Theoretical Computer Science 491 (2013) 1–32

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Exact symbolic–numeric computation of planar algebraic curves

Eric Berberich *, Pavel Emeliyanenko, Alexander Kobel, Michael Sagraloff **

Max-Planck-Institut für Informatik, Campus E1 4, D-66123 Saarbrücken, Germany

ARTICLE INFO

Article history:
Received 2 July 2012
Received in revised form 10 April 2013
Accepted 18 April 2013
Communicated by V.Y. Pan

Keywords:
Algebraic curves
Topology computation
Arrangement
Polynomial systems
Numerical solver
Hybrid methods
Symbolic–numeric algorithms
Exact computation

ABSTRACT

We present a *certified and complete algorithm to compute arrangements of real planar algebraic curves*. It computes the decomposition of the plane induced by a finite number of algebraic curves in terms of a cylindrical algebraic decomposition. From a high-level perspective, the overall method splits into two main subroutines, namely an algorithm denoted Bisolve to isolate the real solutions of a zero-dimensional bivariate system, and an algorithm denoted GeoTop to compute the topology of a single algebraic curve.

Compared to existing approaches based on elimination techniques, we considerably improve the corresponding lifting steps in both subroutines. As a result, generic position of the input system is never assumed, and thus our algorithm never demands for any change of coordinates. In addition, we significantly limit the types of symbolic operations involved, that is, we only use resultant and gcd computations as purely symbolic operations. The latter results are achieved by combining techniques from different fields such as (modular) symbolic computation, numerical analysis and algebraic geometry.

We have implemented our algorithms as prototypical contributions to the C++-project Cgal. We exploit graphics hardware to expedite the remaining symbolic computations. We have also compared our implementation with the current reference implementations, that is, Lgp and Maple's Isolate for polynomial system solving, and Cgal's bivariate algebraic kernel for analyses and arrangement computations of algebraic curves. For various series of challenging instances, our exhaustive experiments show that the new implementations outperform the existing ones.

1. Introduction

Computing the topology of a real planar algebraic curve

$$C = V(f) = \{(x, y) \in \mathbb{R}^2 : f(x, y) = 0\} \tag{1.1}$$

can be considered as one of the fundamental problems in real algebraic geometry with numerous applications in computational geometry, computer graphics and computer aided geometric design. Typically, the topology of C is given in terms of a planar graph $\mathcal{G}_C$ embedded in $\mathbb{R}^2$ that is isotopic to C.[1] For a geometric-topological analysis, we further require

* Corresponding author. Tel.: +49 681 9325 1012; fax: +49 681 9325 1099.
** Corresponding author. Tel.: +49 681 9325 1006; fax: +49 681 9325 1099.
E-mail addresses: eric@mpi-inf.mpg.de (E. Berberich), asm@mpi-inf.mpg.de (P. Emeliyanenko), akobel@mpi-inf.mpg.de (A. Kobel), msagralo@mpi-inf.mpg.de (M. Sagraloff).

[1] Throughout the paper, we consider the stronger notion of an **ambient isotopy**, but omit the word "ambient". The graph $\mathcal{G}_C$ is ambient isotopic to C if there exists a continuous mapping $\phi : [0, 1] \times \mathbb{R}^2 \mapsto \mathbb{R}^2$ with $\phi(0, \cdot) = \mathrm{id}_{\mathbb{R}^2}$, $\phi(1, C) = \mathcal{G}_C$, and $\phi(t_0, \cdot) : \mathbb{R}^2 \mapsto \mathbb{R}^2$ is a homeomorphism for each $t_0 \in [0, 1]$. Notice that, for an ambient isotopy, it is crucial that the deformation takes place in the ambient space (here $\mathbb{R}^2$) in which the objects are embedded.

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2013.04.014

the vertices of $\mathcal{G}_C$ to be located on C.[2] In this paper, we study the more general problem of computing an arrangement of a finite set of algebraic curves, that is, the decomposition of the plane into cells of dimensions 0, 1 and 2 induced by the given curves. The proposed algorithm is *certified* and *complete*, and the overall arrangement computation is exclusively carried out in the initial coordinate system. Efficiency of our approach is shown by implementing our algorithm based on the current reference implementation within CGAL[3] (see also [22,60]) and comparing it to the most efficient implementations which are currently available.

From a high-level perspective, we follow the same approach as in [22,60]. That is, the arrangement computation is reduced to the geometric-topological analysis of single curves and of pairs of curves. The main contribution of this paper is to provide solutions for the basic subtasks needed by these analysis, that is, *isolating the real solutions of a bivariate polynomial system* (BISOLVE) and *computing the topology of a single algebraic curve* (GEOTOP).

BISOLVE: For a given *zero-dimensional* polynomial system $f(x, y) = g(x, y) = 0$ (i.e. there exist only finitely many solutions), with $f, g \in \mathbb{Z}[x, y]$, the algorithm computes disjoint isolating boxes $B_1, \ldots, B_m \subset \mathbb{R}^2$ for *all real solutions*, where each box B_i contains exactly one solution. In addition, the boxes can be refined to an arbitrarily small size. BISOLVE is a classical elimination method which follows the same basic idea as the GRID method from [21] for solving a bivariate polynomial system, or the INSULATE method from [54] for computing the topology of a planar algebraic curve.[4] Namely, all of the latter mentioned methods consider several projection directions to derive a set of candidates of possible solutions and eventually identify those candidates which are actually solutions.

For BISOLVE, we separately eliminate the variables x and y by means of resultant computations. Then, for each possible candidate (represented as a pair of projected solutions in x- and y-direction), we check whether it actually constitutes a solution of the given system or not. The proposed method comes with a number of improvements compared to the aforementioned approaches and also to other existing elimination techniques [22,3,37,46,47]. First, we considerably reduce the amount of purely symbolic computations, namely, our method only requires resultant computations of bivariate polynomials and the computation of gcds for univariate polynomials. Second, our implementation profits from a recent approach [25–27] to compute resultants and gcds exploiting the power of Graphics Processing Units (GPUs). Here, it is important to remark that, in comparison to the classical resultant computation on the CPU, the GPU implementation is typically more than 100-times faster. Our experiments show that, for the huge variety of considered instances, the symbolic computations are no longer a "global" bottleneck of an elimination approach: For a majority of the considered instances, the running time for the purely symbolic operations is less than 10% of the total running time, whereas finding the roots of the resultant polynomial becomes the major bottleneck (with a running time of more than 50%). Third, the proposed method never uses any kind of a coordinate transformation, even for systems with two solutions sharing the same x- or y-coordinate. The latter fact is due to a novel inclusion predicate which combines information from the resultant computation and a homotopy argument to prove that a certain candidate box is isolating for a solution. Since we never apply any change of coordinates, our algebraic tools, such as computing resultants, particularly profit in cases where f and g are sparse. Since our inclusion predicate is local, our method also suits very well for finding solutions within a given local box. In these cases, real root isolation is restricted to the intervals defined by the projection of the box.[5] Finally, we integrated a series of additional filtering techniques which allow us to considerably speed up the computation for the majority of instances.

GEOTOP: There exist a number of certified and complete approaches to determine the topology of an algebraic curve; we refer the reader to [2,15,16,23,31,35,44] for recent work and further references. At present, only a few of these methods also extend to the computation of arrangements of arbitrary algebraic curves [2,22]. Common to most of the above mentioned approaches is that, in essence, they consider the following three phases:

1. *Projection*: Elimination techniques (e.g. resultants) are used to project the *x-critical points* (i.e. points p on the (complex) curve $C = \{(x, y) \in \mathbb{C}^2 : f(x, y) = 0\}$ with $f_y(p) = 0$) of the curve into one dimension. The so obtained projections are called *x-critical values*.
2. *Lifting*: For all real x-critical values α (as well as for real values in between), we compute the *fiber*, that is, all intersection points of C with a corresponding vertical line $x = \alpha$.
3. *Connection* (in the topology computation of a single curve): The so obtained points are connected by straight line edges in an appropriate manner.

In general, the lifting step at an x-critical value α has turned out to be the most time-consuming part because it amounts to determining the real roots of a non square-free univariate polynomial $f_\alpha(y) := f(\alpha, y) \in \mathbb{R}[y]$ with algebraic

---

[2] In general, points located on C are not exactly representable by floating point or rational numbers. Therefore, our algorithm returns all vertices of $\mathcal{G}_C$ in terms of isolating intervals. More precisely, for a vertex (α, β) of the graph $\mathcal{G}_C$, it provides a polynomial $R^{(y)} \in \mathbb{Z}[x]$, and intervals $I(\alpha) \subset \mathbb{R}$ and $I(\beta) \subset \mathbb{R}$ such that $I(\alpha)$ isolates α as a root of $R^{(y)}$, and $I(\beta)$ isolates β as a root of $f(\alpha, y) \in \mathbb{R}[y]$.

[3] Computational Geometry Algorithms Library, www.cgal.org; see also http://exacus.mpi-inf.mpg.de/cgi-bin/xalci.cgi for an online demo on arrangement computation.

[4] For the topology computation of a planar curve $C = \{(x, y) \in \mathbb{R}^2 : f(x, y) = 0\}$, it is crucial to find the solutions of $f = f_y = 0$. The method in [54] uses several projection directions to find these solutions.

[5] Solutions on the boundary of a box with rational corners can be detected using an arbitrary real root isolator such as the Descartes method.