On the computation of an arrangement of quadrics in 3D [1]

Bernard Mourrain and Jean-Pierre Técourt and Monique Teillaud

INRIA, GALAAD, BP 93, 06902 Sophia Antipolis cedex, FRANCE
{*Bernard.Mourrain, Jean-Pierre.Tecourt, Monique.Teillaud*}*@sophia.inria.fr*
http://www-sop.inria.fr/galaad/

Abstract

In this paper, we study a sweeping algorithm for computing the arrangement of a set of quadrics in $\mathbb{R}^3$. We define a "trapezoidal" decomposition in the sweeping plane, and we study the evolution of this subdivision during the sweep. A key point of this algorithm is the manipulation of algebraic numbers. In this perspective, we put a large emphasis on the use of algebraic tools, needed to compute the arrangement, including Sturm sequences and Rational Univariate Representation of the roots of a multivariate polynomial system.

Key words: arrangement, quadric, implicit surface, algebraic number, rational univariate representation, certified computation.

1 Introduction

Arrangements are the underlying structures of many applications, such as robot motion planning, Computer Aided Geometric Design, Computer Biology,... They have been extensively studied in the literature, either theoretically or from a practical point of view, essentially on linear objects. See [15] for a survey.

[1] *This work is partially supported by the IST Program of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-2000-26473 (ECG - Effective Computational Geometry for Curves and Surfaces)* `http://www-sop.inria.fr/prisme/ECG/`

By definition, the arrangement of a set $\mathcal{S}$ of objects in $\mathbb{R}^d$ is the decomposition of $\mathbb{R}^d$ into connected components of dimensions $0, 1, \ldots, d$ where the the input polynomials defining the sites of $\mathcal{S}$ have constant sign.

The manipulation of quadrics plays an important role in solid modeling, since it offers a compromise between the treatment of simple but numerous triangles in meshes and powerful but costly algebraic objects. This motivated recent and interesting developments in computational geometry for algebraic surfaces. Geismann *et al.* presented two methods to compute a given cell in an arrangement of quadrics [12]. The first method uses projection techniques based on resultants, while the second method uses solid modeling techniques. Dupont *et al.* compute a near-optimal (in the number and depth of radicals involved) parameterization of the intersection of quadrics [9]. Their method is implemented in [17] and theoretical bounds on the size of the output coefficients are proven.

The topology of an arrangement is often quite complex, and the description of a given cell can be of non-constant size. Therefore, *vertical decompositions* are often used, allowing to partition the space into simpler constant-sized cells [27]. A sweep-based algorithm was proposed by Shaul and Halperin to produce a vertical decomposition of an arrangements of triangles in $\mathbb{R}^3$. They mentioned that the sweeping approach they used in the case of triangles could be generalized to compute the vertical decomposition of the arrangement of n *well-behaved surface patches* in time $O(n \log^2 n + V \log n)$, where V is the combinatorial complexity of the vertical decomposition [28]. We propose to use this sweeping plane approach to compute the arrangement of a set of quadrics in $\mathbb{R}^3$, which yields an adaptive approach to the arrangement computation problem, in contrary to the previous related work [12]. Note also that the vertical decomposition allows fast location methods.

When applying the standard sweeping technique to the case of an arrangement of quadrics in $\mathbb{R}^3$, the main issue is —as often when applying computational geometry techniques to curved objects— the implementation of the geometric primitives: predicates and constructions. In this paper, we describe the algorithm more precisely for the case of quadrics, and we show how the detection of events and the point location in the sweeping plane can be translated algebraically. We characterize the geometric primitives needed by the algorithm, and we analyze them, in terms of the degree of the algebraic numbers and of the height of the algebraic expressions involved in the computations. We show in particular that comparing events boils down in the worst case to comparing algebraic numbers of degree 16. These aspects were not taken into account in the previous work in the literature.

The algorithm is described in Section 2, where we show how and why we

compute the so-called vertical decomposition of the arrangement. Algebraic issues related to the resolution of polynomial equations and geometric predicate evaluation are discussed more precisely in Section 3. We conclude with future work.

2 Algorithm

We first recall some basic definitions and notation regarding quadrics. Then we give a rough overview of the sweep. Section 2.3 focuses on the decomposition in the sweeping plane and on how events are detected. We show in Section 2.4 that the 3D decomposition that is computed by the algorithm is the vertical decomposition of the arrangement of quadrics. Section 2.5 summarizes the algorithm and Section 2.6 concludes by giving its combinatorial complexity.

2.1 Preliminaries

We denote by (x, y, z) the coordinates of a point of $\mathbb{R}^3$ in the canonical basis.

A quadric Q is the set of points of $\mathbb{R}^3$ that are solutions of a degree 2 polynomial equation in the variables x, y, z, with coefficients in $\mathbb{Z}$. The equation is given by 10 real coefficients, up to a non-zero multiplicative scalar factor, and the polynomial defining the quadric will be also denoted by Q:

$$Q(x, y, z) = ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d.$$

We denote by ∇Q the gradient vector of Q. The partial derivatives of Q are noted $\partial_x(Q)$, $\partial_y(Q)$ and $\partial_z(Q)$.

Equivalently, Q is given by a symmetric matrix

$$M_Q = \begin{pmatrix} a & h & g & p \\ h & b & f & q \\ g & f & c & r \\ p & q & r & d \end{pmatrix}.$$

The upper left sub-matrix of size 3×3 is called the principal sub-matrix of M_Q and denoted as M_Q^s.

The *inertia* or *signature* of M_Q (resp. M_Q^s) is defined as the pair (σ^+, σ^-), where σ^+ and σ^- are respectively the number of positive and negative eigenvalues of M_Q (resp. M_Q^s).

The type of a quadric Q (ellipsoid, hyperboloid, planes, and so on) depends on the inertias of M_Q and M_Q^s. More precisely, a quadric in $\mathbb{R}^3$ is entirely characterized by the inertia of its matrix M_Q and its principal sub-matrix M_Q^s. A complete description of this characterization is recalled in [17] (Table 2).

A cylinder is defined as the surface swept by a planar curve moved in translation along a line orthogonal to the plane of the curve. When the curve is a conic, the swept cylinder is a quadric Q. The inertia of M_Q is (2,1) or (1,2) and the inertia of M_Q^s is different from (2,1) and (1,2).

In this paper, we consider all kinds of quadrics, even degenerate, except the case of a double plane (inertia of M_Q: (1,0) or (0,1), inertia of M_Q^s: (1,0) or (0,1)), which would lead to deal with non square-free polynomials, and to some additional techniques to compute critical points (see Section 3). The very special case of a line, which is a quadric though it is not a surface in the usual sense (inertia of M_Q: (2,0) or (0,2), inertia of M_Q^s: (2,0) or (0,2)) could be treated but leads to particular cases in the algorithmic description, and so, it will be omitted in this paper.

Hereafter, we will also assume that the sweeping direction is generic, so that for every position of the sweeping plane, there is at most one point tangent to a given quadric on this plane (see next section).

2.2 Sweeping a set of quadrics

Let $\mathcal{S} = \{Q_i, i = 1, \ldots, n\}$ be a set of n quadrics. We are going to sweep $\mathcal{S}$ by a plane orthogonal to the x-axis. Every section of the arrangement by the sweeping plane is an **arrangement of conics** in a (y, z)-plane. The conics are continuously evolving during the sweep: the equation of a conic for a given position $x = \mathsf{s}$ of the plane depends on the parameter s; it is given as $Q_i(\mathsf{s}, y, z) = 0$, which is a polynomial equation in the coordinates y, z of a point in the planar section. Such a conic will be also denoted as $Q_i^{(\mathsf{s})}$.

Let us give first a general overview of the algorithm. We are going to maintain the combinatorial structure of the evolving arrangement of conics, by analyzing the 3D events where a topological change occurs:

a) The sweeping plane is tangent to a quadric Q_i.
Algebraically, such an event corresponds to a value s of x (defined as a

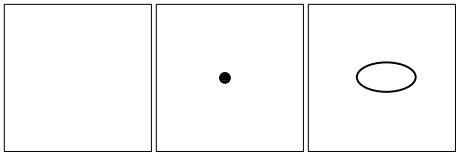
x-critical value) for which there is a solution (y, z) to the system

$$\begin{cases} Q_i(\mathsf{s}, y, z) & = 0 \\ \partial_y(Q_i)(\mathsf{s}, y, z) & = 0 \\ \partial_z(Q_i)(\mathsf{s}, y, z) & = 0 \end{cases}$$

Depending on the type of quadric (or equivalently to the inertia of M_{Q_i} and $M^s_{Q_i}$), the modification in the topology of the arrangement of conics correspond to different configurations.

On the following pictures, some kinds of events are represented. The situations before, at and after the x-critical values are drawn.

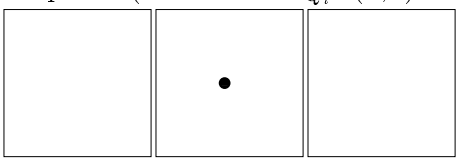
i. ellipsoid, paraboloid, hyperboloid of two sheets (inertia of M_{Q_i}: (3,1) or (1,3))



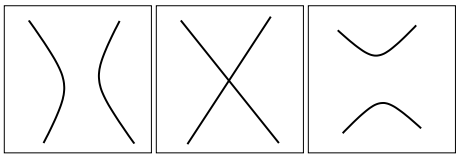
A conic appears or disappears

Note that the locus of points that are first encountered by the sweeping plane is supposed to be a single point in this figure. This will be achieved if the x-axis is chosen among *generic* directions. This will be developed further at the end of this section.

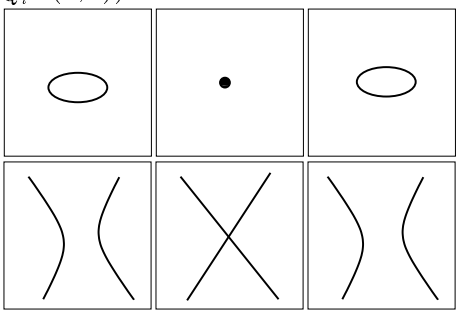
ii. quadric reduced to a point (inertia of M_{Q_i}: (3,0) or (0,3))



iii. hyperbolic paraboloid or hyperboloid of one sheet (inertia of M_{Q_i}: (2,2))



iv. cone (inertia of M_{Q_i}: (2,1))



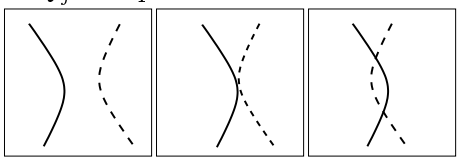
b) The curve of intersection of two quadrics Q_i and Q_j is tangent to the

sweeping plane.

Such an event corresponds to a value s of x for which there is a solution (y, z) of the system

$$\begin{cases} Q_i(\mathsf{s}, y, z) & = 0 \\ Q_j(\mathsf{s}, y, z) & = 0 \\ (\nabla Q_i \bigwedge \nabla Q_j)_x(\mathsf{s}, y, z) & = 0 \end{cases}$$

The last equation of this system means that the x-component of the vector product of ∇Q_i and ∇Q_j is equal to 0.

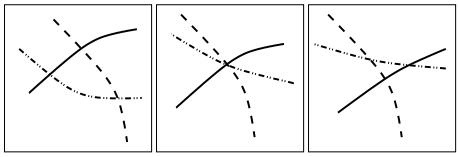


Before or after the event, we have two conics corresponding to the intersection of the quadrics with the sweeping plane. At the event, the two curves become tangent.

c) The sweeping plane contains an intersecting point of three quadrics.

The value s of x is such that there is a solution (y, z) to the system

$$\begin{cases} Q_i(\mathsf{s}, y, z) & = 0 \\ Q_j(\mathsf{s}, y, z) & = 0 \\ Q_k(\mathsf{s}, y, z) & = 0 \end{cases}$$



Before or after the event, we have three conics in the plane that intersect pairwise. At the event, the three conics have a common point.

We do not consider here *degenerate configurations of the set $\mathcal{S}$ of quadrics*, such as:

- several quadrics are tangent, so, more than one quadric is encountered at a given position of the plane (event a.i),
- more than three quadrics intersect at the same point (event c),
- several events coincide, for instance when three quadrics intersect at a point (event c) that is also a point where a quadric is first encountered (event a.i).

Let us give more precisions now on the *generic directions* allowed for the sweep. We assume that the x direction is chosen in such a way that for each

quadric, the number of points where the plane is tangent to the quadric is at most a single point (see case (a.i)).

More generally, we assume that for each event, the change of topology in the arrangement of conics in the sweeping plane is of **constant size**, which is crucial in our algorithm. Concretely, for:

- cylinders (elliptic, hyperbolic, parabolic),
- cones,
- hyperboloids of one sheet and hyperbolic paraboloids,
  all the directions of planes that contain the direction of one line drawn on the cylinder are forbidden. However, the directions of planes containing two lines drawn on the cylinder are allowed (case a.iii).

For intersecting planes (inertia of M_Q: (1,1), inertia of M_Q^s: (1,1)), we also forbid directions of planes containing their line of intersection.

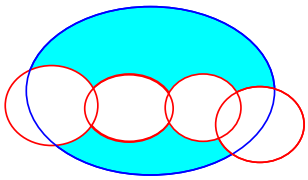
Let us consider the set of forbidden directions for the sweep algorithm on the sphere of directions. For each quadric, at most one curve drawn on this sphere is forbidden. Altogether, the set of generic directions consists of the union of open 2D-cells in the arrangement of at most n curves on the sphere. It is a dense subset of the sphere.

We will also need to avoid directions for the sweep such that two events for two different quadrics occur at the same time. Unfortunately, this cannot be detected before the sweep. If this occurs during the sweep, then a shearing of axes will be performed, similarly to what is done by Eigenwillig *et al* for the sweeping of curves in 2D [10].

2.3 From cells to "trapezoids"

We would like to characterize, by sign conditions, each connected cell of the arrangement of conics in a planar section.

The following picture shows that two different geometric cells (the two gray cells) can be characterized by exactly the same sign conditions.



So, locating a point in a section of the arrangement of quadrics requires more than only checking such sign conditions. Instead, we choose to compute a

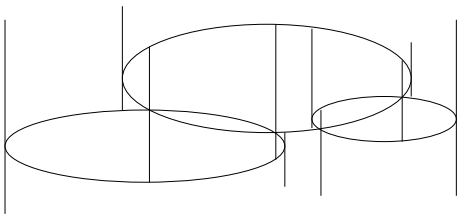


Fig. 1. Trapezoidal decomposition of conics in a plane

"trapezoidal" decomposition of the arrangement in the x-section, as explained in the following paragraph. Note that this trapezoidal decomposition in the x-section evolves during the sweep. Thus we need a description of the cells which applies for different values of x, based on sign functions in trapezoidal regions.

2.3.1 Trapezoids.

In the sweeping plane of equation $x = \mathsf{s}$, we draw segments parallel to the z-axis, in a very similar way as done usually for the trapezoidal map in the case of a planar arrangement of line segments (see Figure 1). A vertical segment, or a *wall* is drawn through:

- intersection points between two conics,
- points where the tangent to the conic is parallel to the z-axis.

The walls cut the conics into conic arcs, whose endpoints' y-coordinates are the y-coordinates of the walls. More precisely, the y-coordinate of an endpoint or a wall is represented implicitly as a y-root of

$$\left(S_{ij}^{(\mathsf{s})}\right) \begin{cases} Q_i(\mathsf{s}, y, z) = 0 \\ Q_j(\mathsf{s}, y, z) = 0 \end{cases} \text{ or } \left(S_k^{(\mathsf{s})}\right) \begin{cases} Q_k(\mathsf{s}, y, z) = 0 \\ \partial_z(Q_k(\mathsf{s}, y, z)) = 0 \end{cases}.$$

$\left(S_{ij}^{(\mathsf{s})}\right)$ is the polynomial system whose solutions are the y-coordinates of intersection points of two conics $Q_i^{(\mathsf{s})}$ and $Q_j^{(\mathsf{s})}$. $\left(S_k^{(\mathsf{s})}\right)$ is the system whose solutions are the y-coordinates of points where the tangent to a conic $Q_k^{(\mathsf{s})}$ is parallel to the z-axis.

In this way, we obtain "trapezoids" of constant size description, given by: two walls, a ceiling and a floor, which are (evolving) conic arcs, and signs used to determine if the trapezoid is above or below the conics.

Deciding whether a point lies in a trapezoid, reduces to comparing the y-coordinates of the point and the walls and then for a fixed y, to compute the sign of the conics defining the ceiling and the floor or the sign of rational

expressions formed on their coefficients. This will be developed in Section 3.3.

2.3.2 Detecting events.

The topology of the trapezoidal decomposition changes whenever the description of a trapezoid is modified, either because a new conic appears in it (events of type (a.i)), or because its ceiling and floor intersect, or because its walls coincide.

As already noticed in [28], all events, except events of type (a.i), do not need to be computed (and sorted) in advance. Indeed, we can use the same reasoning as in the case of the classic Bentley-Ottmann algorithm for sweeping line segments. All events except (a.i) correspond to moments when two features of the trapezoidal decomposition meet. This can occur only if, just before they meet, they are "contiguous" in the decomposition, meaning here that they belong to the same trapezoid.

All these events will be detected during the sweep, as follows: each time a new trapezoid is created, we compute the x for which it disappears, by computing the values when its walls coincide and when its ceiling and floor intersect (see Figure 2).

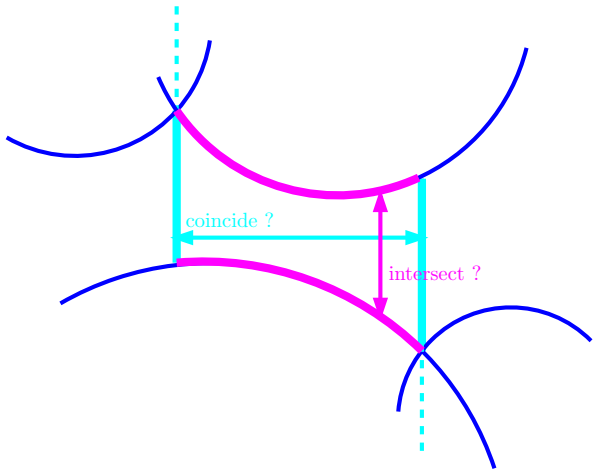


Fig. 2. Detection of future events

As mentioned above, for a given value s of x, a wall is defined by a y-root of a system $\left(S_{ij}^{(\mathsf{s})}\right)$ or $\left(S_{k}^{(\mathsf{s})}\right)$. Of these two cases, the system having the highest algebraic degree is the first, corresponding to the case when the wall is defined by an intersection of two conics. So, the worst type (in terms of algebraic degree) of event occurs when the y-coordinate of the intersection between two conics coincides with the y-coordinate of the intersection between two other

conics, which is expressed as follows:

$$\mathsf{s} \text{ such that } \exists y, \exists (z, z'), \begin{cases} Q_i(\mathsf{s}, y, z) = 0 \\ Q_j(\mathsf{s}, y, z) = 0 \end{cases} \text{ and } \begin{cases} Q_k(\mathsf{s}, y, z') = 0 \\ Q_l(\mathsf{s}, y, z') = 0 \end{cases} \quad (1)$$

The positions where the ceiling and the floor touch each other correspond to systems associated to the events of type (b).

The way this is solved is developed in Section 3.2.

2.3.3 Handling events.

When an event of type (a.i) is encountered, a point location is performed. The decomposition into trapezoids allows us to locate such a point easily in practice, either in a naive way by testing all the trapezoids, or by walking along a line, or by using a dynamic point location structure [14].

When a trapezoid disappears, the 2D arrangement needs to be updated: the trapezoid is replaced by other trapezoids, and its neighbors are modified, too.

In particular, all the events of Section 2.2 are some of these events.

Events where walls coincide are events introduced by the vertical decomposition. Figure 3 shows one case of event, which is the case when two vertical walls, both defined by the intersection of two conic arcs, coincide. After the event, the two walls are exchanged, and all the neighboring trapezoids are modified.

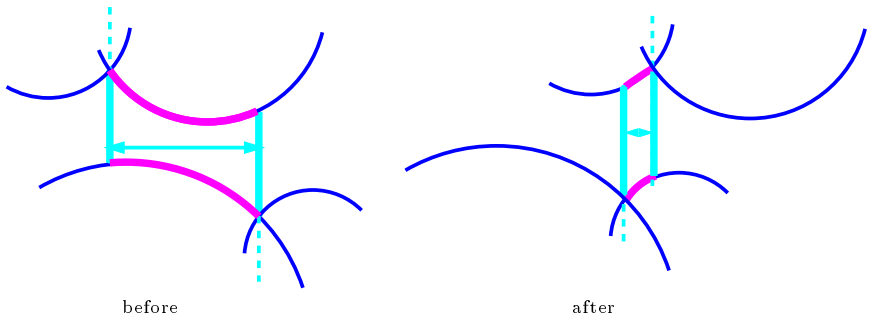


Fig. 3. A case of event when two walls coincide

Another advantage of the trapezoidal decomposition is that it induces the so-called *vertical decomposition* [4,5,27] of the arrangement of quadrics in $\mathbb{R}^3$.

Indeed, the regions swept by the trapezoids are the simple cells of the vertical decomposition. Namely, let us describe the correspondence between features in the 2D arrangement of conics in the sweeping plane and the 3D vertical decomposition (see Figure 4):

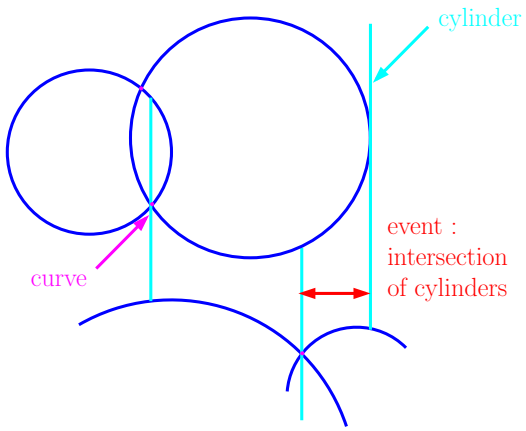


Fig. 4. Trapezoids and cylinders

- An intersection point between two conics in the sweeping plane is sweeping an intersection curve between two quadrics in 3D.
- A vertical wall in the arrangement of conics in the sweeping plane is sweeping a vertical cylinder in the 3D vertical decomposition.
  Depending on whether the wall is defined by an extremal point or an intersection point, the cylinder is defined by one quadric or by the intersecting curve of two quadrics.
- An event when two vertical walls coincide in the sweeping plane corresponds to the line of intersection of the two corresponding cylinders in the 3D vertical decomposition. In the case when the two walls are defined by the two vertices of the same conic arc in the sweeping plane, that are the intersections with two other conics, then, the event (the two walls coincide) corresponds to the intersection of 3 quadrics (case (c)).

In addition, when a quadric is first hit (resp. when a quadric is left) then we add, in the 3D decomposition, a planar boundary whose shape coincides exactly with the trapezoid in the sweeping plane where the new conic is inserted (resp. the trapezoid that is obtained after the removal of the conic). In this way, we obtain 3D cells of constant size.

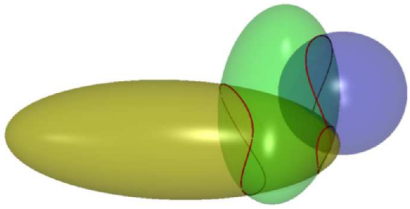


Fig. 5. Vertical cylinders are self-intersecting

Note that, since the intersection of two quadrics is a curve of degree 4, cylinders in the vertical decomposition are self-intersecting (see Figure 5, that shows quadrics and their intersecting curve seen from $z = +\infty$): indeed, the projection of the curve on a plane orthogonal to the z-axis is self-intersecting, so, the surface that is constructed by the vertical walls defined by points of this curve during the sweep (remember that the walls of the trapezoidal decomposition in the sweeping plane are parallel to the z-axis) is a self-intersecting cylinder.

2.5 Algorithm

Let us summarize here the above discussion and the different steps of the algorithm.

Algorithm 2.1

(1) **Initialization:** compute an initial trapezoidal map, and an initial list L of events to be considered.
   i. $L \longleftarrow \emptyset$
   ii. Compute events of type (a.i) and insert them in L in sorted order
   iii. Initialize the sweep by computing the arrangement of conics at a given value s
   iv. For each trapezoid in this planar arrangement
      - compute the events
        - when its ceiling and floor intersect
        - when its wall coincide
      - insert these events into the sorted list L

(2) **Sweep loop:**
   While $L \neq \emptyset$,
   let e be the first event in L
   (a) update the trapezoidal map by
      - removing the trapezoids that disappear at event e

```
      • replacing them by the trapezoids created at event e
(b) for each trapezoid created
      • compute the events
          · when its ceiling and floor intersect
          · when its wall coincide
      • insert these events into the sorted list L
(c) remove e from L
```

Here, the sweeping plane is initialized by picking any `s` value and computing the arrangement of the sections of all the quadrics for this value. This assumes that the sweep will then be performed in both directions, starting from this value `s`. However, initializing the sweeping plane at $-\infty$ could be better, using the infimaximal boxes scheme introduced by Mehlhorn and Seel [19].

2.6 Combinatorial complexity

Proposition 2.2 *For a generic sweeping direction, algorithm 2.1 computes the vertical decomposition of the arrangement of n quadrics in general position in $\mathcal{O}(n \log^2 n + V \log n)$ time, where V is the size of the vertical decomposition.*

Shaul and Halperin describe the data structures that allow to obtain this complexity result [28]. They use in particular the dynamic point location structure of Goodrich and Tamassia [14] for location into monotone subdivisions.

Whereas the complexity of the arrangement is $\mathcal{O}(n^3)$, the complexity V of the vertical decomposition is known to be bounded by $\mathcal{O}(n^3 \beta(n))$ where $\beta(n) = 2^{\alpha(n)^{16}}$ and $\alpha(n)$ is the very slowly growing pseudo-inverse of Ackermann's function [4].

Note that, though the size of this decomposition is larger than the size of the arrangement, it is asymptotically smaller than the size $\mathcal{O}((n\,d)^{2^{k-1}}) = \mathcal{O}(n^4)$ (where n is the number of quadrics, $d = 2$ their degree, $k = 3$ the dimension) of Collins' decomposition [6,2].

3 Algebraic aspects

We focus here on the algebraic aspects of the method. A key ingredient is the manipulation of algebraic numbers. We describe how to represent them and how to evaluate predicates on them. Next, we describe how to deal with solving of multivariate systems, and how to reduce the problem to the manipulation

of algebraic numbers. Finally, we detail their use in the geometric problems, which occur in the quadric arrangement computation.

3.1 Representation and comparison of algebraic numbers

We recall that a real algebraic number α is the root of a polynomial in $\mathbb{Z}[X]$. The degree of α is the minimal degree of such a polynomial $\in \mathbb{Z}[X]$, which vanishes at α.

We represent an algebraic number α as a pair: a univariate polynomial P such that $P(\alpha) = 0$ and an interval $]a, b[$ containing α and such that there is no other root of P in this interval.

The isolation of the roots of a square-free univariate polynomial can be made using Descartes' rule [8]. Several effective algorithms have been developed, based on this simple rule [16,25,23,21].

In the predicates used in the algorithm, we need to manipulate algebraic numbers. In all the cases the predicates reduce to the comparison of algebraic numbers, which are performed as follows: we are given two numbers $\alpha = (P,]a, b[)$ and $\beta = (Q,]c, d[)$. If the two intervals are disjoint, we can immediately conclude. If it's not the case we will use Sturm sequences. More precisely we use the following definition:

Definition 3.1 *(Sturm sequences) Let f, g be two univariate polynomials. A polynomial sequence $f_0 = f, f_1 = g, \ldots, f_s$ is a Sturm sequence if:*

- *f_s divides all the f_i, $i = 1 \ldots s$. Let $\delta_i = f_i / f_s$, $i = 1 \ldots s$.*
- *If c is a real such that $\delta_j(c) = 0$ with $0 < j < s$ then $\delta_{j-1}(c)\delta_{j+1}(c) < 0$.*
- *If c is a real such that $\delta_0(c) = 0$ then $\delta_0(x)\delta_1(x)$ has the sign of $x - c$ in a neighborhood of c.*

An efficient way to compute a Sturm sequence is to compute a Sturm-Habicht sequence.

Definition 3.2 *(Sturm-Habicht sequences) Let P and Q be univariate polynomials, $p = sup(deg(P), deg(Q)+1)$, $\mathrm{coef}_k(P)$ the coefficient of x^k in P, and $\delta_k = (-1)^{k(k-1)/2}$.*

The Sturm-Habicht sequence of P and Q is defined inductively as follows:

- *$H_p = P$, $h_p = 1$.*
- *$H_{p-1} = Q$.*

Assume that we have computed $H_p, \ldots, H_{j-1}$, $h_p, \ldots, h_j$ with $h_j \neq 0$ and

$H_{j-1} \neq 0$. *Let* $k = deg(H_{j-1})$. *Then:*

- *If* $k < j-1$, *let* $H_k = \delta_{j-k}\frac{\mathrm{coef}_k(H_{j-1})^{j-1-k}}{h_j^{j-1-k}}H_{j-1}$, $h_{j-1} = 1$.
  For $l \in \mathbb{N}$ *with* $k < l < j-1$, *let* $H_l = 0$, $h_l = 0$.
- *Let* $h_k = \mathrm{coef}_k(H_k)$, $H_{k-1} = \delta_{j-k+2}\frac{\mathrm{Prem}(Hj, H_{j-1})}{h_j^{j-k+1}}$ *(where* Prem *is the pseudo-remainder)*

The main interest of this construction is that the polynomials in the constructed sequence are related to the sub-resultants of P and Q. Thus the size of their coefficients is bounded almost linearly, by Hadamard identity (see [2]).

For any sequence S of real polynomials and $a \in \mathbb{R}$, we denote by $V(S,a)$ the number of variations of signs of the value of the polynomials in S at a. Then we have the well-known theorem of Sturm (see for instance [2]):

Proposition 3.3 *Assume* $S = Sturm(P, P'Q)$ *and* $]a,b[$ *is an interval such that* $P(a)P(b) \neq 0$. *The difference* $V(S,a)$ *and* $V(S,b)$ *is equal to the difference between the number of roots* α *of* P *in* $]a,b[$ *(without multiplicity) such that* $Q(\alpha) > 0$ *and the number of roots* α *of* P *in* $]a,b[$ *such that* $Q(\alpha) < 0$ *:*

$$V(S,a) - V(S,b) = Z_{Q>0}(P) - Z_{Q<0}(P).$$

Remark 3.4 *If* P *or* Q *is square-free, in proposition 3.3, the computation of* $Sturm(P,Q)$ *is sufficient.*

Let us describe briefly how we use this result to compare two algebraic numbers $\alpha = (P,]a,b[)$ and $Q = (\beta,]c,d[)$, assuming for simplicity that α and β are *simple roots* of P and Q. If $b < c$ (resp. $d < a$) we have $\alpha < \beta$ (resp. $\beta < \alpha$). Let us assume now that $a < c < b < d$ (the other cases being treated similarly). First we compute s the sign of $(P(a)P(c))$. If $s < 0$, then we have $\alpha \in]a,c[$ and $\alpha < \beta$. If $s = 0$, we have $\alpha = c$ (since $\alpha \neq a$), which implies that $\alpha < \beta$. Otherwise $s > 0$, P has no root in the interval $[a,c]$. We compute $S = Sturm(P, P'Q)$ and $v := V(S,c) - V(S,b)$. Let us assume first that $Q(c) > 0$, $Q(b) < 0$. Then if $v = 1$, by Sturm's theorem $Q(\alpha) > 0$ and $\alpha < \beta$. If $v = -1$, $Q(\alpha) < 0$ and $\alpha > \beta$. If $v = 0$, then $Q(\alpha) = 0$ and $\alpha = \beta$. If now $Q(c) < 0$, $Q(b) > 0$, we negate the previous output. Finally, if $Q(c)$ and $Q(b)$ are of the same sign, then $\alpha < \beta$.

Regarding the complexity of this method, the effective computation of sequences is made using Sturm-Habicht sequences. For two polynomials P and Q of degree p and q and with respective bit-length of coefficients t and t' then the bit-length of the coefficients in the Sturm-Habicht sequence is $\mathcal{O}((t + t' + \log(p+q))(p+q))$ and the computation of the sequence is made in $\mathcal{O}(pq)$ arithmetic operations [26]. We will use this result to bound the complexity of the algebraic operations involved in the sweeping algorithm.

The events involved in the algorithm require as we have seen in Sections 2.2 and 2.3, the resolution of quadratic equations in at most 4 variables (system (1)). This problem can be reduced to an univariate problem and thus to the manipulation of real algebraic numbers, as follows.

We denote by $f_1, \ldots, f_m$ the polynomial equations in $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \ldots, x_n]$ that we want to solve. The quotient ring $\mathbb{R}[\mathbf{x}]/(f_1, \ldots, f_m)$ of polynomials modulo $f_1 = 0, \ldots, f_m = 0$ is denoted by $\mathcal{A}$. In the case that we consider here, where the number of complex roots is finite, the quotient algebra $\mathcal{A}$ is a finite dimensional vector space.

We consider the operators of multiplication M_{x_i} $i = 1, \ldots, n$ by the variables x_i in the ring $\mathcal{A}$:

$$M_{x_i} : \mathcal{A} \to \mathcal{A}$$
$$a \mapsto x_i\, a$$

The system has finitely many roots over the algebraic closure of the field $\mathbb{R}$, if the quotient algebra is a vector space of finite dimension over $\mathbb{R}$. Then, the algebraic resolution of the system is performed by analyzing the eigenvalues and eigenvectors of these operators. A numerical approximation of the roots of the system is obtained by computing the common eigenvectors of these operators [1,7,20,22,29].

But these operators can also be used to describe the solution points as the image, by a rational map, of the roots of a univariate polynomials. In other words the real coordinates of the solutions are rational functions evaluated at real algebraic numbers, whose defining equations can be defined explicitly from the matrices M_{x_i}. Let us described briefly an algorithm that yields the so-called *Rational Univariate Representation* (RUR) of the roots:

Algorithm 3.5
`Input: The tables of multiplication by` $x_1, \ldots, x_n$ `in` $\mathcal{A}$

`(1) Compute the determinant` $\Delta(\mathbf{u}) := \det(u_0\,\mathrm{Id} + u_1\,M_{x_1} + \cdots + u_n\,M_{x_n})$ `and its square free part` $d(\mathbf{u})$.
`(2) Choose a generic` $t \in \mathbb{R}^{n+1}$ `and compute the first coefficients of`

$$d(t + \mathbf{u}) = d_0(u_0) + u_1\,d_1(u_0) + \cdots + u_n\,d_n(u_0) + \cdots$$

`considered as a polynomial in` $u_1, \ldots, u_n$.

Output: the roots of the system $f_1 = 0, \ldots, f_m = 0$ are

$$\zeta_1 = \frac{d_1(\alpha)}{d_0'(\alpha)}, \ldots, \zeta_n = \frac{d_n(\alpha)}{d_0'(\alpha)}$$

for α a root of the univariate polynomial equation $d_0(\alpha) = 0$.

For details on this construction, see [18,13,24,2,11]. The generic condition required on $t \in \mathbb{R}^{n+1}$ is that it *separates* the roots: $\sum_{i=0}^{n} \zeta_i t_i \neq \sum_{i=0}^{n} \zeta_i' t_i$ if ζ and ζ' are two distinct solutions of the system. Methods to find a generic t are described for instance in [24].

In order to get a minimal rational univariate representation, one can factorize $d_0(u_0)$ and keep the irreducible factors, which divide the numerator of the fraction obtained by substitution of x_i by $\frac{d_i(u_0)}{d_0'(u_0)}$.

This RUR allows us to replace the treatment of solutions of a multivariate system by the manipulation of algebraic numbers of degree at most the number of complex solutions of the system. We are going to use this representation of event points in the arrangement, in order to reduce the evaluation of predicates on these points to the comparison of algebraic numbers.

Another important aspect of the RUR is that we can compute a RUR of a polynomial system with coefficients in an algebraic extension $\mathbb{Q}[\theta]$, for θ an algebraic number. Assuming that P is the minimal polynomial of θ, computing the tables of multiplication and the RUR of the roots over $\mathbb{Q}[\theta]$ require field arithmetic operations and equality test in $\mathbb{Q}[\theta]$, which are performed by reduction modulo P.

3.3 Point location in the trapezoidal map

Let us describe now how we decide whether a point lies in a trapezoid. As we will see, it reduces to compare polynomial functions of algebraic numbers, and eventually to compute signs of rational expressions in the coefficients of the quadrics.

Suppose that we are at a sweeping position $x - x(p) = 0$ and we want to test if $p = (x(p), y(p), z(p))$ belongs to a given trapezoid. Then, the following operations must be performed:

(1) comparison with each wall of the trapezoid; such a comparison consists

in answering the test

$$y(p) \quad \begin{matrix} < \\ = \\ > \end{matrix} \quad y\text{-root of } \left(S_{ij}^{(x(p))}\right) \text{ or } \left(S_{k}^{(x(p))}\right) \quad ?$$

(2) comparison with the ceiling and the floor of the trapezoid; such a comparison is achieved by the test

$$\text{sign of } Q_i(x(p), y(p), z(p)) \quad ?$$

If the point is outside $Q_i^{(x(p))}$ (see Figure 6), this sign alone does not allow us to test whether the p is above or below the conic.

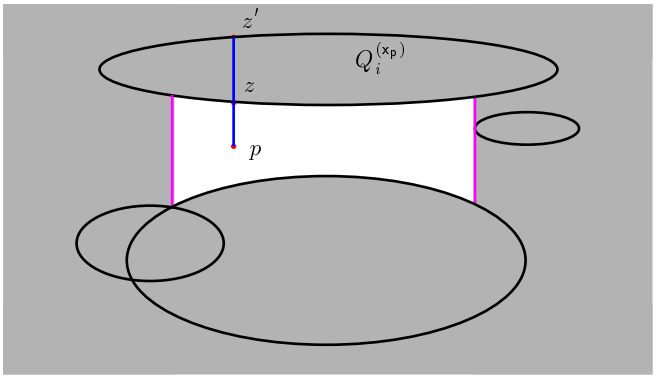


Fig. 6. Testing whether a point belongs to a trapezoid

Then an additional test leads to the conclusion

$$z(p) \quad \begin{matrix} < \\ > \end{matrix} \quad \text{middle of } z\text{-roots of } Q_i(x(p), y(p), z) = 0 \quad ?$$

We use the comparison with the middle of the roots instead of the comparison with the roots, because it is a rational function of the coefficients of $Q_i(x(p), y(p), z)$, whereas the roots above p involved square-roots.

A point location must be performed for each event of type (a.i) corresponding to the apparition of a new conic in the planar section. Such an event s is given by a solution of a system of the form (see Section 2.2):

$$\begin{cases} Q_{i_1}(\mathsf{s}, y, z) = 0 & \text{degree 2 in } \mathsf{s}, y, z \\ \partial_y(Q_{i_1}(\mathsf{s}, y, z)) = 0 & \text{degree 1 in } \mathsf{s}, y, z \\ \partial_z(Q_{i_1}(\mathsf{s}, y, z)) = 0 & \text{degree 1 in } \mathsf{s}, y, z \end{cases} \tag{2}$$

We compute the RUR associated to this system:

$$\begin{cases} f_2(\alpha) = 0 \\ x \quad = F_1(\alpha) \\ y \quad = F_2(\alpha) \\ z \quad = F_3(\alpha) \end{cases}$$

where f_2 is a univariate polynomial, $F_i, i = 1 \ldots 3$ are rational fractions. By Bézout theorem, the number of solution of the system (2) is at most 2. Thus the degrees of f_2 and α is at most 2. We isolate the roots of f_2 using Sturm sequences or Descartes' rule, or any specialized method. For this system we have at most two real solutions that are represented as explained in section 3.1. From this isolation operation, we also deduce the order of the corresponding x-coordinates of the solutions of the system.

We detail the steps of location of a point $p_\alpha = (F_1(\alpha), F_2(\alpha), F_3(\alpha))$ in the case where the floor and ceiling of the trapezoid are associated with the quadrics Q_{k_1} and Q_{k_2}.

Step 1: We have to evaluate the sign of Q_{k_l}, $l = 1, 2$ at p_α, which by substitution by the RUR of p_α boils down to the sign of polynomial functions of α. This is performed by application of Sturm's theorem (see proposition 3.3).

Step 2: We need to compare y_α with the y-coordinate of the solutions of systems of types $(S_{ij}^{(x_\alpha)})$ or $(S_k^{(x_\alpha)})$. We compute the RUR of such a system over $\mathbb{Q}(\alpha)$:

$$\begin{cases} g_\alpha(\beta) = 0 \\ y \quad = G_\alpha(\beta) \\ z \quad = H_\alpha(\beta) \end{cases}$$

where g_α is a polynomial with coefficients in $\mathbb{Q}(\alpha)$ and G_α, H_α are rational fractions with coefficients in $\mathbb{Q}(\alpha)$. The isolation of the real roots of $g_\alpha(y) = 0$ is performed using Descartes' rule or Sturm sequences over $\mathbb{Q}(\alpha)$. For this purpose, we need to compute the sign of algebraic expressions of α, which are obtained by application of proposition 3.3.

3.4 Detecting and comparing events

Events of type (a.i) have already been presented. Let us now study how other events can be detected.

A trapezoid is defined by two vertical walls, a floor and a ceiling. To predict

how and when a trapezoid will disappear, we need to compute when its floor and its ceiling collide, or when its vertical walls coincide.

The first computation consists in computing the parameter s for which two conics $Q_i^{(\mathsf{s})}(y,z)$ and $Q_j^{(\mathsf{s})}(y,z)$ get tangent in a point (y,z). A solution s (the x-coordinate of a solution of the system (b)) is an algebraic number of degree at most 8, since it corresponds to the x-coordinate of a point at the intersection of 3 quadrics.

For the second computation, we have to perform comparisons between abscissae y of intersection points between conics in the (y,z) sweeping plane for $x = \mathsf{s}$. Two conics (say $Q_i^{\mathsf{s}}(y,z)$ and $Q_j^{\mathsf{s}}(y,z)$) have up to four intersection points, and we distinguish among them according to their index in the lexicographic y,z-order.

This order changes when the resultant $R_{\mathsf{s}}(y)$ of $Q_i^{\mathsf{s}}(y,z)$ and $Q_j^{\mathsf{s}}(y,z)$ in z has a multiple root in y. These events can also be characterized by the system:

$$\begin{cases} Q_i(x,y,z_1) & = 0 \\ Q_j(x,y,z_1) & = 0 \\ Q_i(x,y,z_2) & = 0 \\ Q_j(x,y,z_2) & = 0 \\ \lambda(z_1 - z_2) = 1 & \end{cases}.$$

We maintain this information during the sweep. The update in the trapezoidal decomposition consists simply in renumbering the points of intersection of Q_i and Q_j.

The second computation consists in computing the parameter s for which the y-roots of two given systems of type $(S_{ij}^{(\mathsf{s})})$ or $(S_k^{(\mathsf{s})})$ coincide. The worst case, in terms of algebraic degree, is when the y-coordinate of the intersection between two conics coincides with the y-coordinate of the intersection between two other conics (see Equation (1)). This leads to the computation of points of intersection of 4 quadrics in a space of dimension 4:

$$\begin{cases} Q_1(x,y,z_1) = 0 \\ Q_2(x,y,z_1) = 0 \\ Q_3(x,y,z_2) = 0 \\ Q_4(x,y,z_2) = 0 \end{cases}.$$

(if $Q_{i_1} = Q_{i_2}$, we add the new equation $\lambda(z_1 - z_2) = 1$, or we replace Q_{i_1}

by $\frac{Q_{i_1}(x,y,z_1)-Q_{i_2}(x,y,z_2)}{z_1-z_2}$). The coordinates of the solutions of such a system lie in an algebraic extension of degree at most 16. They are represented using a RUR by

$$\begin{cases} f(\alpha) = 0 \\ x \quad = F_1(\alpha) \\ y \quad = F_2(\alpha) \\ z_1 \quad = F_3(\alpha) \\ z_2 \quad = F_4(\alpha) \end{cases} \tag{3}$$

where f is of degree ≤ 16 and $\alpha = (f,]a, b[)$ is an algebraic number.

For the complete update of the trapezoidal decomposition, we need to detect which point of intersection of Q_1 and Q_2, and which point of intersection of Q_3 and Q_4 have the same y-coordinate, at the position $x = F_1(\alpha)$, given by (3). We have to check which point of intersection of $Q_1(F_1(\alpha), y, z), Q_2(F_1(\alpha), y, z)$ is the point $(F_2(\alpha), F_3(\alpha))$. This reduces to compare the y-coordinates (and the z-coordinates) of the solutions of such a system, using again a RUR of the solutions over $\mathbb{Q}(\alpha)$. We proceed similarly for Q_3 and Q_4.

3.5 Height estimation

In order to analyze more precisely the complexity of predicates of the algorithm, we introduce the notion of *height* of a reduced rational function of the input parameters (here the coefficients of the quadrics) as the maximum degree of its numerator and denominator as polynomial functions of the input parameters. The height of an algebraic number will be the maximum of the heights of the coefficients of its minimal unitary polynomial defining this number. Notice that this notion of height differs slightly from the height used in arithmetic. Let us analyze our algorithm in terms of this height.

The comparison of two algebraic numbers of degree d_1 and d_2 is a key operation of our algorithm. From a practical point of view, if H is a bound on the height of the coefficients of the polynomials defining the two algebraic numbers, then their comparison via Sturm Habicht method (section 3.1) involves only computations of sub-minors of the Sylvester matrix of the two polynomials, that is polynomial functions of the coefficients of the defining polynomials of degree at most $d_1 + d_2$. Thus, it involves algebraic expressions of height at most $(d_1 + d_2)\, H$. Similarly, computing the sign of a polynomial expression of degree d_1 and height $\leq H$ at a root of degree d_2 and height $\leq H$ requires the

evaluation of signs of height $\leq (d_1 + d_2)\, H$.

Our algorithm for the arrangement of quadrics requires the intersection of 3 or 4 quadrics in 3 or 4 variables. A rational univariate representation of the roots can be deduced from the resultant of the system and a generic linear form [3]. The resultant of a system $f_0 = 0, \ldots, f_{n+1} = 0$ in n variables is of degree $\prod_{j \neq i} \deg(f_i)$ in the coefficients of the polynomial f_i. We deduce that the RUR of the solutions of 3 (resp. 4) quadratic equations in 3 (resp. 4) variables have heights less than $2 \times 2 + 2 \times 2 + 2 \times 2 = 12$ and (resp. $2 \times 2 \times 2 + 2 \times 2 \times 2 + 2 \times 2 \times 2 + 2 \times 2 \times 2 = 32$). We denote hereafter by H an effective bound on the height of the algebraic numbers involved in the algorithm: $H \leq 32$.

For computing the vertical decomposition of the arrangement, we need to compare algebraic numbers of degree at most 16 and height H, the height of the algebraic quantities will be bounded by $(16 + 16)\, H = 32\, H$. We also check that the degree of the algebraic expressions involved in the location steps (section 3.3, step 1 and 2) and the treatment of events (section 3.4) is bounded by 16 and their height is bounded by H. Thus computing their sign at a root of degree ≤ 16 and height $\leq H$ requires algebraic quantities of height $\leq 32\, H$.

To describe the complete arrangement, the vertices of a given facet are sorted around this facet. In particular, two vertices on a given curve are ordered on this curve. Such a curve (intersection of two quadrics) is of degree 4 and a vertex is the intersection of this curve with another quadric. Thus it is of degree 8 and height $H \leq 12$. Comparing two vertices reduces to comparing algebraic numbers of degree 8. As we have seen, this involves algebraic expressions of height less than $(8 + 8)\, H = 16\, H$.

4 Conclusion

4.1 Partial decomposition

The trapezoidal decomposition allows an output-sensitive complexity, in terms of the size of the 3D vertical decomposition (see Section 2.6). However, as seen in Section 3.5, the current vertical decomposition leads to a non-optimal (in terms of degree) algebraic complexity.

In order to avoid to go up to degree-16 algebraic numbers, we may consider *partial decompositions*. In the case of triangles on the space, a partial decom-

position is obtained by drawing walls, in the planar section, only through endpoints of line segments, and not through intersection points between two segments [28]. It is well known that in practice, the number of cells in the vertical decomposition can be much bigger than the number of cells in the arrangement. It was shown experimentally, in the case of triangles in 3D, that the number of cells in a partial decomposition can be smaller.

In the case of quadrics, we have seen that the highest degree predicates we get come from the comparison of the y-coordinates of intersections of conics in the sweeping plane. The first idea for a partial decomposition would be to draw vertical walls only through the points of vertical tangency of the conics in the planar sections, which generalize naturally endpoints of line segments. In this way, the degree of the algebraic numbers involved in the computations would decrease. However working out all the details of the algorithm raises a few difficulties and could not be done so far.

4.2 Implementation

The algorithm will be implemented using the CGAL [2] library and the SYNAPS [3] library.

The robustness of the algorithm strongly depends on the evaluation of the predicates. The algorithm assumes exact predicates.

The SYNAPS library allows to manipulate algebraic numbers represented implicitly as roots of polynomials. The comparisons of two such algebraic numbers is performed by numerical refinement steps and the use of a Sturm-Habicht sequence if necessary (as described in section 3.1).

The LEDA [4] and CORE [5] libraries currently allow to compare exactly algebraic numbers, when these numbers have an explicit representation using radicals, which may not be the case here.

Some work is in progress in each of these libraries to achieve exact comparison of implicit algebraic numbers. We made preliminary experiments with SYNAPS, that can efficiently compute approximations of roots of polynomial systems with error bounds that can be deduced for instance from error bounds on corresponding eigenproblems. The implementation of these error bounds is under progress.

[2] **http://www.cgal.org**
[3] **http://www-sop.inria.fr/galaad/logiciels/synaps/**
[4] **http://www.algorithmic-solutions.com/enleda.htm**
[5] **http://www.cs.nyu.edu/exact/core/**

References

[1] W. Auzinger and H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *Proc. Intern. Conf. on Numerical Math.*, volume 86 of *Int. Series of Numerical Math*, pages 12–30. Birkhäuser Verlag, 1988.

[2] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry.* Springer-Verlag, Berlin, 2003. ISBN 3-540-00973-6.

[3] L. Busé, M. Elkadi, and B. Mourrain. Using projection operators in computer aided geometric design. In *Topics in Algebraic Geometry and Geometric Modeling,*, pages 321–342. Contemporary Mathematics, 2003.

[4] Bernard Chazelle, H. Edelsbrunner, Leonidas J. Guibas, and Micha Sharir. A singly-exponential stratification scheme for real semi-algebraic varieties and its applications. In *Proc. 16th Internat. Colloq. Automata Lang. Program.*, volume 372 of *Lecture Notes Comput. Sci.*, pages 179–193. Springer-Verlag, 1989.

[5] Bernard Chazelle, H. Edelsbrunner, Leonidas J. Guibas, and Micha Sharir. A singly-exponential stratification scheme for real semi-algebraic varieties and its applications. *Theoret. Comput. Sci.*, 84:77–105, 1991.

[6] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conference on Automata Theory and Formal Languages*, volume 33 of *Lecture Notes Comput. Sci.*, pages 134–183. Springer-Verlag, 1975.

[7] R.M. Corless, P.M. Gianni, and B.M. Trager. A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In W.W. Küchlin, editor, *Porc. ISSAC*, pages 133–140, 1997.

[8] R. Descartes. *Géométrie*, volume 90-31 of *A source book in Mathematics.* Havard University press, 1636.

[9] Laurent Dupont, Daniel Lazard, Sylvain Lazard, and Sylvain Petitjean. Near-optimal parameterization of the intersection of quadrics. In *Proc. 19th Annu. ACM Sympos. Comput. Geom.*, pages 246–255, 2003.

[10] Arno Eigenwillig, Lutz Kettner, Elmar Schömer, and Nicola Wolpert. Complete, exact, and efficient computations with cubic curves. In *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, 2004. to appear.

[11] M. Elkadi and B. Mourrain. *Introduction à la résolution des systèmes d'équations algébriques*, 2003. Notes de cours, Univ. de Nice (310 p.).

[12] N. Geismann, M. Hemmer, and E. Schömer. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 264–273, 2001.

[13] L. Gonzalez-Vega, F. Rouillier, and M.F. Roy. *Symbolic Recipes for Polynomial System Solving.* Some Tapas of Computer Algebra. Springer, 1997.

[14] M. T. Goodrich and R. Tamassia. Dynamic trees and dynamic point location. *SIAM J. Comput.*, 28:612–636, 1998.

[15] D. Halperin. Arrangements. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 21, pages 389–412. CRC Press LLC, Boca Raton, FL, 1997.

[16] J. R. Johnson and Werner Krandick. Polynomial real root isolation using approximate arithmetic. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (Kihei, HI)*, pages 225–232 (electronic), New York, 1997. ACM.

[17] Sylvain Lazard, Luis Mariano Peñaranda, and Sylvain Petitjean. Intersecting quadrics: An efficient and exact implementation. In *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, 2004. to appear.

[18] F.S. Macaulay. On the Resolution of a given Modular System into Primary Systems Including some Properties of Hilbert Numbers. *Proc. London Math. Soc.*, pages 66–121, 1912.

[19] Kurt Mehlhorn and Michael Seel. Infimaximal frames: A technique for making lines look like segments. In *Abstracts 17th European Workshop Comput. Geom.*, pages 78–81. Freie Universität Berlin, 2001.

[20] B. Mourrain. Computing isolated polynomial roots by matrix methods. *J. of Symbolic Computation, Special Issue on Symbolic-Numeric Algebra for Polynomials*, 26(6):715–738, Dec. 1998.

[21] B. Mourrain, F. Rouillier, and M.-F. Roy. Bernstein's basis and real root isolation. Technical Report 5149, INRIA Rocquencourt, 2004.

[22] B. Mourrain and Ph. Trébuchet. Algebraic methods for numerical solving. In *Proc. of the 3rd International Workshop on Symbolic and Numeric Algorithms for Scientific Computing'01 (Timisoara, Romania)*, pages 42–57, 2002.

[23] B. Mourrain, M. Vrahatis, and J.C. Yakoubsohn. On the complexity of isolating real roots and computing with certainty the topological degree. *J. of Complexity*, 18(2):612–640, 2002.

[24] F. Rouillier. Solving zero-dimensional polynomial systems throuhg Rational Univariate Representation. *App. Alg. in Eng. Com. Comp.*, 9(5):433–461, 1999.

[25] F. Rouillier and P. Zimmermann. Efficient isolation of a polynomial real roots. *Journal of Computational and Applied Mathematics*, 162(1):33–50, 2003.

[26] M.F. Roy. Basic algorithms in real algebraic geometry: from Sturm theorem to the existential theory of reals. In *Lectures on Real Geometry in memoriam of Mario Raimondo*, volume 23 of *Exposition in Mathematics*, pages 1–67, 1996.

[27] Micha Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, New York, 1995.

[28] H. Shaul and D. Halperin. Improved construction of vertical decompositions of three-dimensional arrangements. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 283–292, 2002.

[29] Ph. Trébuchet. *Vers une résolution stable et rapide des équations algébriques.* PhD thesis, Université Pierre et Marie Curie, 2002.