tight error bounds on the a_k's approximate values. This will involve techniques known as *fixed-point a posteriori validation* [6, §3.3], where error bounds are obtained afterwards from the application of a suitable fixed-point theorem. A specific difficulty here to tackle is that singular equations are typical examples of ill-posed problems: they become regular but highly ill-conditioned under infinitesimal perturbations.

Analyzing the bit complexity of the resulting algorithm will be a key step to assess its efficiency compared to purely symbolic Newton-Puiseux and to confirm hopes to tackle currently untractable applications.

(O2) Providing reliable, efficient and open-source implementations. Besides theoretical results, another important part of the Ph.D. is a neat implementation of this validated symbolic-numeric Newton-Puiseux algorithm. Our first requirement is an efficient implementation in Julia relying on the Arb library for validated numerics [21]. Depending on the specific skills of the Ph.D. student, further implementation-related directions can be explored:

- Parallelism in the execution tree of Newton-Puiseux can be exploited to design high performance computing (HPC) implementation of the algorithm. This would involve the expertise of Pierre Fortin (Professor in our team CFHP) at the intersection between HPC, computer arithmetic and computer algebra.

- Guaranteeing the properties of an algorithm on the paper is good, but guaranteeing its *implementation* is even better. Formal proof and theorem provers [17] make possible to design *certified (i.e., formally verified) implementations* at the very level of logic. Provided that the Ph.D. student is familiar with the Coq theorem prover [4], an implementation of this algorithm into the library ApproxModels [8] developed by one of us would be a valuable achievement.

Our objective is treating examples of much higher degrees, say 1,000 – 10,000, than the purely symbolic Newton-Puiseux algorithm.

(O3) Applications to computational algebraic geometry. The resolution of singularities of algebraic curves thanks to the symbolic-numeric Newton-Puiseux algorithm will be the cornerstone of improvements for algorithms in real and complex algebraic geometry. This will enable us to tackle computationally currently challenging applications in the following domains:

- *Connectivity queries* are essential for motion planning in robotics [9, 14]. They can be handled by computing a *roadmap* of the algebraic variety, thus reducing the problem to connectivity queries on real algebraic curves. Several recent algorithms [20, 19] rely on projecting the curve onto a plane and analyzing the resulting 2D singular curves [13]. Efficient Puiseux series over the reals will make it possible to analyze these singularities and lift the branches in the original space. We hope to improve the complexity of computing real curve topology, which is one of the bottlenecks in the roadmap methods for robotics.

- *Homotopy methods* make it possible to compute numerical roots of polynomial systems by deformation (the homotopy) from simpler systems [3]. The curves tracking the roots may cross each other, resulting in singularities that we will be able to treat rigorously with the validated symbolic-numeric Newton-Puiseux algorithm.

- The *Abel-Jacobi map* [5, §1] links crucial information of a complex algebraic curve (a Riemann surface) to computational data, namely contour integrals along paths connecting singularities. Computing them rigorously is a major step towards proofs of existence of particular solutions to nonlinear wave equations in physics [2, 12, 11, 23]: KdV (Korteweg-de Vries), KP (Kadomtsev-Petviashvili) and NLS (nonlinear Schrödinger). This also has applications in computer algebra, e.g. integrating algebraic functions [28].

III. Ph.D. Candidate and Supervisors

The three supervisors of this Ph.D. are members of the CFHP team (Computer Algebra and HPC) in the CRIStAL research unit at Université de Lille:

- The director, François Boulier (Professeur des Universités, HDR), is an expert in algorithmic methods for commutative and differential algebra.

- The co-director, Adrien Poteaux (Maître de Conférences), is an expert in algorithmic methods for algebraic curves and Puiseux series.

- The co-advisor, Florent Bréhard (Chargé de Recherche), is an expert in validated numerics and formally verified numerical algorithms.

The candidate must have preferably a mixed background in computer science (scientific/numerical programming, algorithmics) and mathematics (algebra, complex analysis), and a taste for *computational* mathematics towards applications. Previous experiences in computer algebra, HPC and/or formal proof will also be considered.

IV. References

[1] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*. Springer Berlin, Heidelberg, 2003.

[2] E. D. Belokolos, A. I. Bobenko, V. Z. Enolskii, A. R. Its, and V. B. Matveev. *Algebro-geometric approach to nonlinear integrable equations*, volume 1994. Springer, 1994.

[3] C. Beltrán and A. Leykin. Robust certified numerical homotopy tracking. *Found. Comput. Math.*, 13:253–295, 2013.

[4] Y. Bertot and P. Castéran. *Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions*. Springer Science & Business Media, 2013.

[5] A. I. Bobenko and C. Klein. *Computational Approach to Riemann Surfaces*, volume 2013. Springer Science & Business Media, 2011.

[6] F. Bréhard. *Certified numerics in function spaces : polynomial approximations meet computer algebra and formal proof*. PhD thesis, University of Lyon, France, 2019.

[7] F. Bréhard, N. Brisebarre, M. Joldes, and W. Tucker. Efficient and validated numerical evaluation of abelian integrals. *ACM Trans. Math. Softw.*, pages 1 – 38, 2023. accepted for publication. `https://hal.science/hal-03561096`.

[8] F. Bréhard, A. Mahboubi, and D. Pous. A certificate-based approach to formally verified approximations. In *10th International Conference on Interactive Theorem Proving (ITP 2019)*, volume 141 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:19. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019.

[9] J. Canny. *The complexity of robot motion planning*. PhD thesis, MIT, 1987.

[10] D. Cox, J. Little, D. O'Shea, and M. Sweedler. *Ideals, varieties, and algorithms*. Springer, 1994.

[11] B. Deconinck and M. S. Patterson. Computing the Abel map. *Physica D: Nonlinear Phenomena*, 237(24):3214–3232, 2008.

[12] B. Deconinck and H. Segur. The KP equation with quasiperiodic initial data. *Physica D: Nonlinear Phenomena*, 123(1):123–152, 1998.

[13] D. N. Diatta, S. Diatta, F. Rouillier, M.-F. Roy, and M. Sagraloff. Bounds for polynomials on algebraic numbers and application to curve topology. *Discrete & Computational Geometry*, 67(3):631–697, 2022.

[14] M. S. E. Din and E. Schost. A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. *Journal of the ACM*, 63(6), 2017.

[15] L. Gonzalez-Vega, I. Necula, S. Perez-Diaz, J. Sendra, and J. R. Sendra. Algebraic methods in computer aided geometric design: Theoretical and practical applications. In *Geometric computation*, pages 1–33. World Scientific, 2004.

[16] A. Guergueb, J. Mainguené, and M.-F. Roy. Examples of automatic theorem proving a real geometry. In *Proceedings of the international symposium on Symbolic and algebraic computation*, pages 20–24, 1994.

[17] T. C. Hales. Formal proof. *Notices of the AMS*, 55(11):1370–1380, 2008.

[18] J. D. Hauenstein and A. J. Sommese. What is numerical algebraic geometry? *Journal of Symbolic Computation*, 79:499–507, 2017.

[19] M. N. Islam, A. Poteaux, and R. Prébet. Algorithm for connectivity queries on real algebraic curves. In *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, page 345–353. ACM, 2023.

[20] K. Jin and J. Cheng. On the complexity of computing the topology of real algebraic space curves. *Journal of Systems Science and Complexity*, 34:809–826, 2021.

[21] F. Johansson. Arb: Efficient arbitrary-precision midpoint-radius interval arithmetic. *IEEE Transactions on Computers*, 66(8):1281–1292, 2017.

[22] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, 2009.

[23] M. S. Patterson. *Algebro-geometric algorithms for integrable systems*. PhD thesis, University of Washington, 2007.

[24] A. Poteaux and M. Rybowicz. Good reduction of Puiseux series and applications. *Journal of Symbolic Computation*, 47(1):32 – 63, 2012.

[25] A. Poteaux and M. Weimann. Computing Puiseux series: a fast divide and conquer algorithm. *Annales Henri Lebesgue*, 4:1061–1102, 2021.

[26] A. Poteaux and M. Weimann. Local polynomial factorisation: improving the Montes algorithm. In *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, ISSAC '22. ACM, 2022.

[27] A. Sommese and C. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific, 2005.

[28] B. M. Trager. *Integration of Algebraic Functions*. PhD thesis, MIT, 1984.

[29] W. Tucker. A rigorous ODE solver and Smale's 14th problem. *Found. Comput. Math.*, 2:53–117, 2002.

[30] W. Tucker. *Validated Numerics: A Short Introduction to Rigorous Computations*. Princeton University Press, Princeton, 2011.

[31] J. B. van den Berg and J.-P. Lessard. Rigorous numerics in dynamics. *Notices Amer. Math. Soc.*, 62(9):1057–1061, 2015.