Arrangements on Surfaces of Genus One: Tori and Dupin Cyclides

Eric Berberich* Michael Kerber*

Abstract

An algorithm is presented to compute the exact arrangement induced by arbitrary algebraic surfaces on a parametrized ring Dupin cyclide, including the special case of the torus. The intersection of an algebraic surface of degree n with a reference cyclide is represented as a real algebraic curve of bi-degree $(2n, 2n)$ in the cyclide's two-dimensional parameter space. We use Eigenwillig and Kerber [11] to compute a planar arrangement of such curves and extend their approach to obtain more asymptotic information about curves approaching the boundary of the cyclide's parameter space. With that, we can base our implementation on a general software framework by Berberich et. al. [3] to construct the arrangement on the cyclide. Our contribution provides the demanded techniques to model the special topology of the reference surface of genus one. Our experiments show no combinatorial overhead of the framework, i.e., the overall performance is strongly coupled to the efficiency of the implementation for arrangements of algebraic plane curves.

1 Introduction

Consider a surface S in $\mathbb{R}^3$ and a set C of curves on S. The arrangement $\mathcal{A}(C)$ is the subdivision of S into cells of dimensions zero, one, and two with respect to C. The cells are called vertices, edges, and faces, respectively.

Berberich et al. [3] introduced a general software framework for sweeping a set of curves on a parametric surface S. We present an implementation for the case that S is a ring Dupin cyclide and the arrangement on it is induced by intersections of S with algebraic surfaces of arbitrary degree. Our approach always computes the exact arrangement, undistorted by rounding errors, of the given input. It also handles all degeneracies like singular points or intersections with high multiplicity.

Dupin Cyclides have been introduced by Dupin [9] as surfaces whose lines of curvature are all circular. One can think of a (ring) Dupin cyclide as a torus with variable tube radius. Dupin cyclides are the generalization of the "natural" geometric surfaces like planes, cylinders, cones, spheres and tori, what makes them useful for applications in solid modeling; compare, e.g., [6], [15], [16], [8].

Our algorithm is this: we follow the framework of [3], and perform a sweep-line algorithm [2] on the intersection curves of the Dupin cyclide with the surfaces in the parameter space. The primitives of the sweep are specified by a model of the *GeometryTraits* concept which is given by the recent work of Eigenwillig and Kerber [11]. With that model, one can sweep over algebraic plane curves of arbitrary degree. The applied sweep line algorithm interacts with a model of the *TopologyTraits* concept; this model controls the creation and manipulation of arrangement features at the boundary of the parameter space, i.e., *identifications* in our case. We implemented such a model for the case of a Dupin cyclide. The arrangement on the Dupin cyclide is represented by a doubly-connected edge-list (DCEL), where points are attached to vertices and curves are stored with edges. Our implementation in C++ deeply benefits from generic programming capabilities, i.e., we are using CGAL's[1] class template `Arrangement_on_surface_2` that expects proper models of the `GeometryTraits` and the `TopologyTraits` concept.

Related work: Arrangements in the plane have been well studied during the past decades, and also quite a number of exact and efficient implementations appeared [13]. Two-dimensional arrangements on surfaces, especially with exact implementation, became more popular recently, e.g., arrangements of great arcs on a sphere [3], arrangements of small arcs on a sphere by Cazals and Loriot [7]. The most complicated surfaces considered so far are arrangements induced by quadrics intersecting a reference quadric. Three approaches exist. The first actually computes more, namely the adjacency relationship between intersection of a set of quadrics [10]. The other two project the intersection curves onto the xy-plane. The original work [4] maintains two arrangements, one for the lower part of the reference quadric and one for its upper part; a connection between them is missing. Instead, [3] introduces a small extension of the projection to simulate the parameter space of the reference quadric. This way, it benefits from the framework that we also apply for ring cyclides. Instead of such a simulation, the sweep on a cyclide is explicitly performed in parameter space.

A more detailed version of this paper appears in [5].

*Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany, email: {eric,mkerber} @mpi-inf.mpg.de

[1] See the project homepage: http://www.cgal.org

2 Dupin cyclides

The maybe most intuitive way of constructing a Dupin cyclide (called *cyclide* for brevity) goes back to Maxwell, we cite it from Boehm [6]:

> Let a sufficiently long string be fastened at one end to one focus f of an ellipse, let the string be kept always tight while sliding smoothly over the ellipse, then the other end z sweeps out the whole surface of a cyclide Z.

Note that choosing a circle in this construction yields a torus. We assume that the cyclide is in *standard position and orientation*, i.e., the chosen base ellipse is given by $(x/a)^2 + (y/b)^2 = 1, \quad a \geq b > 0$.

We define $c := \sqrt{a^2 - b^2}$, and μ as the length of the string minus a. We assume that $c < \mu \leq a$ which means that the cyclide has no self-intersections (it is a ring cyclide).

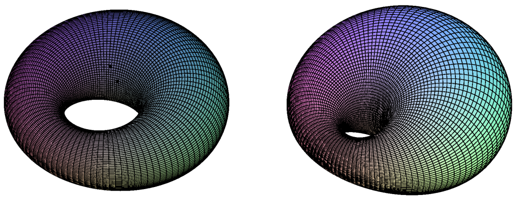

Figure 2.1: (Left) Cyclide with $a = 1$, $b = 0.99$, $\mu = 0.5$, (Right) Cyclide with a=13, $b = 12$, $\mu = 9$

The parameterization of the cyclide [14] is given by

$$\begin{pmatrix} \phi \\ \psi \end{pmatrix} \mapsto \begin{pmatrix} \frac{\mu(c-a\cos\phi\cos\psi)+b^2\cos\phi}{a-c\cos\phi\cos\psi} \\ \frac{b(a-\mu\cos\psi)\sin\phi}{a-c\cos\phi\cos\psi} \\ \frac{b(c\cos\phi-\mu)\sin\psi}{a-c\cos\phi\cos\psi} \end{pmatrix}$$

with $\phi, \psi \in [-\pi, \pi]$. For $\phi = \pi$ and $\phi = -\pi$, this yields the *tube circle* $(x + a)^2 + z^2 = (\mu + c)^2$ in the plane $y = 0$, for $\psi = \pi$ and $\psi = -\pi$, it yields the *outer circle* $(x + c)^2 + y^2 = (a + \mu)^2$ in the plane $z = 0$. The tube circle and the outer circle meet in the *pole* $p := (-\mu - c - a, 0, 0)$.

To get a rational parameterization of the cyclide without trigonometric functions, we use the identities

$$\cos\theta = \frac{1 - \tan^2\frac{\theta}{2}}{1 + \tan^2\frac{\theta}{2}} \quad \sin\theta = \frac{2\tan\frac{\theta}{2}}{1 + \tan^2\frac{\theta}{2}},$$

and set $u := \tan\frac{\phi}{2}, v := \tan\frac{\psi}{2}$. We write the obtained parametrization in homogeneous coordinates, i.e., the common denominator is written as a separate variable: Define $u_+ := 1 + u^2$, $u_- := 1 - u^2$, $v_+ := 1 + v^2$ and $v_- := 1 - v^2$ then $\hat{P} : \mathbb{R}^2 \to \mathbb{R}^4$ is given by

$$\begin{pmatrix} u \\ v \end{pmatrix} \mapsto \begin{pmatrix} \mu(cu_+v_+ - au_-v_-) + b^2u_-v_+ \\ 2u(av_+ - \mu v_-)b \\ 2v(cu_- - \mu u_+)b \\ au_+v_+ - cu_-v_- \end{pmatrix}$$

The image of $\hat{P}$ is the cyclide without the tube circle and the outer circle. Intuitively, the cyclide is cut along the outer circle and the tube circle, and "rolled out" to the plane. Therefore, we call the outer circle and the tube circle the *cut circles*. Paths on the cyclide crossing the cut circles correspond to paths in the parameter space crossing the infinite boundary. More precisely, an intersection with the tube (outer) circle causes a horizontally (vertically) asymptotic path in the parameter space. Paths passing through the cyclide's pole correspond to paths converging to one of the "corners" $(\pm\infty, \pm\infty)$ in parameter space.

3 Our implementation

We use the software framework implemented in CGAL's new `Arrangement_on_surface_2` package [3]. It provides an arrangement class that can be used to construct, maintain, overlay, and query two-dimensional arrangements on a parametric surface. It conceptually performs a sweep in the parameter space, i.e., a line $u = u_0$ is swept to the right through the parameter space.

Special diligence is needed for such curves at boundaries of the parameter space. The parameter space of the cyclide contains so called *identifications* of both pairs of opposite boundaries, i.e., for its parameterization P_S, it holds $\forall v \in V, P_S(u_{\min}, v) = P_S(u_{\max}, v)$ and $\forall u \in U, P_S(u, v_{\min}) = P_S(u, v_{\max})$, so for each point on the outer- and the tube-circle there exist two pre-images (four for the pole) in parameter space. This leads to problems for the sweep, since the event queue of the sweep line algorithms needs a unique order, and since only one DCEL-vertex should be constructed for each multiple pre-image. The modularity of CGAL's new `Arrangement_on_surface_2` package tackles these problems. To instantiate the package's main class, models of two concepts must be provided as template parameter.

First, the *GeometryTraits* fullfills the CGAL's ARRANGEMENTTRAITS_2 concept. It defines the types `Curve_2`, `X_monotone_curve_2`, and `Point_2`, and provides predicates and constructions on points and subcurves, e.g., lexicographic comparison of two points, or the construction of all intersection of two x-monotone curves.

Second, the *TopologyTraits* is responsible to determine the underlying DCEL-representation, to create the empty representation and to construct and maintain DCEL-features related to the boundary of the parameter space.

We describe next our models for both concepts:

GeometryTraits: We aim to represent the curves on the cyclide as algebraic curves in parameter space, and to realize the geometric predicates by computations in the parameter space.

For a cyclide with homogeneous parametrization

$\hat{P}$ and a surface F with homogeneous equation $\hat{F} \in \mathbb{Z}[x, y, z, w]$, the cut curve in the parameter space is implicitly defined by $f := \hat{F}(\hat{P}(u, v)) \in \mathbb{Z}[u, v]$. The resulting curve f has bidegree up to $(2 \cdot \deg F, 2 \cdot \deg F)$, hence we need a model of CGAL's ArrangementTraits_2 concept for algebraic curves in $\mathbb{R}^2$, regardless of their degree.

Such a model has recently been provided by Eigenwillig and Kerber [11] based on the observation that all required operations emerge from the topological and geometric analyses of single curves [12] and pairs of them. No condition is imposed on the input, i.e., curves can have arbitrary degree, and contain degeneracies, like covertical intersections, vertical asymptotes and isolated points.

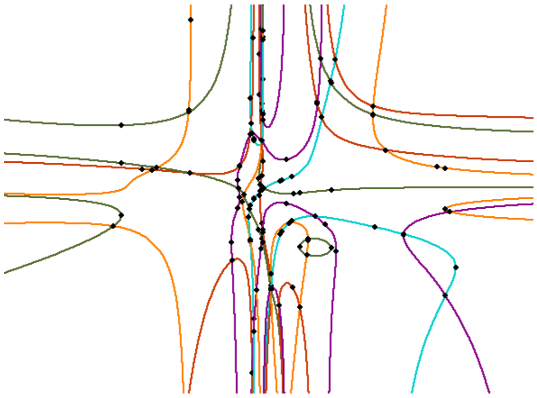

Figure 3.1: Cut-out of an arrangement in the parameter space of a cyclide, induced by 5 cubic surfaces

TopologyTraits: We provide a new TopologyTraits class for the cyclide. To handle the identifications at boundaries, it maintains two sorted sequences of DCEL-vertices to store the intersection of intersection curves with the cut circles. The position of such intersections is determined by horizontal and vertical asymptotes of curve-ends approaching infinity, the boundary of the parameter space.

Whenever the arrangement detects a curve-end approaching a cut circle, it asks the topology traits whether a DCEL-vertex is already stored for this position. If not, a new one is created and stored in the proper sequence; if yes, that one is used and the identification interactively takes place.

The analysis of curves contains the information about vertical asymptotes of curves (compare [12]), but not about horizontal asymptotes. We also implemented this step, with the following idea: there can be only finitely many positions where horizontal asymptotes might appear, since they are roots of a leading coefficient with respect to u. The curve-ends of arcs towards $u = \pm\infty$ can be assigned to such asymptotes by analysing the curve at some value u_0 "far" on the left or on the right.

Instance	#S	#V,#E,#F	t	t (2D)
ipl-1	10	119,190,71	**0.14**	0.14
ipl-1	20	384,682, 298	**0.58**	0.58
ipl-1	50	1837,3363,1526	**2.14**	2.00
ipl-2	10	358,575,217	**1.07**	1.25
ipl-2	20	1211,2147,937	**3.14**	3.04
ipl-3	10	542,847,305	**4.84**	4.62
ipl-3-6points	10	680,1092,412	**32.43**	31.17
ipl-3-2sing	10	694,1062,368	**5.82**	5.57
ipl-4	10	785,1204,419	**50.42**	49.97
ipl-4-6points	10	989,1529,540	**461.74**	450.54
ipl-4-2sing	10	933,1471,538	**53.01**	52.78

Table 1: Running times (in seconds) to construct arrangements on S_1 induced by algebraic surfaces

4 Results

We performed tests of our C++ implementation, executed on an AMD Dual-Core Opteron(tm) 8218 multi-processor Debian Etch platform, each core equipped with 1 MB internal cache and clocked at 1 GHz. The total memory consists of 32 GB. As compiler we used `g++` in version 4.1.2 with flags `-O2 -DNDEBUG`. Two results were computed for each instance, one that computes the arrangement using the *cyclidean topology* (`onSurface`), the other is computing the two-dimensional arrangement of the induced intersection curves *in parameter space*, i.e., with the topology of an unbounded plane (`Arrangements`). Our implementation allows to translate and rotate the reference cyclide in space. For the experiments presented in this work, we use the torus S_1 with $a = 2$, $b = 2$, $\mu = 1$, centered at the origin and the cyclide S_2 with $a = 13$, $b = 12$, $\mu = 11$, translated by a rational vector and rotated by a rational matrix.

We interpolated surfaces of fixed degree by randomly chosing points on a three-dimensional grid, having no or some degeneracies wrt S_1: the surfaces in "6points" instances share at least 6 common points, one of them is the pole of S_1. The surfaces in the "2sing" instances induce (at least) two singular intersections. The running times are listed in Table 1 that show good behavior of the implementation, even for higher degree surfaces. Degeneracies with respect to the reference surface result in higher running times as the instance "6points" shows. But this effect already appears in parameter space, as the last column indicates. In general, it is remarkable that in all tested instances, the spent time on the cyclides is (almost) identical to the computation of the curves in their parameter space. This let us conclude that the cyclidean topology is as efficient as the one for the unbounded plane and that the extra computation of horizontal asymptotes seems to be a cheap task. Most time is spent for geometric operations on algebraic curves. Thus, we infer that the chosen approach strongly hinges on the efficiency of the underlying 2D-implementation for arrangements of algebraic curves.

Instances	#S	#V,#E,#F	t
quadrics	10	428,646,219	**1.59**
degree-3	5	240,314,74	**1.56**
Overlay	-	942,1508,566	**1.91**
degree-3	10	794,1218,424	**6.25**
degree-4	10	325,418,93	**13.36**
Overlay	-	1623,2644,1021	**13.83**
degree-4	10	816,1188,372	**50.86**
degree-4	5	325,418,93	**13.52**
Overlay	-	1581,2488,907	**47.30**

Table 2: Running times (in seconds) to construct arrangements induced by algebraic surfaces of different degree on S_2, and to overlay them afterwards.

We also generated instances of random surfaces with degree up to 4 intersecting S_2, picked two of them, computed their arrangement and also the overlay of these arrangements. Reading Table 2, one sees that the overlay step is usually faster than the two initial constructions, as only a few new pairs of algebraic curves have to be analyzed newly.

Finally, we remark, that we also can immediately use other techniques implemented for CGAL's `Arrangement_on_surface_2`, such as point location, extending the DCEL by user data, and notifications.

5 Conclusion

Our work demonstrates the usefulness of generic programming: the combination of the planar arrangement algorithm for arbitrary curves with the software framework for arrangement on surfaces yields an arrangement algorithm for tori and Dupin cyclides almost immediately. New code was only written for the computation of the parameterized intersection curves, for the asymptotic behavior of infinite curve arcs, and for the topology traits of the cyclide. Relying on already tested and optimized code reduces the implementation effort, and makes the algorithm more robust and more efficient. We are already working on the adaptation of our traits classes with respect to the next version of the framework that will support geometric objects on identifications.

We also believe that the performance could be further improved: the computed arrangements often contain numerous vertically asymptotic arcs (compare Figure 3.1). The strategy proposed in [12] to shear non-regular curves and shearing back afterwards therefore results in a change of coordinates for many curves. A comparably efficient alternative approach that avoids to shear might be more suitable for this special subclass of curves.

Acknowledgements

We want to thank Ron Wein, who answered our questions on CGAL's arrangements in depth.

References

[1] L. Arge, M. Hoffmann, E. Welzl (eds.): *Algorithms - ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings*, *LNCS*, vol. 4698. Springer, 2007.

[2] J. L. Bentley, T. A. Ottmann: "Algorithms for Reporting and Counting Geometric Intersections". *IEEE Trans. on Computers* **C-28** (1979) 643–647.

[3] E. Berberich, E. Fogel, D. Halperin, K. Mehlhorn, R. Wein: "Sweeping and Maintaining Two-Dimensional Arrangements on Surfaces: A First Step". In: Arge et al. [1], 2007 645–656.

[4] E. Berberich, M. Hemmer, L. Kettner, E. Schömer, N. Wolpert: "An Exact, Complete and Efficient Implementation for Computing Planar Maps of Quadric Intersection Curves". In: *Proc. of the 21st Annual Symp. on Comput. Geom. (SCG 2005)*, 2005 99–106.

[5] E. Berberich, M. Kerber: "Exact Arrangements on Tori and Dupin Cyclides". In: *Proc. of the 2005 ACM Symp. on Solid and Physical Modeling, (SPM 08)*, 2008, to appear

[6] W. Boehm: "On Cyclides in Geometric Modeling". *Computer Aided Geometric Design* **7** (1990) 243–255.

[7] F. Cazals, S. Loriot: *Computing the exact arrangement of circles on a sphere, with applications in structural biology.* Technical Report 6049, INRIA Sophia-Antipolis, 2007.

[8] V. Chandru, D. Dutta, C. M. Hoffmann: "On the geometry of Dupin cyclides". *The Visual Computer* **5** (1989) 277–290.

[9] C. Dupin: *Applications de Géométrie et de Méchanique.* Bachelier, Paris, 1822.

[10] L. Dupont, M. Hemmer, S. Petitjean, E. Schömer: "Complete, Exact and Efficient Implementation for Computing the Adjacency Graph of an Arrangement of Quadrics". In: Arge et al. [1], 2007 633–644.

[11] A. Eigenwillig, M. Kerber: "Exact and Efficient 2D-Arrangements of Arbitrary Algebraic Curves". In: *Proc. of the Nineteenth Annual ACM-SIAM Symp. on Discrete Algorithms (SODA08)*, 2008 122–131.

[12] A. Eigenwillig, M. Kerber, N. Wolpert: "Fast and Exact Geometric Analysis of Real Algebraic Plane Curves". In: C. W. Brown (ed.) *Proocedings of the 2007 International Symposium on Symbolic and Algebraic Computation (ISSAC 2007)*, 2007 151–158.

[13] E. Fogel, D. Halperin, L. Kettner, M. Teillaud, R. Wein, N. Wolpert: "Arrangements". In: J.-D. Boissonnat, M. Teillaud (eds.) *Effective Computational Geometry for Curves and Surfaces*, chap. **1**, 1–66. Spinger, 2006.

[14] A. Forsyth: *Lectures on the Differential Geometry of Curves and Surfaces.* Cambridge Univ. Press, 1912.

[15] M. J. Pratt: "Cyclides in computer aided geometric design". *Computer Aided Geometric Design* **7** (1990) 221–242.

[16] M. J. Pratt: "Cyclides in computer aided geometric design II". *Computer Aided Geometric Design* **12** (1995) 131–152.